

Multi-Objective Trajectory Optimization of a Hypersonic Reconnaissance Vehicle with Temperature Constraints

DISSERTATION

Tadeusz J. Masternak, DR-IV, DAFC

AFIT-ENV-DS-14-D-21

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

DISTRIBUTION STATEMENT A:
APPROVED FOR PUBLIC RELEASE, DISTRIBUTION UNLIMITED

The views expressed in this document are those of the author and do not reflect the official policy or position of the United States Air Force, the Department of Defense or the United States Government.

This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

AFIT-ENV-DS-14-D-21

MULTI-OBJECTIVE TRAJECTORY OPTIMIZATION OF A HYPERSONIC RECONNAISSANCE VEHICLE WITH TEMPERATURE CONSTRAINTS

DISSERTATION

Presented to the Faculty

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

in Partial Fulfillment of the Requirements for the

Degree of Doctor of Philosophy

Tadeusz J. Masternak, B.S., M.S., M.B.A

DR-IV, DAFC

December 2014

DISTRIBUTION STATEMENT A:
APPROVED FOR PUBLIC RELEASE, DISTRIBUTION UNLIMITED

AFIT-ENV-DS-14-D-21

MULTI-OBJECTIVE TRAJECTORY OPTIMIZATION OF A HYPERSONIC RECONNAISSANCE VEHICLE WITH TEMPERATURE CONSTRAINTS

Tadeusz J. Masternak, B.S., M.S., M.B.A
DR-IV, DAFC

Approved:

//signed//	8 December 2014
David R. Jacques, PhD (Chairman)	Date
//signed//	8 December 2014
William P. Baker, PhD (Member)	Date
//signed//	8 December 2014
Richard G. Cobb, PhD (Member)	Date

Accepted:

//signed//	9 December 2014
Adedeji B. Badiru, PhD Dean, Graduate School of Engineering and Management	Date

Abstract

This research determines temperature-constrained optimal trajectories for a scramjet-based hypersonic reconnaissance vehicle by developing an optimal control formulation and solving it using a variable order Gauss-Radau quadrature collocation method with a Non-Linear Programming (NLP) solver. The vehicle is assumed to be an air-breathing reconnaissance aircraft that has specified takeoff/landing locations, air-borne refueling constraints, specified no-fly zones, and specified targets for sensor data collections. A three degree of freedom scramjet aircraft model is adapted from previous work and includes flight dynamics, aerodynamics, and thermal constraints. Vehicle control is accomplished by controlling angle of attack, roll angle, and propellant mass flow rate. This model is incorporated into an optimal control formulation that includes constraints on both the vehicle and mission parameters, such as avoidance of no-fly zones and coverage of high-value targets. To solve the optimal control formulation, a MATLAB-based package called General Pseudospectral Optimal Control Software (GPOPS-II) is used, which transcribes continuous time optimal control problems into an NLP problem. In addition, since a mission profile can have varying vehicle dynamics and en-route imposed constraints, the optimal control problem formulation can be broken up into several "phases" with differing dynamics and/or varying initial/final constraints. Optimal trajectories are developed using several different performance costs in the optimal control formulation: minimum time, minimum time with control penalties, and maximum range. The resulting analysis demonstrates that optimal trajectories that meet specified mission parameters and constraints can be quickly determined and used for larger-scale operational and campaign planning and execution.

This is dedicated to my parents for always pushing me to get an education and to my wife and daughter for putting up with me getting more educated.

Acknowledgements

I would like to thank my research advisor, Dr. David Jacques, for his guidance and support throughout this research effort. In addition, I would like to thank my committee members, Drs. William Baker and Richard Cobb, for their invaluable guidance and insights.

Thanks to the Science, Mathematics And Research for Transformation (SMART) Scholarship for Service Program for giving me the resources to make this research a reality.

I would also like to thank Russell Howard and Maj Gen Dwyer Dennis for supporting me in getting my SMART scholarship and enabling me to pursue my studies full-time. Thanks also go to the AFRL/RQ leadership team for allowing me to perform this research on a full-time basis for an extended time.

Finally, special thanks go to my future co-workers and my wingmen for helping me make my research topic a reality: Maj Chris Arendt, Dave Adamczak, Bill Gillard, Glenn Liston, MSG Jeff Morris, Col Nate Smith, Col Angie Suplisson, and Dr. James Weber.

Tadeusz J. Masternak

Table of Contents

Page

Abstract iv

Acknowledgements vi

List of Figures xi

List of Tables xix

Nomenclature xxx

I. Introduction 1

1.1 Motivation 1
1.2 Historical Perspectives 4
1.3 Problem Description 9
1.3.1 Hypersonic Reconnaissance Vehicle Overview 10
1.3.2 Mission Scenarios 10
1.3.3 Problem Statement 14
1.4 Research Objectives 14
1.4.1 Research Assumptions 15
1.5 Dissertation Overview 16

II. Literature Review 17

2.1 Trajectory Optimization 17
2.2 Vehicle Models 19
2.3 Aerothermal Models 20
2.4 Optimal Control Theory 23
2.5 Pseudospectral Methods 25
2.5.1 Spectral Methods 25
2.5.2 Optimal Control Numerical Methods 27
2.5.3 Variable-Order Gaussian Quadrature Collocation Method 27
2.6 Multi-Objective Optimization Methods 32
2.7 Sensor Modeling 33
2.8 Summary 36

III. Modeling 37

3.1 Gravitational Model 37
3.2 Atmospheric Model 37
3.3 Vehicle Model 42
3.3.1 Aerodynamics Model 44
3.3.2 Propulsion Model 46

Page

3.4 State Dynamics . . . 48
3.4.1 Reference Frames . . . 49
3.4.2 States and Controls . . . 53
3.4.3 Equations of Motion . . . 53
3.5 Aerothermal Modeling . . . 56
3.6 Modeling Summary . . . 58

IV. Methodology . . . 60

4.1 Problem Statement . . . 60
4.2 Optimal Control Problem Formulation . . . 60
4.2.1 Maximum Range Problem . . . 61
4.2.2 Minimum Time Problem . . . 61
4.2.3 Minimum Time with Control Penalty Problem . . . 62
4.2.4 Minimum Time with Control Penalty, No-Fly Zone and g-limits Problem . . . 63
4.2.5 Minimum Time with Control Penalty, No-Fly Zone and Sensor Constraints Problem . . . 63
4.3 Sensor Modeling as Constraints . . . 65
4.4 Thermal Modeling as Constraints . . . 65
4.5 Methodology Summary . . . 67

V. Scenarios Results and Analysis . . . 69

5.1 Analysis Overview . . . 69
5.2 Scenario Results and Discussion . . . 74
5.2.1 Maximum Range (Climb and Cruise) Scenario . . . 74
5.2.2 Minimum Time (Climb and Cruise) Scenario . . . 80
5.2.3 Minimum Time with Control Penalty (Climb and Cruise) Scenario . . . 88
5.2.4 Minimum Time (Climb, Cruise, and Land) Scenario . . . 91
5.2.5 Minimum Time with Control Penalty (Climb, Cruise, and Land) Scenario . . . 98
5.2.6 Minimum Time with Control Penalty (Climb, Cruise, Refuel, Cruise, and Land) Scenario . . . 101
5.2.7 Minimum Time with Control Penalty, No-Fly Zone and g-limits (Climb, Cruise, Refuel, Cruise, and Land) Scenario . . 108
5.2.8 Minimum Time with Control Penalty, No-Fly Zone and Sensor Constraints (Climb, Cruise, Sense, and Cruise) Scenario 110
5.2.9 Minimum Time with Temperature Constraints (Climb and Cruise) Scenario . . . 119
5.2.10 Minimum Time with Control Penalty and Temperature Constraints (Climb, Cruise, Refuel, Cruise, and Land) Scenario . . . 122

Page

5.3 Rapid Mission Planning Utility . . . 129
5.4 Results and Analysis Summary . . . 132
VI. Parametric Analysis . . . 134

6.1 Parametric Analysis Overview . . . 134
6.2 Parametric Analysis Results . . . 135
6.2.1 Total Flight Time Profile . . . 135
6.2.2 Maximum Height Profile . . . 135
6.2.3 Maximum Velocity Profile . . . 136
6.2.4 Maximum Dynamic Pressure Profile . . . 137
6.2.5 Maximum g's Profile . . . 138
6.2.6 Maximum Nose Heating Rate Profile . . . 140
6.2.7 Maximum Nose Heating Load Profile . . . 140
6.2.8 Maximum Nose Temperature Profile . . . 141
6.2.9 Maximum Nose Temperature Duration Profile . . . 142
6.2.10 Maximum Body Heating Rate Profile . . . 144
6.2.11 Maximum Body Heating Load Profile . . . 144
6.2.12 Maximum Body Temperature Profile . . . 145
6.2.13 Maximum Body Temperature Duration Profile . . . 146
6.3 Parametric Analysis Insights . . . 148
6.4 Parametric Analysis Summary . . . 149

VII. Conclusions and Recommendations . . . 152

7.1 Conclusions . . . 152
7.2 Contributions . . . 154
7.3 Recommendations for Future Research . . . 155

Appendix A. Derivation of Chapman Equation Constants . . . 157

Appendix B. Linear-Quadratic Optimal Control Problem Example . . . 159

Appendix C. Scenario Results Detailed Charts . . . 163

C.1 Maximum Range; Climb and Cruise . . . 164
C.2 Minimum Time; Climb and Cruise . . . 172
C.3 Minimum Time with Control Penalty; Climb and Cruise . . . 180
C.4 Minimum Time; Climb, Cruise, and Land . . . 189
C.5 Minimum Time with Control Penalty; Climb, Cruise, and Land . . . 197
C.6 Minimum Time with Control Penalty; Climb, Cruise, Refuel, Cruise, and Land . . . 205
C.7 Minimum Time with Control Penalty, No-Fly Zone and g-limits; Climb, Cruise, Refuel, Cruise, and Land . . . 214
C.8 Minimum Time with Control Penalty, No-Fly Zone and Sensor Constraints; Climb, Cruise, Sense, and Cruise . . . 223
C.9 Minimum Time with Temperature Constraints; Climb and Cruise . . . 232

Page

C.10 Minimum Time with Control Penalty and Temperature Constraints (Climb, Cruise, Refuel, Cruise, and Land) 240

Appendix D. Parametric Analysis Detailed Charts 249

D.1 Trajectory Height vs. Path Length 249
D.2 Trajectory Height vs. Time 258
D.3 Stoichiometric Fuel-Air Ratio vs. Time 266
D.4 Dynamic Pressure vs. Time 275
D.5 g's vs. Time 283
D.6 Thermal Parameters vs. Time 292
D.6.1 Nose Heating Rate vs. Time 292
D.6.2 Nose Heating Load vs. Time 300
D.6.3 Nose Temperature vs. Time 309
D.6.4 Body Heating Rate vs. Time 317
D.6.5 Body Heating Load vs. Time 326
D.6.6 Body Temperature vs. Time 334

Appendix E. Draft Conference Article 343

Bibliography 376

Vita 384

List of Figures

Figure		Page
1	Air-breathing Propulsion Flight Regimes [50]	6
2	X-51 Waverider Separating from Booster [55]	7
3	Notional Mission Trajectory	12
4	Notional Mission Profile	13
5	Principles of Aerothermal Modeling [87]	21
6	Indirect vs. Direct Methods [8]	28
7	GPOPS Operational Flowchart [60]	29
8	Pseudospectral Collocation Point Selection Methods [32]	30
9	Notional Optical Radiometry Model [44]	34
10	1976 Standard Atmosphere Speed of Sound (a) Values	38
11	1976 Standard Atmosphere Speed of Sound (a) Curve Fit	39
12	1976 Standard Atmosphere Density (ρ) Curve Fits	40
13	1976 Standard Atmosphere Density (ρ) Curve Fits at High Altitude	41
14	GHAME Aircraft Configuration [84]	43
15	GHAME Lift Coefficient Curves [92] expressed as either (a) a linear function with slope and intercept coefficients as a function of M and (b) as a 3-D function of M and α	45
16	GHAME Drag Coefficient Curves [92] expressed as either (a) a linear function with slope and intercept coefficients as a function of M and (b) as a 3-D function of M and α	46
17	GHAME Specific Impulse (I_{sp}) [84]	47
18	GHAME Inlet Capture Area Ratio Curves [84] expressed as either (a) a linear function with slope and intercept coefficients as a function of M and (b) as a 3-D function of M and α	48
19	Reference Frames	50

Figure Page

20 Earth Reference Frame [82] 50

21 Velocity Reference Frame [82] 51

22 Stability Reference Frame [82] 52

23 Body Reference Frame [82] 53

24 LANMIN/EXITS Integration with MATLAB/GPOPS 67

25 Optimal Trajectory: Maximum Range, 2-D Flight Trajectory 75

26 Optimal Trajectory: Maximum Range, States 76

27 Optimal Trajectory: Maximum Range, States (Ascent Portion) 76

28 Optimal Trajectory: Maximum Range, Controls 77

29 Optimal Trajectory: Maximum Range, Controls (Ascent Portion) 78

30 Optimal Trajectory: Maximum Range, Path Constraints and g's 78

31 Optimal Trajectory: Maximum Range, Maximum Relative Error 79

32 Optimal Trajectory: Minimum Time (Climb and Cruise), 2-D Flight Trajectory 80

33 Optimal Trajectory: Minimum Time (Climb and Cruise), 2-D Flight Profile 81

34 Optimal Trajectory: Minimum Time (Climb and Cruise), States 82

35 Optimal Trajectory: Minimum Time (Climb and Cruise), Controls 82

36 Optimal Trajectory: Minimum Time (Climb and Cruise), Controls (Ascent Portion) 83

37 Optimal Trajectory: Minimum Time (Climb and Cruise), Path Constraints and g's 83

38 Optimal Trajectory: Minimum Time (Climb and Cruise), Heating Rates 84

39 Optimal Trajectory: Minimum Time (Climb and Cruise), Heating Loads 85

Figure Page

40 Optimal Trajectory: Minimum Time (Climb and Cruise), Temperatures . . . 86

41 Optimal Trajectory: Minimum Time (Climb and Cruise), Maximum Relative Error . . . 86

42 Optimal Trajectory: Minimum Time (Climb and Cruise), Mesh Interval History . . . 87

43 Optimal Trajectory: Minimum Time (Climb and Cruise), Mesh Interval and Collocation Point History . . . 88

44 Optimal Trajectory: Minimum Time with Control Penalty (Climb and Cruise), 2-D Flight Profile ($\epsilon = 10^{-1}$) . . . 89

45 Optimal Trajectory: Minimum Time with Control Penalty (Climb and Cruise), 2-D Flight Profile with $\epsilon = 10^{-3}$ . . . 90

46 Optimal Trajectory: Minimum Time with Control Penalty (Climb and Cruise), Controls . . . 90

47 Optimal Trajectory: Minimum Time with Control Penalty (Climb and Cruise), Controls (angular rate) . . . 91

48 Optimal Trajectory: Minimum Time with Control Penalty (Climb and Cruise), Heating Rates . . . 92

49 Optimal Trajectory: Minimum Time (Climb, Cruise, and Land), States 93

50 Optimal Trajectory: Minimum Time (Climb, Cruise, and Land), States (angular) . . . 93

51 Optimal Trajectory: Minimum Time (Climb, Cruise, and Land), Controls . . . 94

52 Optimal Trajectory: Minimum Time (Climb, Cruise, and Land), Path Constraints and g's . . . 95

53 Optimal Trajectory: Minimum Time (Climb, Cruise, and Land), Heating Rates . . . 96

54 Optimal Trajectory: Minimum Time (Climb, Cruise, and Land), Heating Loads . . . 96

55 Optimal Trajectory: Minimum Time (Climb, Cruise, and Land), Temperatures . . . 97

Figure Page

56 Optimal Trajectory: Minimum Time (Climb, Cruise, and Land), Maximum Relative Error 97

57 Optimal Trajectory: Minimum Time with Control Penalty (Climb, Cruise, and Land), Maximum Relative Error 98

58 Optimal Trajectory: Minimum Time with Control Penalty (Climb, Cruise, and Land), States 99

59 Optimal Trajectory: Minimum Time with Control Penalty (Climb, Cruise, and Land), Controls 100

60 Optimal Trajectory: Minimum Time with Control Penalty (Climb, Cruise, and Land), Controls (angular rate) 100

61 Optimal Trajectory: Minimum Time with Control Penalty (Climb, Cruise, and Land), Path Constraints and g's 101

62 Optimal Trajectory: Minimum Time with Control Penalty (Climb, Cruise, and Land), Heating Rates 102

63 Optimal Trajectory: Minimum Time with Control Penalty (Climb, Cruise, Refuel, Cruise, and Land), 3-D Flight Trajectory 103

64 Optimal Trajectory: Minimum Time with Control Penalty (Climb, Cruise, Refuel, Cruise, and Land), 2-D Flight Trajectory 104

65 Optimal Trajectory: Minimum Time with Control Penalty (Climb, Cruise, Refuel, Cruise, and Land), States 104

66 Optimal Trajectory: Minimum Time with Control Penalty (Climb, Cruise, Refuel, Cruise, and Land), Mach Number 105

67 Optimal Trajectory: Minimum Time with Control Penalty (Climb, Cruise, Refuel, Cruise, and Land), Controls 106

68 Optimal Trajectory: Minimum Time with Control Penalty (Climb, Cruise, Refuel, Cruise, and Land), Controls (angular rate) 106

69 Optimal Trajectory: Minimum Time with Control Penalty (Climb, Cruise, Refuel, Cruise, and Land), Heating Rates 107

70 Optimal Trajectory: Minimum Time with Control Penalty (Climb, Cruise, Refuel, Cruise, and Land), Temperatures 108

Figure Page

71 Optimal Trajectory: Minimum Time with Control Penalty (Climb, Cruise, Refuel, Cruise, and Land), Mesh Interval History 109

72 Optimal Trajectory: Minimum Time with Control Penalty, No-Fly Zone and g-limits (Climb, Cruise, Refuel, Cruise, and Land), 3-D Flight Trajectory 109

73 Optimal Trajectory: Minimum Time with Control Penalty, No-Fly Zone and g-limits (Climb, Cruise, Refuel, Cruise, and Land), No-Fly Distance 110

74 Optimal Trajectory: Minimum Time with Control Penalty, No-Fly Zone and g-limits (Climb, Cruise, Refuel, Cruise, and Land), Controls 111

75 Optimal Trajectory: Minimum Time with Control Penalty, No-Fly Zone and g-limits (Climb, Cruise, Refuel, Cruise, and Land), Path Constraints and g's 111

76 Optimal Trajectory: Minimum Time with Control Penalty, No-Fly Zone and Sensor Constraints (Climb, Cruise, Sense, and Cruise), 3-D Flight Trajectory 112

77 Optimal Trajectory: Minimum Time with Control Penalty, No-Fly Zone and Sensor Constraints (Climb, Cruise, Sense, and Cruise), 3-D Flight Profile 113

78 Optimal Trajectory: Minimum Time with Control Penalty, No-Fly Zone and Sensor Constraints (Climb, Cruise, Sense, and Cruise), 2-D Flight Trajectory 113

79 Optimal Trajectory: Minimum Time with Control Penalty, No-Fly Zone and Sensor Constraints (Climb, Cruise, Sense, and Cruise), No-Fly Distance 114

80 Optimal Trajectory: Minimum Time with Control Penalty, No-Fly Zone and Sensor Constraints (Climb, Cruise, Sense, and Cruise), Sensor Constraints 115

81 Optimal Trajectory: Minimum Time with Control Penalty, No-Fly Zone and Sensor Constraints (Climb, Cruise, Sense, and Cruise), Sensor Constraints (sensor activity) 115

82 Optimal Trajectory: Minimum Time with Control Penalty, No-Fly Zone and Sensor Constraints (Climb, Cruise, Sense, and Cruise), States 116

Figure Page

83 Optimal Trajectory: Minimum Time with Control Penalty, No-Fly Zone and Sensor Constraints (Climb, Cruise, Sense, and Cruise), States (angular) 116

84 Optimal Trajectory: Minimum Time with Control Penalty, No-Fly Zone and Sensor Constraints (Climb, Cruise, Sense, and Cruise), Controls 117

85 Optimal Trajectory: Minimum Time with Control Penalty, No-Fly Zone and Sensor Constraints (Climb, Cruise, Sense, and Cruise), Controls (angular rate) 117

86 Optimal Trajectory: Minimum Time with Control Penalty, No-Fly Zone and Sensor Constraints (Climb, Cruise, Sense, and Cruise), 3-D Flight Profile 118

87 Optimal Trajectory: Minimum Time with Control Penalty, No-Fly Zone and Sensor Constraints (Climb, Cruise, Sense, and Cruise), Sensor Look Angles During Sensing Phase 119

88 Optimal Trajectory: Minimum Time with Control Penalty, No-Fly Zone and Sensor Constraints (Climb, Cruise, Sense, and Cruise), Bank and Elevation Angles During Sensing Phase 120

89 Optimal Trajectory: Minimum Time with Control Penalty, No-Fly Zone and Sensor Constraints (Climb, Cruise, Sense, and Cruise), Sensor Look Angles During Entire Trajectory 121

90 Optimal Trajectory: Minimum Time with Temperature Constraints (Climb and Cruise), States 122

91 Optimal Trajectory: Minimum Time with Temperature Constraints (Climb and Cruise), Heating Rates 123

92 Optimal Trajectory: Minimum Time with Temperature Constraints (Climb and Cruise), Heating Loads 123

93 Optimal Trajectory: Minimum Time with Temperature Constraints (Climb and Cruise), Temperatures 124

94 Optimal Trajectory: Minimum Time with Control Penalty and Temperature Constraints (Climb, Cruise, Refuel, Cruise, and Land), 3-D Flight Trajectory 125

Figure Page

95 Optimal Trajectory: Minimum Time with Control Penalty and Temperature Constraints (Climb, Cruise, Refuel, Cruise, and Land), 2-D Flight Profile 126

96 Optimal Trajectory: Minimum Time with Control Penalty and Temperature Constraints (Climb, Cruise, Refuel, Cruise, and Land), States 127

97 Optimal Trajectory: Minimum Time with Control Penalty and Temperature Constraints (Climb, Cruise, Refuel, Cruise, and Land), Heating Rates 128

98 Optimal Trajectory: Minimum Time with Control Penalty and Temperature Constraints (Climb, Cruise, Refuel, Cruise, and Land), Heating Loads 128

99 Optimal Trajectory: Minimum Time with Control Penalty and Temperature Constraints (Climb, Cruise, Refuel, Cruise, and Land), Temperatures 129

100 Optimal Trajectory: Minimum Time with Control Penalty and Temperature Constraints (Climb, Cruise, Refuel, Cruise, and Land), Maximum Relative Error 130

101 Scenario Run Times 131

102 Total Flight Time vs. Mach Number and Great Circle Length 136

103 Maximum Height vs. Mach Number and Great Circle Length 137

104 Maximum Velocity vs. Mach Number and Great Circle Length 138

105 Maximum Dynamic Pressure vs. Mach Number and Great Circle Length 139

106 Maximum g's vs. Mach Number and Great Circle Length 139

107 Maximum Nose Heating Rate vs. Mach Number and Great Circle Length 140

108 Maximum Nose Heating Load vs. Mach Number and Great Circle Length 141

109 Maximum Nose Temperature vs. Mach Number and Great Circle Length 142

Figure Page

110 Maximum Nose Temperature Duration vs. Mach Number and Great Circle Length 143

111 Maximum Body Heating Rate vs. Mach Number and Great Circle Length 144

112 Maximum Body Heating Load vs. Mach Number and Great Circle Length 145

113 Maximum Body Temperature vs. Mach Number and Great Circle Length 146

114 Maximum Body Temperature Duration vs. Mach Number and Great Circle Length 147

115 Flight Times and Maximum Nose Temperatures vs. Mach Number . . . 149

116 Flight Times for Either Temperature Path Constraint or Changing Maximum Mach Number 150

117 Flight Times Differences Between Temperature Path Constraint and Changing Maximum Mach Number 150

118 Solution to Linear-Quadratic Optimal Control Problem, comparing exact to discretized approximate solutions 161

List of Tables

Table		Page
1	GHAME Reference Data [13] [84] [83]	44
2	MINIVER Settings	58
3	Scenario GPOPS-II settings	70
4	State and Control Limits	71
5	Scenario Attributes Summary	72
6	Scenario Timing Attributes	131
7	Numerical Approximation Error for Linear-Quadratic Optimal Control Problem	162

Nomenclature

— Letters —

A	aerodynamic axial force
A_0/A_c	inlet capture area ratio
A_{ic}	inlet capture area
A_{ref}	vehicle aerodynamic reference area
$\mathbf{a}_k$	basis function coefficients
A_0/A_{c_0}	inlet capture area ratio at zero alpha
A_0/A_{c_α}	inlet capture area ratio due to alpha
b_i	constraint for ith event group
ΔC_A	aerodynamic axial force coefficient increment due to viscous effects
C	optimal control path constraints
C_A	axial force coefficient
C_D	drag coefficient
C_L	lift coefficient
C_N	normal force coefficient
C_{D_0}	minimum drag coefficient
C_{L_α}	coefficient of lift due to alpha
$C_{L_{\alpha_0}}$	coefficient of lift at zero alpha

$cs(t)$	co-state
C_{N_α}	aerodynamic normal force coefficient
C_{N_β}	aerodynamic side force coefficient
C_{A_0}	aerodynamic zero-alpha axial force coefficient
C_{A_α}	aerodynamic axial force coefficient
C_{L_0}	zero-coefficient lift force
D	drag force
D	feasible design space
d	distance of the candidate path to the jth keep-out zone center
$\mathbf{d}(t)$	variation of candidate path from a reference path
e	event group
$f(\cdot)$	arbitrary function
F_N	combined aerodynamic and propulsive forces orthogonal to the velocity in the lift-drag plane
F_T	combined aerodynamic and propulsive forces along velocity vector
G	universal gravitational constant
g	gravitational acceleration
g_a	axial acceleration (in g's)
g_c	standard gravitational acceleration

g_i	inequality constraints
g_n	normal acceleration (in g's)
h	vehicle altitude
h_j	equality constraints
I_{sp}	specific impulse
J	performance measure or objective function or cost function
k	drag coefficient constant
$\mathcal{L}$	optimal control running cost (Lagrangian)
L	lift force
$\mathcal{M}$	optimal control terminal cost (Mayer)
$\dot{m}$	vehicle mass rate of change
$\dot{m}_f$	propellant mass flow rate
M	mach Number
m	vehicle mass
m_e	mass of the earth
N	aerodynamic normal force
n	polynomial order
N_e	number of event groups
N_s	number of sensor parameters

N_t	number of aerothermal analysis locations
N_z	number of no-fly zones
$p(t)$	penalty (cost) function
p^s	sigmoid penalty function steepness
$p^{\max}$	penalty function maximum value
$\mathbf{Q}$	weighted state deviation matrix
$\mathbf{q}$	integral constraint(s)
$\dot{Q}_{conv}$	convection heating rate
q	scalar weighting factor for deviation distance
q_∞	dynamic pressure
Q_{conv}	convection heating load
$\mathbf{R}$	weighted control usage matrix
r	radial distance from vehicle to earth center of mass
r	scalar weighting factor for control usage
r_e	mean earth radius
r_i	no-fly zone penalty weight
r_{nofly_j}	radius of jth no-fly zone
S	multi-objective set of admissible states
S	sensor parameter integrand

$s_i(\mathbf{x}(t))$	sensor parameter figure of merit for the entire trajectory
$s_i^{\max}$	sensor parameter constraint
S_{az}	sensor azimuth angle
S_{el}	sensor elevation angle
S_{LOS}	Line of Sight angle from horizon to sensor
S_{SR}	sensor slant range
$\mathbf{T_e^v}$	transformation matrix, from earth to velocity coordinate systems
$\mathbf{T_i^e}$	transformation matrix, from inertial to earth coordinate systems
$\mathbf{T_s^b}$	transformation matrix, from stability to body coordinate systems
$\mathbf{T_v^s}$	transformation matrix, from velocity to stability coordinate systems
T	thrust
t	time
T_i	surface temperature at the ith vehicle position
$T_i^{\max}$	maximum surface temperature at the ith vehicle position
$\mathbf{u}$	control(s)
U	set of admissible controls
$\mathbf{u}^*$	optimal control(s)
V	velocity
v	velocity (constant)

W	performance threshold
w_i	objective function term weights
$\dot{\mathbf{x}}$	state dynamics (equations of motion)
$\mathbf{x}$	state(s)
X	set of admissible states
x	x-position
$x_{b_{sensor}}$	sensor vector cartesian coordinates x-component in body frame
$xnofly_j$	x-position of jth no-fly zone
$\mathbf{x}^*$	optimal state(s) (trajectories)
$\mathbf{x}_N$	general form of discretized, approximate solution
$\mathbf{x}_{nofly_j}$	position of jth no-fly zone
Y	aerodynamic side force
y	y-position
$y_{b_{sensor}}$	sensor vector cartesian coordinates y-component in body frame
$ynofly_j$	y-position of jth no-fly zone
$\mathbf{z}$	arbitrary matrix
z	z-position
$z_{b_{sensor}}$	sensor vector cartesian coordinates z-component in body frame

$znofly_j$	z-position of jth no-fly zone

Symbols

α	angle of attack
β	yaw angle
δ_{az}	sensor azimuth look angle
δ_{el}	sensor elevation look angle
$\dot{\alpha}$	angle of attack rate
$\dot{\sigma}$	roll angle rate
ϵ	convex combination coefficient
η	relative error
γ	climb angle
κ	multi-objective cost function normalization factor
μ	geocentric gravitational constant
ν	basis functions
ω	angular velocity of the Earth
Φ	fuel-air ratio
ϕ	longitude
ϕ	optimal control terminal cost (Mayer)

Ψ	optimal control boundary conditions
ψ	heading angle
ρ	atmospheric density
σ	roll angle
θ	latitude
φ	fuel-air equivalence ratio

— Subscripts —

$_0$	initial condition
$_f$	final condition
$_k$	index

— Superscripts —

^{T}	transpose
^{lower}	lower limit
^{upper}	upper limit

— Acronyms —

3DOF	Three Degrees of Freedom
4DOF	Four Degrees of Freedom

6DOF	Six Degrees of Freedom
A2/AD	Anti-Access/Area Denial
AFIT	Air Force Institute of Technology
AFRL	Air Force Research Laboratory
AIAA	American Institute of Aeronautics and Astronautics
ASCII	American Standard Code for Information Interchange
CAV	Common Aero Vehicle
CONUS	Continental United States
EO	Electro-Optical
EXITS	Explicit Interactive Thermal Structures Code
FOR	Field of Regard
FOV	Field of View
GHAME	Generic Hypersonic Aerodynamic Model Example
GNC	Guidance, Navigation, and Control
GPM	Gauss Pseudospectral Method
GPOPS	General Pseudospectral Optimal Control Software
GUI	Graphical User Interface
HAST	Hypersonic Aerospace Sensor Technology
HAVOC	Hypersonic Vehicle Optimization Code

HBVP	Hamiltonian Boundary-Value Problem
IPOPT	Interior Point OPTimizer
ISR	Intelligence, Surveillance, and Reconnaissance
kts	knots (nautical miles per hour)
LANMIN	Langley MINIVER Code
LG	Legendre-Gauss
LGL	Legendre-Gauss-Lobatto
LGR	Legendre-Gauss-Radau
LOS	Line-of-Sight
LQR	Linear Quadratic Regulator
LWIR	Long-wave Infrared
MATLAB	Matrix Laboratory
MINIVER	Miniature Version
MOO	Multi-objective Optimization
mph	miles per hour
MWR	Method of Weighted Residuals
NASA	National Aeronautics and Space Administration
NASP	National Aerospace Plane
NLP	Non-Linear Programming

OCP	Optimal Control Problem
OTIS	Optimal Trajectories by Implicit Simulation
POST-II	Program to Optimize Simulated Trajectories-II
PS	Pseudospectral
psf	pounds per square foot
RASAC	Rapid Access-to-Space Analysis Code
RLV	Reusable Launch Vehicle
SNOPT	Sparse Nonlinear Optimizer
SNR	Signal to Noise Ratio
SQP	Sequential Quadratic Programming
SSTO	Single Stage to Orbit
TBCC	Turbine Based Combined Cycle
TPS	Thermal Protection System
UAV	Unmanned Aerial Vehicle
WFE	Wavefront Error
WPAFB	Wright-Patterson Air Force Base

MULTI-OBJECTIVE TRAJECTORY OPTIMIZATION OF A HYPERSONIC RECONNAISSANCE VEHICLE WITH TEMPERATURE CONSTRAINTS

I. Introduction

> Hypersonic technology allows the Air Force to respond extremely quickly to far away contingencies and potentially to out run any defense when it arrives in a war zone. The Air Force ... clearly believes it could have a major impact on military operations if it becomes affordable [77].

The United States Air Force has recently changed their perspective on hypersonic vehicles. The Air Force had previously embarked on technology development efforts to advance the state-of-the-art in hypersonic vehicle technology but there has not been a strong linkage to operational requirements nor has there been a strong operational push from warfighters. Evolution of Air Force Vision and Doctrine is now driving the need for systems that deliver weapons very quickly, provide very fast turnaround intelligence products or provide on-demand access to space [78].

1.1 Motivation

To implement the Air Force Vision of *Global Vigilance, Reach and Power*, the Air Force focuses on six distinctive capabilities [2].

- Air and Space Superiority
- Global Attack
- Rapid Global Mobility
- Precision Engagement

- Information Superiority
- Agile Combat Support

Of these capabilities, four can be enabled or enhanced by hypersonic technologies: Air and Space Superiority, Global Attack, Precision Engagement, and Information Superiority.

All of these capabilities are implemented as Air Force core functions, which are directly related to the Air Force's core duties and responsibilities [22]. These core functions are:

- Nuclear Deterrence Operations
- Air Superiority
- Space Superiority
- Cyberspace Superiority
- Command and Control
- Global Integrated Intelligence, Surveillance, and Reconnaissance (ISR)
- Global Precision Attack
- Special Operations
- Rapid Global Mobility
- Personnel Recovery
- Agile Combat Support
- Building Partnerships

Of these core functions, four can be enabled or enhanced by hypersonic technologies: Nuclear Deterrence Operations, Space Superiority, Global Integrated ISR, and Global Precision Attack.

Hypersonic systems can contribute to meeting these core functions by two primary attributes: responsiveness and survivability. Given the ability to operate at

high Mach numbers, a hypersonic reconnaissance vehicle can respond very quickly to collect sensor data against time-sensitive targets. Unlike satellites, whose orbits are predictable, or orbit changes are restricted by limited fuel capacity, a hypersonic reconnaissance vehicle has great flexibility and unpredictability. This ability to operate at high speeds, as well as high altitudes, also gives a hypersonic reconnaissance vehicle inherent survivability against aggressor weapon systems. These attributes allow a hypersonic reconnaissance vehicle to see threats from significant distances and then use, at a minimum, a passive defense strategy of either out-running or out-maneuvering threats.

Over the last decade, the United States has been involved in two regional conflicts. While these conflicts did not involve near-peers like Russia or China, the United States could have benefited by having weapon systems that were very responsive and could deliver weapon effects very quickly. Having a weapon system that operates at high Mach numbers allows decision makers to attack targets on very short notice with little lag between mission approval and delivering weapons on target.

> On August 20th 1998 [President] Clinton ordered American warships in the Arabian Sea to fire a volley of more than 60 Tomahawk cruise missiles at suspected terrorist training camps near the town of Khost in eastern Afghanistan. The missiles, flying north at about 880kph (550mph), took two hours to reach their target. Several people were killed, but the main target of the attack, Osama bin Laden, left the area shortly before the missiles struck. American spies located the al-Qaeda leader on two other occasions as he moved around Afghanistan in September 2000. But the United States had no weapons able to reach him fast enough [1].

In order to develop and implement these advanced hypersonic systems, tools will be needed in order to do both the upfront systems engineering as well as the operational mission planning. Systems engineering trade study tools will be needed throughout the acquisition process in order to assess and evaluate system concepts to study the vehicle tradespace. These tools need to be flexible and responsive so

that they can quickly evaluate many different system parameters, such as vehicle properties and mission scenarios. One of these needed tools will be the capability to generate optimized trajectories for selected mission parameters. This capability will need to be able to develop optimal trajectories that meet specified mission objectives and constraints, such as mission success criteria and path constraints. Similar tools will have to be developed to allow mission planners the ability to develop initial mission profiles to meet mission taskings. These mission planners will have to be able to consider the same objectives and constraints as the systems engineers.

As will be discussed in latter sections, there has been little research to incorporate the thermal operating constraints of hypersonic vehicles. Most of the research has focused on using simpler constraints such as heat rate or heat load, but this does not take into account the performance of the vehicle's thermal protection system. Useful optimal trajectories will have to incorporate predicted operational structural temperatures to ensure that the vehicle can be operated in its intended flight and temperature regime.

1.2 Historical Perspectives

While the term hypersonic was not coined until 1949 by H. S. Tsien [53], interest in high-speed flight goes back to the early 20th century [39]. Prior to World War II, aerodynamicists and propulsion developers in several European countries developed concepts and patents for using ramjets for supersonic flight. Not until 1949, with a flight of a two-stage Bumper-WAC research rocket, did a vehicle achieve hypersonic flight [78].

Starting with the German Sānger-Bredt Silbervōgel antipodal aircraft design in the 1930s, advanced vehicle designs were developed for high-speed flight using ramjets, scramjets, or rockets for propulsion [78]. Several other systems, such as the Convair

Space Plane and the Dyna-Soar hypersonic boost-glide vehicle actually started development but were never completed due to technical or programmatic issues [78] [53].

Starting in the 1950s, ramjets were used in development of tactical missiles such as the Bomarc air defense missile, the Talos naval surface-to-air missile, several Soviet surface-to-air missiles, and the French ASMP air-to-surface missile. Each of these systems operated in the Mach 2 to 3 range, where ramjets become efficient propulsion systems [30]. The United States has not used ramjets in weapon systems due to a lack of unique military need since less capable systems have been able to meet military needs at a lower cost than ramjets [78].

While development of these ramjet-based systems provided some operational experience in high-speed flight, they did not address many of the issues with hypersonic flight since ramjet and scramjets operate in different but overlapping altitude/velocity regimes. Figure 1 shows that ramjets operate most efficiently (as measured by specific impulse, I_{sp}) between Mach 2 and Mach 6 [48], depending on internal engine design configuration. Above Mach 6, ramjet internal temperatures become higher than what current materials can tolerate [39]. While scramjets can operate starting around Mach 4, they are operated most efficiently above Mach 8, depending on internal engine design configuration and fuel used.

The United States' first significant experience with hypersonic flight came with the North American X-15 vehicle, which operated at velocities up to Mach 6.7. It was the first airplane to operate at hypersonic speeds and was designed to survive the stressing thermal conditions in hypersonic flight. While this aircraft provided experience with hypersonic flight, it still used a traditional rocket for propulsion [78].

The first advanced system development that included a hypersonic air-breathing engine was the X-30 demonstrator project for the National Aerospace Plane (NASP) program. This program was intended to be a Single Stage to Orbit (SSTO) vehicle

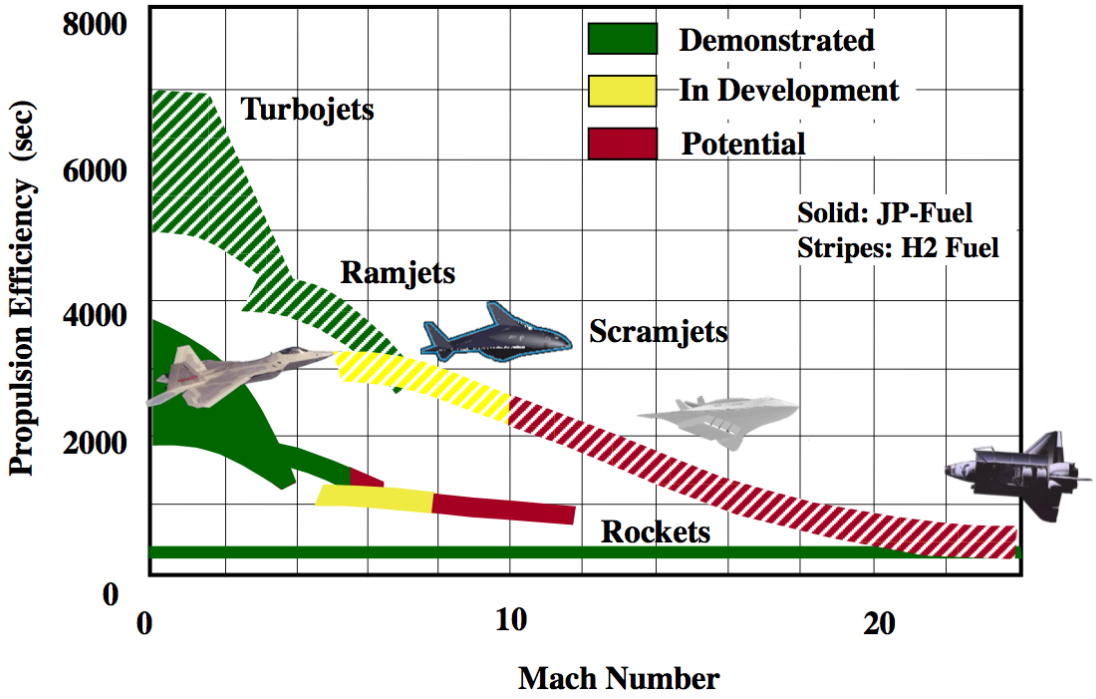


Figure 1. Air-breathing Propulsion Flight Regimes [50]

enabling routine space access. This vehicle used a waverider design and hydrogen fuel for its scramjet engine. Due to ambitious requirements that were not achievable with then state-of-the-art technologies, the program was cancelled, but it drove the next generation of hypersonic test vehicles [78].

Early in the 2000s, the National Aeronautics and Space Administration (NASA) developed the Hyper-X program to demonstrate scramjet technologies. Specifically, the X-43A research program was developed to demonstrate, validate and advance technology for hypersonic aircraft powered by an airframe-integrated scramjet engine using hydrogen for fuel. The X-43 was the first successful scramjet to operate at hypersonic speeds, having two successful flights. The vehicle attained a velocity near Mach 10 on its second flight [48].

To operate a scramjet-based vehicle that was more operationally usable and flexible, the scramjet would need to use hydrocarbon-based fuel, instead of the unwieldy

hydrogen fuel used by the X-43A. To demonstrate hypersonic flight using hydrocarbon fuel, the X-51 Waverider program was developed. Out of four flight tests, two were successful, resulting in flights with Mach 5+ velocities but more importantly sustaining the longest air-breathing hypersonic flights to date.



Figure 2. X-51 Waverider Separating from Booster [55]

Operating at hypersonic velocities results in a very stressing environment. Designing a system to operate in this regime drives several technologies: high-speed aerodynamics and thermodynamics analysis; propulsion combustor operations (mixing, fluid/thermal dynamics); use of hydrocarbon and hydrogen fuels at high temperatures; propulsion/airframe integration; materials and thermal protection systems designed to operate at high temperatures; and high-speed ground-test facilities. In addition to significant technical challenges, operational issues would have to be resolved in any hypersonic system development. These issues, which sometimes conflict

with performance and design goals, include volume constraints, long-term and low maintenance storage, safe handling characteristics, and compatibility with fuels used in existing operational systems. An operational hypersonic air-breathing system will have to incorporate design features to successfully resolve all these issues.

In developing hypersonic systems, two other challenges need to be addressed. Since scramjets need to start operations at moderately high Mach numbers, an additional propulsion system needs to be integrated to get the vehicle to a high enough Mach number to start the scramjet. This adds to vehicle size, complexity and weight. In addition, since operating at high Mach numbers can create high-velocity and high-temperature effects in the flow-field surrounding the vehicle, the performance of electromagnetic communications and sensor systems can be impacted. They would have to be designed and integrated such that they can operate while accommodating the vehicle's stressing operational environment.

While the recent technological advances have been moderately significant, there still needs to be significant technology investments before a hypersonic air-breathing vehicle can become operational. Along with the core technologies (propulsion, materials, aerodynamics), systems engineering and operational tools will have to be enhanced to support development of hypersonic systems. These tools need to be able to quickly evaluate the performance of a hypersonic system considering candidate scenarios with their specified constraints and objectives.

Over the last 30 years, several purpose specific trajectory optimization tools have been developed for both terrestrial vehicles and spacecraft. NASA has developed both Program to Optimize Simulated Trajectories-II (POST-II) and Optimal Trajectories by Implicit Simulation (OTIS) trajectory optimization tools for use in their vehicle development programs. While these tools have been successfully used on programs such as the Space Shuttle and many interplanetary probes, they require a significant

learning curve to use and have limitations in key areas such as setting path constraints. In addition, neither of these tools incorporate constraints from aerothermal effects on aircraft as well as sensor parameters. Georgia Tech has developed Rapid Access-to-Space Analysis Code (RASAC) which incorporates aerothermal effects but RASAC does not generate optimal trajectories and is not actively used in research today [59]. There has been a recent trend to use generalized optimal control software incorporating pseudospectral techniques. While successful, published research using these tools was focused on specific scenarios and did not provide a complete tool for systems engineers and mission planners to use in rapid trade studies and mission planning.

1.3 Problem Description

The mission objective of a hypersonic reconnaissance system is to fly a reconnaissance mission over contested or denied regions, from an initial point to a final point to collect sensor data against specified collection targets. Mission objectives are specified, such as minimizing a system parameter or a combination of multiple parameters, such as time of flight, fuel expended, or even a sensor parameter such as sensor collection time.

Along a trajectory, waypoints can be defined such that the vehicle must pass thru a specified location and even possibly at a specified time. Alternately, these waypoints can be defined as tanker rendezvous criteria where only the rendezvous altitude and velocity are defined. Since these waypoints specify fixed parameters in the trajectory, they can be used to break up the trajectory into multiple segments called phases.

In addition to the specified aerodynamic and propulsion dynamics, aerothermal constraints will have to be included. They can be specified as heat rates, heat loads, or vehicle temperature constraints. These temperature constraints are the only means

to take into account performance of the vehicle's thermal protection system.

Additional constraints can be imposed as well. No-fly zones can be specified to avoid locations or areas to increase survivability, meet mission objectives, or to avoid prohibited airspace. A minimum level of survivability can also be specified. In addition, constraints related to vehicle performance, such as maximum allowable dynamic pressure, can be imposed as well as constraints related to data sensor performance, such as sensor resolution.

1.3.1 Hypersonic Reconnaissance Vehicle Overview.

The hypersonic reconnaissance vehicle considered in this research is a hypothetical design intended to conduct reconnaissance operations in contested or denied operations in the future. Key attributes of this research's conceptual vehicle are listed below.

- Conduct reconnaissance missions up to Mach 8
- Turbine Based Combined Cycle (TBCC) design
- 300,000 lb gross weight
- Regional to intercontinental range
- Multiple integrated Electro-Optical (EO) sensors for the reconnaissance mission
- Operations in Anti-Access/Area Denial (A2/AD) environments
- Hydrogen propellant
- Thermal protection systems
- Warm or hot aircraft structure (temperatures higher than the melting point of aluminum and steel and possibly requiring advanced carbon-based materials)

1.3.2 Mission Scenarios.

In this research, several different scenarios will be considered and evaluated. Each of these scenarios are driven by the mission objectives, which drives the selection

of the mission parameters to be optimized. In each of these scenarios, the problem constraints described previously are selected to match the desired mission parameters.

- Minimum transit time: In this scenario, the vehicle's trajectory is optimized to minimize the time taken to go from an initial to a final point.
- Minimum transit time with control penalty: In this scenario, the vehicle's trajectory is optimized to minimize the time taken to go from an initial to a final point, but the use of controls are penalized by incorporating them along with the transit time in the objective cost function.
- Maximum range: In this scenario, the vehicle's trajectory is optimized to maximize the distance flown from an initial position.

The previous three scenarios could be combined to develop a single optimization parameter. In order to develop a scalar value to be optimized, preferences for each part of the hybrid elements have to be articulated and included in the optimization calculations. An example scenario (Figure 3) that could be evaluated would include the following attributes as shown in Figure 4.

- Horizontal takeoff from Continental United States (CONUS) or in-theater base
- Climb to cruise altitude
- High-speed cruise to transit to target area
- Descend to tanker-operations altitude
- Refuel from tanker
- Climb to cruise altitude
- High-speed cruise to ingress to target location(s)
- High-speed dash and conduct reconnaissance sensor
- High-speed cruise to egress from target area
- Descend to tanker-operations altitude
- Refuel from tanker
- Climb to cruise altitude
- High-speed cruise to return to base
- Descend and land at base

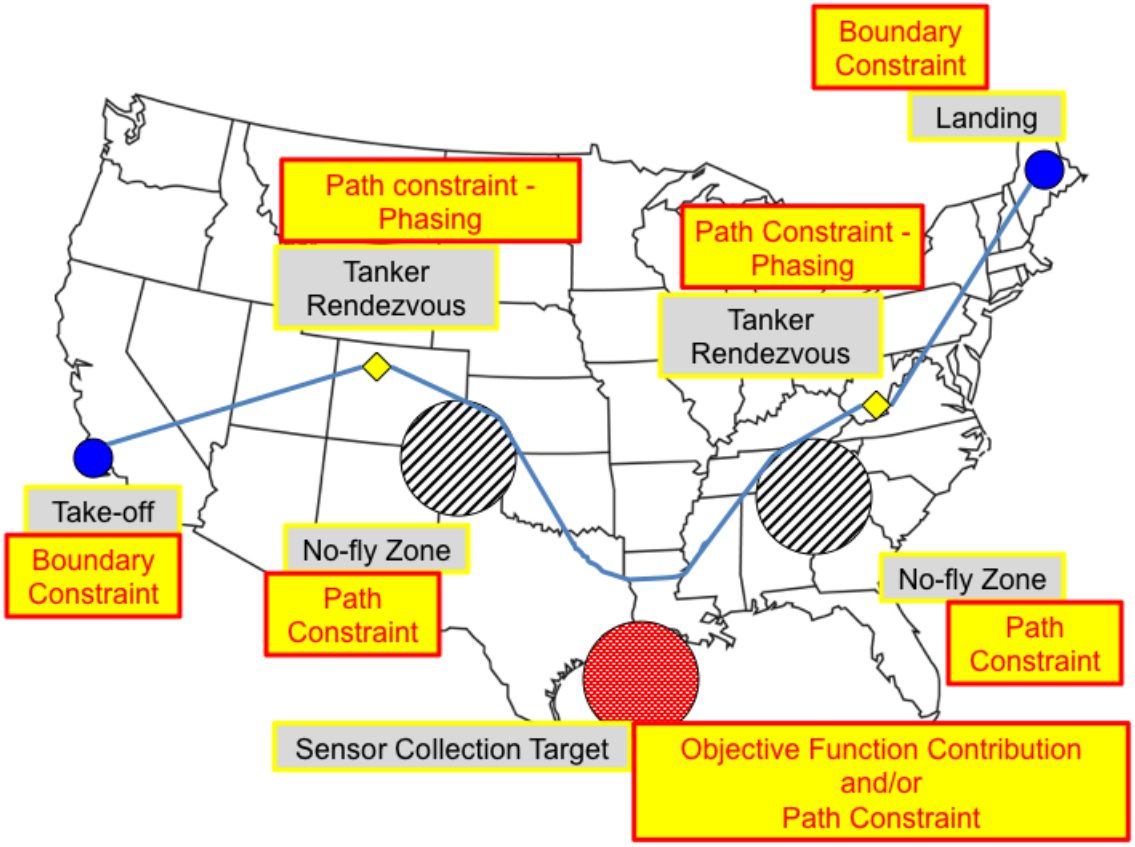


Figure 3. Notional Mission Trajectory

1 Takeoff
2 Climb to cruise altitude
3 High-speed cruise
4 Descend for refueling
5 Tanker refueling
6 Climb to cruise altitude
7 High-speed cruise (ingress to target area)
8 Conduct (high-speed) sensor operations
9 High-speed cruise (egress from target area)
10 Descend for refueling
11 Tanker refueling
12 Climb to cruise altitude
13 High-speed cruise
14 Descend for landing
15 Land
16 Reserves

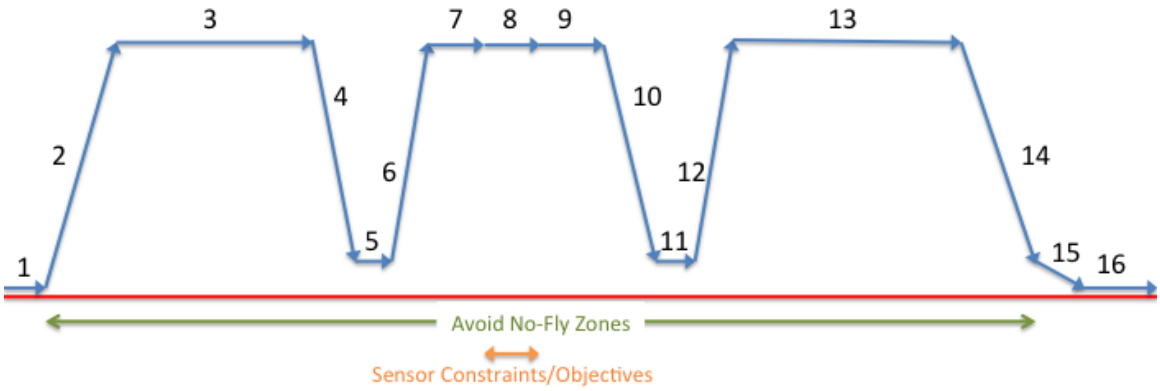


Figure 4. Notional Mission Profile

1.3.3 Problem Statement.

Currently, there is no capability to do rapid[1] optimal trajectory calculations for mission planning and systems engineering trade studies of hypersonic systems that incorporate the ability to easily vary objective functions and constraints (e.g., waypoints, no-fly zones, aerothermal constraints, sensor parameters), to include operational temperature constraints. The methodologies currently in use only use simplistic models for aerothermal constraints and do not incorporate sensor parameters. The current methodologies are also numerically inefficient. A trajectory optimization approach, or a general optimal control software approach, that is computationally efficient and versatile, while based on a robust mathematical foundation, would provide significant advantages over many other existing trajectory optimization tools [63].

1.4 Research Objectives

The goal of this research effort is to develop a methodology and tool for determining optimal trajectories for hypersonic vehicles that uses an optimal control formulation with several path constraints, including vehicle temperature. In addition to developing this methodology, this research effort includes a parametric analysis to examine trends and performance attributes for a specific scenario that could simplify or accelerate trajectory generation.

This research is unique in that it integrates the use of a hypersonic vehicle model, a sensor model, and an aerothermal effects model. This technique will allow hypersonic researchers to quickly generate optimal flight profiles that can be used in trade studies and systems engineering architecture assessments. In addition, by using insight from this parametric analysis, acceptable near-optimal trajectories can be generated quicker than using the full methodology including temperature con-

[1]For this purposes of this research, "rapid" is considered thirty minutes or less.

straints. Researchers will have the ability to incorporate flight path characteristics such as temperature constraints, originating/terminating locations, waypoints, keep out zones, and varied sensor targets. Researchers will also have the ability to apply priorities to mission objectives, such as minimum time, minimum fuel, time on target, or optimal sensor product figures of merit such as ground sample distance.

1.4.1 Research Assumptions.

In this research, an optimal control formulation is used as a general construct tailored for several different implementations: minimum time trajectory between two fixed points, minimum time trajectory between two fixed points with control penalty, and maximum flight range. The optimal control formulation will use system dynamics that are a simplification of AFRL's three degrees of freedom (3DOF) model, removing the control laws for vehicle control surfaces [82]. In order to incorporate aerodynamic and thrust forces in the vehicle equations of motion, an empirical data set called Generic Hypersonic Aerodynamic Model Example (GHAME) will be used [84]. While widely used in modeling the dynamics of hypersonic vehicles, this data set is based on a limited set of data points for the modeled aerodynamic and thrust coefficients. Using this set of discrete data points can introduce discontinuities in the equation of motion calculations, resulting in difficulties for the tool's Non-Linear Programming (NLP) solver to find optimal solutions. To reduce the chance of this happening, the GHAME data set will be interpolated using cubic splines to ensure second derivative continuity.

The aerothermal model will use NASA's MINIVER (Miniature Version) aeroheating software. While it provides moderate-fidelity aerothermal results, it has a limited capability to model aerothermal effects for complex geometric shapes, so in this research, a simple wedge will be used to determine the time-dependent skin

temperatures at several points along the windward side of the wedge.

The sensor model will just include geometric parameters for the line-of-sight between the sensor(s) and the observation targets, as well as sensor look angles to the target. Since there is limited available research on the modeling of thermal and flow-field effects on EO signal propagation, these effects are not included in this research. As a result, the resulting trajectory constraints from the sensor modeling will probably not be as severe as if the flow-field effects were modeled.

Even if all these methods are successful, the tool can be given a set of parameter for which there is no feasible solution in the allowable solution space, resulting in an over-constrained problem. While this cannot be easily predicted a priori, analysis of the outputs from the tool's NLP solver can be used to determine if the problem is over constrained or if the problem is infeasible due to the given initial conditions.

1.5 Dissertation Overview

This dissertation contains seven chapters. This first chapter introduced the motivation, background, and objectives for this research. Chapter II provides a review of the previous techniques used in hypersonic vehicle modeling and trajectory calculation. Chapter III describes the models used in the research and how they were adapted to work within the mathematical framework used in the research. Chapter IV introduces the mathematical tools and formulations used and provides a framework for answering the research questions. Chapter V details results of the research as scenario-based optimal trajectories. Chapter VI discusses the result of a parametric analysis for a specific scenario across a range of scenario inputs. Finally, Chapter VII summarizes the conclusions and contributions from this research, and suggests possible future research.

II. Literature Review

THE focus of this research is to develop a methodology and tool for determining temperature-constrained optimal trajectories for scramjet-based hypersonic surveillance vehicles that uses an optimal control formulation and solves it using a variable order Gauss-Radau quadrature collocation method with a Non-Linear Programming (NLP) solver, that could include waypoints, no-fly zones, thermal constraints, and sensor constraints. Developing this integrated system drives the fusion of several different disciplines, to include trajectory optimization, vehicle modeling, aerothermal modeling, optimal control theory, pseudospectral (PS) methods, and multi-objective optimization methods.

2.1 Trajectory Optimization

Trajectory optimization techniques have been developed steadily since Robert Goddard posed the first aerospace optimal control problem in 1919 and its first significant use in the Gemini and Apollo space programs [14]. With the advances in optimal control and numerical computations, trajectory optimization techniques have advanced and been refined to the point that there are many dedicated trajectory optimization software packages, such as Optimal Trajectories by Implicit Simulation (OTIS), developed by the National Aeronautics and Space Administration (NASA) and Boeing, and Program to Optimize Simulated Trajectories (POST-II), developed by NASA and Lockheed-Martin [57]. Typical dedicated trajectory optimization software packages are generalized point mass, discrete parameter targeting and optimization programs. Since these program can break up the problem into multiple phases, they can analyze complex events during trajectories, such as powered and unpowered periods, staging events, and even vehicle reconfigurations or separation into several

vehicles [57].

While OTIS and POST-II are utilized throughout the aerospace industry, they both have significant limitations. POST-II uses only direct shooting methods which can limit its applicability. While OTIS can be used with either direct shooting methods or low-order direct collocation methods, the software does not automatically check to see if all the specified constraints are met at the end of the optimization [57]. Other researchers have refined and applied direct shooting methods to hypersonic optimal trajectories, but retain limitations such as simple lookup tables for aerodynamic forces or just model unpowered flight [25]. Other researchers have simplified the problem, such as assuming a constant air mass flow and limiting the number of independent variables [42]. Finally, genetic algorithms have been applied to develop hypersonic optimal trajectories, albeit for simplified problems [91].

A trajectory optimization approach, or a general optimal control software approach, that is computationally efficient and versatile, while based on a robust mathematical foundation, would provide significant advantages over many other existing trajectory optimization tools [63]. Several researchers such as Murillo [53], Shi [73], Wu [72] and Song [74] have used a direct collocation approach using a Legendre-Gauss-Radau collocation method to develop optimal trajectories for hypersonic vehicles, but with significant simplifications or fairly narrow conditions, such as looking only at unpowered reentry trajectories or just the cruise phase of a hypersonic flight profile.

Since trajectory optimization is a specific application of optimal control theory, the formulation for a trajectory optimization problem is the same for a general optimal control problem, as shown in Chapter IV. This general formulation is also used in general purpose optimal control software, e.g. DIDO [61] (named after Queen Dido of Carthage, who is known for her solution to an optimal control problem in antiquity)

or General Pseudospectral Optimal Control Software (GPOPS) [60], so the same formulations can also be used to compute vehicle optimal trajectories.

2.2 Vehicle Models

Hypersonic vehicle models have been developed using many different approaches. Specifically, these methods model both the aerodynamic (typically lift coefficient, C_L and drag coefficient, C_D) and propulsion coefficients (typically specific impulse, I_{sp} and possibly inlet capture area ratio, A_0/A_c) used in the equations of motion or similar approaches to calculate vehicle trajectory states. Depending on other model attributes and the desired model fidelity, three degrees of freedom (3DOF) (three components of translation, angular controls), four degrees of freedom (4DOF) (three components of translation plus angular rate controls), and six degrees of freedom (6DOF) (three components of translation plus three components of rotation) are commonly used.

Several researchers have used coefficients from flight or development hardware for their performance analysis. For example, Allwine [5] used the simulation environment for the X-37 test vehicle to calculate aerodynamic and propulsion coefficients. Bollino [12] [11] and Shaffer [70] [69] used the aerodynamic and propulsion design parameters for the X-33 Reusable Launch Vehicle (RLV).

Another approach used for vehicle modeling is the development of a customized model or use of an in-house vehicle synthesis tool to calculate the needed aerodynamic and propulsion coefficients. Bolender and Doman [10], Frendreis and Skujins and Cesnik [29], Mirmirani and el al. [51], Zhou [90], and Windhorst and Ardema and Bowles [86] [85] have successfully used these tools as either an interactive means to calculate coefficients on the fly or to develop coefficient look-up tables or coefficient curve-fitted equations used in their trajectory analysis.

The most common source of aerodynamic and propulsion coefficients for an air-breathing hypersonic vehicle is the Generic Hypersonic Aerodynamic Model Example (GHAME) data set [84]. This data set was derived from the Space Shuttle and lifting body flight data and developed in the 1980s to provide a common set of aerodynamic, aerothermal, and propulsion data for use in analyzing Single Stage to Orbit (SSTO) air-breathing hypersonic vehicles. It has frequently been used in hypersonic vehicle trajectory optimization. Murillo used a GHAME data set to develop minimum fuel ascent trajectories for SSTO hypersonic air-breathing vehicles, using both open-loop and closed-loop guidance approaches with finite difference methods [53]. Araki used a GHAME data set to investigate reentry dynamics and handling qualities of hypersonic vehicles using perturbation techniques [6]. Janicki used GHAME to develop analytical approximations for the solutions to the vehicle's equations of motion [40]. Other researchers such as Van Buren and Mease [16], Yu and Nai-gang [89], and Dewell et al. [23] also used GHAME to develop optimal trajectories for SSTO ascent, hypersonic boost-glide, and fuel-optimal cruise flight profiles.

For this research, Air Force Research Laboratory's (AFRL) general purpose 3DOF model with the GHAME data set was used, since both of these have been used in past research at AFRL and are fully releasable [92].

2.3 Aerothermal Models

Aerothermal modeling is defined as the modeling of the heating of surfaces produced by passage of gases over their surface, primarily caused by friction and by fluid compression effects. This heating occurs in the hypersonic boundary layer, where the velocity of the fluid is reduced to zero at the vehicle surface, resulting in high heat rates (flux) that are much higher than in the surrounding environment [65].

A moderate fidelity aerothermal model would calculate surface Thermal Protec-

tion System (TPS) temperatures at different points along the vehicle surface by incorporating the following heat transfer mechanisms [87] as shown in Figure 5, with the terms shown in the figure:

- Gas conduction from moving fluid (q_c)
- Gas radiation from moving fluid (q_{rad})
- Solid conduction into vehicle surface (TPS) (q_{cond})
- Solid radiation from vehicle surface (TPS) (q_{rerad})
- Ablation of vehicle surface (TPS) (q_{mdot})

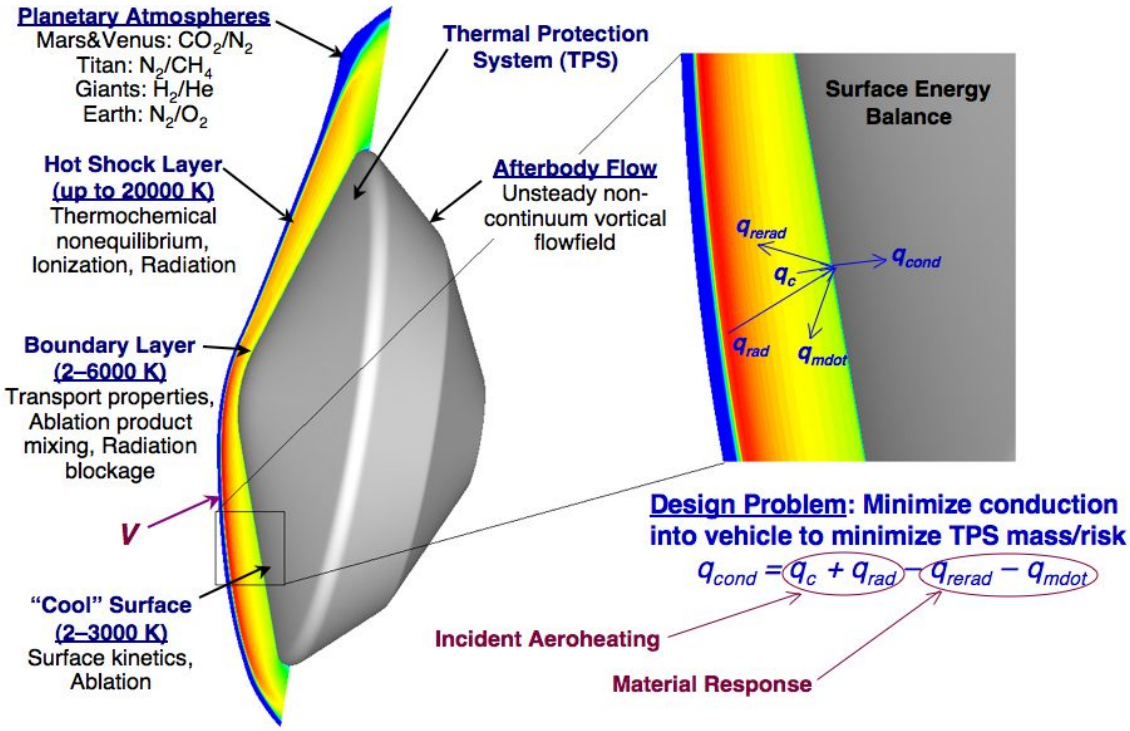


Figure 5. Principles of Aerothermal Modeling [87]

Most of the related research use a low fidelity aerothermal model since they only calculate the heat fluxes or heat loads at a vehicle leading edge where the flow stagnates. This condition can be approximated by using Chapman's equation, where the heat rate is proportional to powers of the ambient environment gas density and the vehicle velocity, given in Equation 1 [17]. Chapman's equation uses several aerothermal

simplifications and experimental data to derive the equation constants[1] and powers. While it was developed to approximate the heat flux for a planetary reentry vehicle, it also provides an acceptable approximation for the stagnation heat flux for an air-breathing hypersonic vehicle. Several researchers, such as Bollino [11], Grant [36], Dong [25], Shi [73], Van Buren [16], Zhou [91], Al-Garni [4], Jansch [60], and Mooij [52] have used Chapman's approximation or its integrated form (heat load) (Equation 2) as trajectory optimization path constraints. Instead of using Chapman's equation, Windhorst, Ardema, and Bowles have used an in-house design tool called Hypersonic Vehicle Optimization Code (HAVOC) to compute heat fluxes and loads [86] [85], with $\dot{Q}_{conv}$ as the heat flux, Q_{conv} as the heat load, ρ as the atmospheric density, V as the vehicle velocity, t_0 as the initial time, t_f as the final time.

$$\dot{Q}_{conv} \propto \sqrt{\rho} V^3 \tag{1}$$

$$Q_{conv} = \int_{t_0}^{t_f} \dot{Q}_{conv} dt \tag{2}$$

Chudej, Pesch, Wachter, Sachs, and Dinkelmann have written several papers [18] [67] [24] [81], as well as Dinkelmann's PhD dissertation, that use their mathematical model to describe the unsteady heat transfer effects in a TPS. They use their model to calculate heat load along a trajectory or surface temperatures using most of the heat transfer mechanisms described above.

Another method to calculate surface temperatures is to use NASA's MINIVER (Miniature Version) aerothermal code. It is a NASA-developed conceptual/preliminary design tool used for aerothermal prediction analysis to model aerodynamic heating effects for vehicle-specific flow fields. It has been used extensively for analysis

[1]Researchers often state the vehicle specific constants without any derivation. Nizami provides a summary of a derivation for a generic vehicle [58].

and design of thermal protection systems, from the Space Shuttle to many of the X-Vehicles and also advanced hypersonic and SSTO concepts. It models post-shock and local flow properties as well angle of attack effects [88]. MINIVER has been folded into an integrated TPS analysis capability, integrated with other analysis tools such as trajectory development and TPS design tools. Use of these tools in an integrated environment can result in a quick parametric analysis of TPS designs [88].

Since MINIVER is based on legacy FORTRAN 66 code, it has not been used often as an on-call subroutine. Ordaz integrated MINIVER with MATLAB to evaluate thermal management systems for hypersonic vehicles to develop a conceptual design methodology for the evaluation of thermal management systems on air-breathing hypersonic vehicles [59]. Louderback ported the first half of the MINIVER code (LANMIN (Langley MINIVER Code), which calculates the flowfield for a specified trajectory) to `C#`, but he implemented the updated code thru a Graphical User Interface (GUI) and not via a command-line so it would not be usable as a subroutine for another program such as a MATLAB script [46].

2.4 Optimal Control Theory

The birth of optimal control occurred in the late 17^{th} Century with a problem posed by Johann Bernoulli. "What is the shape of a wire such that a bead sliding along it travels the distance between bead's endpoints in minimum time?" Solutions were developed by some of the greatest mathematical minds in history, including both Bernoulli brothers, Leibniz, and Newton. Their work on this problem also gave birth to the Calculus of Variations [76] [15]. In more modern times, both Pontryagin's Maximum Principle and Bellman's Dynamic Programming gave engineers advanced tools to more fully develop optimal control concepts and tools [76] [15].

Simply stated, optimal control theory deals with the problem of finding a set of

control inputs for a nonlinear dynamic system which result in system states that minimize (or maximize) a specified performance measure while meeting specified system dynamics and staying within specified constraints and boundary conditions. A more formal definition is given as:

> Find an admissible control $\mathbf{u}^* \in U$ that causes the system state dynamics $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t)$ to follow an admissible trajectory $\mathbf{x}^* \in X$ that minimizes the performance measure J. $\mathbf{x}^*$ is called the optimal trajectory and $\mathbf{u}^*$ is called the optimal control [43].

Specifically, an optimal control problem can be formulated as:

$$\text{Minimize: } J = \mathcal{M}\left(\mathbf{x}_0, \mathbf{x}_f, t_0, t_f\right) + \int_{t_o}^{t_f} \mathcal{L}\left(\mathbf{x}(t), \mathbf{u}(t), t\right) dt \tag{3}$$

$$\begin{aligned} &\text{Subject to, } \forall t \in [t_0, t_f]: \\ &\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) \\ &\mathbf{\Psi}\left(\mathbf{x}(t_0), \mathbf{x}(t_f), t_0, t_f\right) = 0 \\ &\mathbf{C}\left(\mathbf{x}(t), \mathbf{u}(t), t\right) \leq 0 \end{aligned} \tag{4}$$

In these equations, $\mathcal{M}$ represents the terminal cost for the endpoints (also called the Mayer term), while $\mathcal{L}$ represents the running cost (also called the Lagrangian term). In addition, $\mathbf{x}$ represents the states, $\dot{\mathbf{x}}$ the state dynamics, $\mathbf{u}$ the control, $\mathbf{\Psi}$ the boundary conditions, and $\mathbf{C}$ the path constraints.

This is the formulation for a single-phase optimal control problem, where the constraints and dynamics are consistent across the admissible solution space. For control problems that have dynamics and/or controls that have different definitions across the admissible solution space, the optimal control problem will have to be broken down into a sequence of single-phase optimal control problems that still span

the entire problem being considered. This division of the optimal control problem, called phasing, will be discussed in Section 2.5.3.1.

Optimal control theory is commonly used to solve many different types of optimal trajectory problems, including application to hypersonic vehicles. Research addressing this type of problem can be categorized by either an ascending (powered) or a glide/reentry (unpowered) vehicle. The dynamics, constraints, and models used in both problem types are very similar, with the major distinction being that the ascending problem also implements a propulsion model for the thrust component.

In this research, three of the most common types of optimal control problems will be considered: minimum time, minimum time with control penalty, and maximum range. Each problem type will drive a different form of the objective functional. Chapter IV will describe the generalized form each objective functional with their simplification to relationships that will be used in the numerical solvers, with their related systems dynamics and constraints.

2.5 Pseudospectral Methods

PS techniques are a class of numerical methods developed in part to solve partial differential equations. They have been increasing in popularity as tools to solve complex optimal control problems [60].

2.5.1 Spectral Methods.

PS methods emerged from the development of spectral methods in the 1960s and 1970s. Spectral methods are themselves a class of Method of Weighted Residuals (MWR) techniques, which are used in applied mathematics and scientific computing to numerically solve ordinary/partial differential equations and eigenvalue problems involving differential equations. Several different approaches can be used in MWR:

Finite Difference, Finite Element, and Spectral Methods. All three approaches approximate the desired solution by forming a truncated series expansion of the solution, given as:

$$\mathbf{x}(t) \approx \mathbf{x}_N(t) = \sum_{k=1}^{N} \mathbf{a}_k \cdot \nu_k(t) \tag{5}$$

where $\mathbf{x}(t)$ is the exact solution, $\mathbf{x}_N(t)$ is the n^{th} order approximated solution, $\nu_k(t)$ are the basis functions for the expansion, and $\mathbf{a}_k$ are the coefficients of the basis functions. In Spectral Methods, both Finite Difference and Finite Element use basis functions (called trial functions) that are only valid in the local region of the expansion points. Since Spectral Methods use trial functions that are globally smooth over the entire problem interval, Spectral Method errors decay exponentially as the number of approximation nodes increases, which is significantly faster than the polynomial rates for Finite Difference and Finite Element methods, which have errors that decay no faster than quadratically [35].

Implementation of the Spectral Method is normally accomplished using one of three techniques: Tau method, Galerkin method or Collocation (i.e., PS) method. All three of these methods are used to develop the coefficients of the truncated series expansion of the solution (called test functions), but differ in the approach to determine the test functions. In the Tau method, test functions are selected so that the approximate solution satisfies the boundary conditions and the differential equations are enforced by requiring that approximation residuals be orthogonal to all the test functions. In the Galerkin method, the test functions are the same as trial functions and use a method similar to the Tau method in the way the differential equation is enforced. In the Collocation method, test functions are Dirac delta functions centered at a set of collocation points and the differential equation is satisfied exactly at the collocation points [7] [28].

2.5.2 Optimal Control Numerical Methods.

Numerical solution approaches for optimal control problems can be broken into either indirect or direct methods, as shown in Figure 6 [75]. In indirect methods, Calculus of Variations is applied to derive a Hamiltonian Boundary-Value Problem (HBVP), which converts the problem into a differential-algebraic system that is solved numerically, with a result that can be very accurate. While formed on a solid base of Calculus of Variations, this method can be difficult to implement since it requires derivation of first-order optimality conditions, which may not be possible to develop analytically [75]. In direct methods, states and controls are parameterized using function approximations, while the objective function is approximated using quadrature. Then the optimal control problem is transcribed to an NLP, which is solved using one of many possible software routines. While not as analytically rigorous and providing solutions that are difficult to validate for optimality, direct methods are widely used because direct methods are based on well-established numerical methods, HBVPs do not have to be derived, and optimal control problems are easier to formulate and solve [63].

2.5.3 Variable-Order Gaussian Quadrature Collocation Method.

In this research, the Variable-Order Gaussian Quadrature Collocation Method called GPOPS is used. It is a PS technique that is a direct method and uses orthogonal trial functions and collocation points (i.e., direct orthogonal collocation). This approach approximates the states and controls using a basis of Lagrange polynomials and collocates them with the state equation dynamics at the roots of Legendre polynomials (i.e. Legendre-Gauss-Radau (LGR) points). Since Legendre polynomials are orthogonal, they can be easily integrated or differentiated, using numerical methods, to determine objective function integrals or the state dynamics. The continuous-time

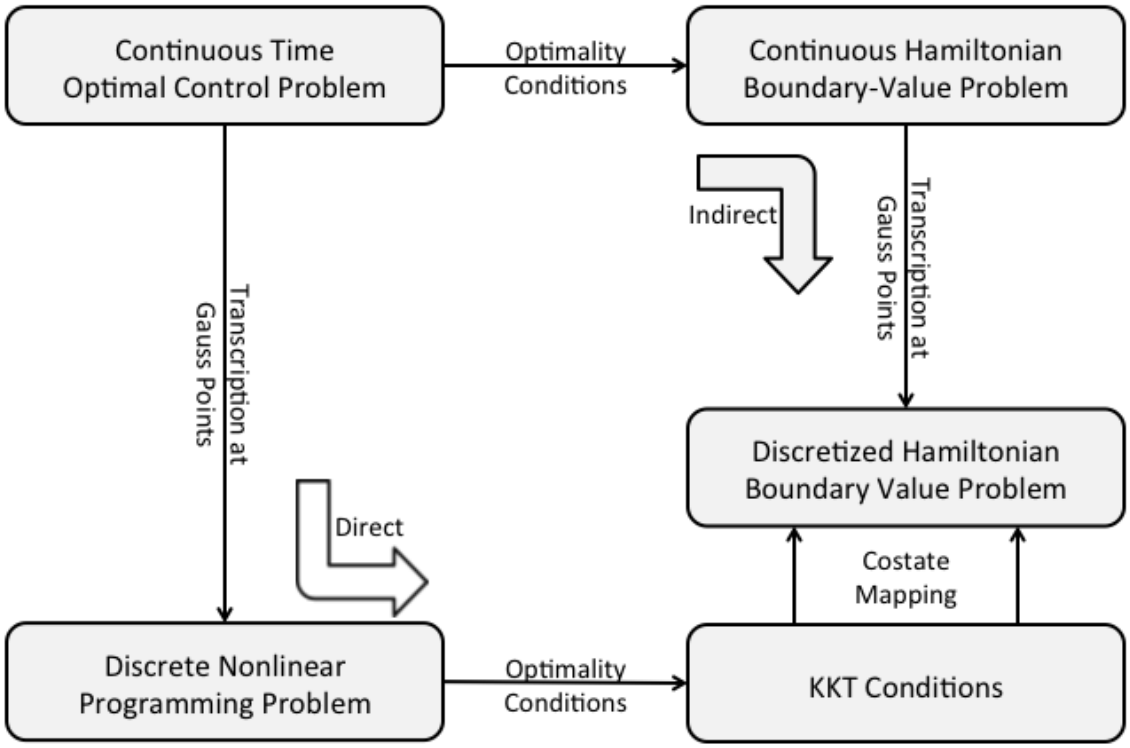


Figure 6. Indirect vs. Direct Methods [8]

optimal control problem is then transcribed to a finite-dimensional NLP and the NLP is solved using off-the-shelf NLP solvers. The GPOPS operational flow is shown in Figure 7 [60].

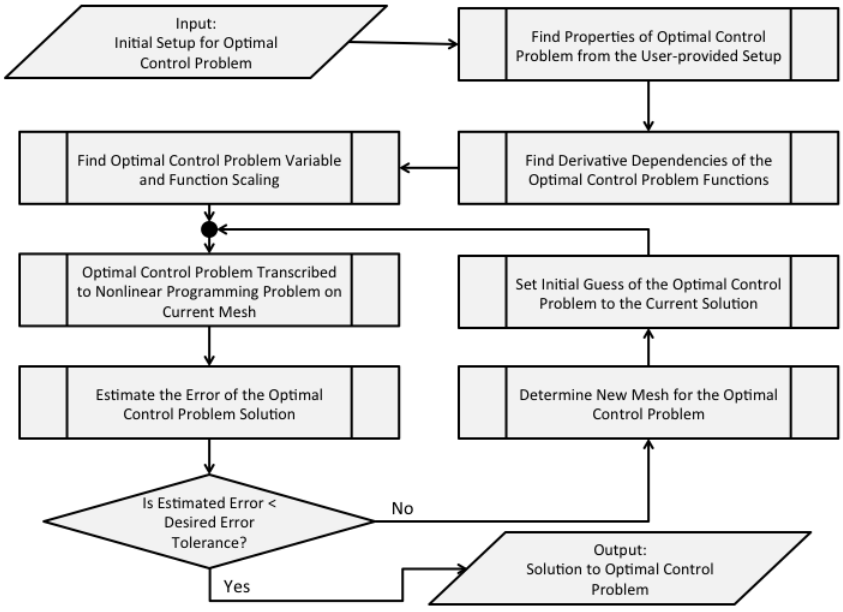


Figure 7. GPOPS Operational Flowchart [60]

As PS methods have developed, three different approaches for collocation point selection have emerged: Gauss Pseudospectral Method (GPM) (also known as Legendre-Gauss (LG)), LGR, and Legendre-Gauss-Lobatto (LGL) methods. All three of these approaches define collocation points by determining the roots of Legendre polynomials. Each of these collocation point approaches are similar in that the density of collocation points is higher at interval (each interval uses separate polynomial interpolants) endpoints, avoiding Runge's Phenomena which gives large interpolation errors when using high-order polynomial interpolants near interval endpoints [32]. The three approaches differ in how they include (if at all) interval endpoints as shown in Figure 8.

The selection of method drives two conditions: how the numerical method enforces

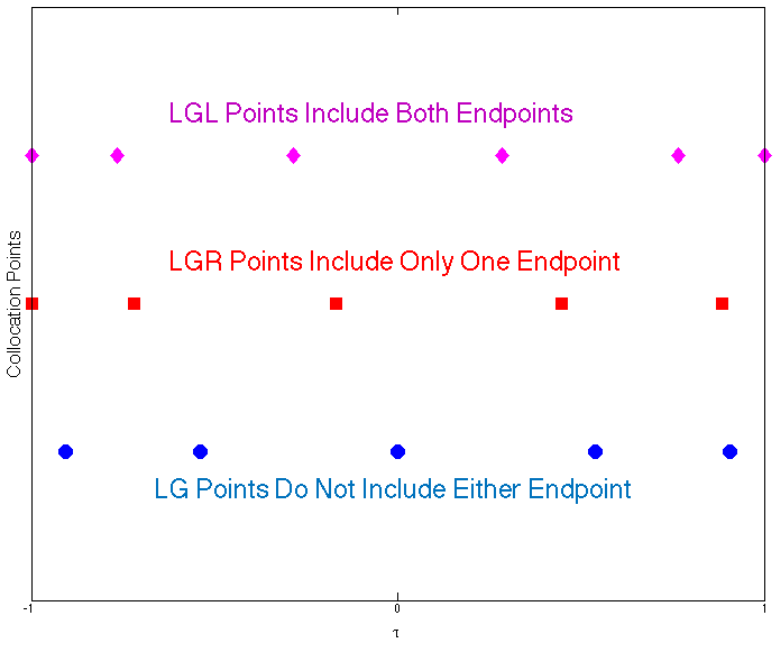


Figure 8. Pseudospectral Collocation Point Selection Methods [32]

continuity of boundary conditions and the maximum order for high-order polynomial to still obtain an exact result in Gaussian quadrature. For example, for LGR, only one endpoint is used (Gaussian quadrature is used to estimate the other endpoint) and Gaussian quadrature will provide an exact integration for a polynomial up to an order $2n - 1$ [63].

As mentioned previously, the states and controls are approximated using Lagrange interpolating polynomials. These polynomials are orthogonal can easily be differentiated and/or integrated, which will be invoked to compute state dynamics and/or objective function integrals. Since these polynomials also follow the Lagrange Isolation Property, the polynomials can be invoked at the collocation points to minimize integration errors [63].

Once the continuous-time optimal control problem is transcribed to a discrete, finite-dimensional NLP static optimization problem, it can be given to an NLP solver

for a solution computation. For example, GPOPS can invoke one of two NLP solvers: Interior Point OPTimizer (IPOPT) [41] or Sparse Nonlinear Optimizer (SNOPT) [33]. IPOPT is an interior-point line-search filter method, while SNOPT uses an Sequential Quadratic Programming (SQP) algorithm. They both use Newton methods to calculate gradients and quasi-Newton methods to estimate the Hessians. In addition, IPOPT can use second-derivative information from user-defined Hessians [80] [60].

2.5.3.1 Phasing.

One key aspect of using PS methods is their ability to be flexible to accommodate a wide variety of problem formulations. A prime example is the ability to separate an optimal control problem into phases; these are also called arcs by Betts [9] or PS knots by Ross [66]. These phases partition the problem's time domain, enabling different dynamics for portions of the trajectory. Phases also enforce state constraints at events, which are selected positions along the trajectory boundaries. Phases are implemented by enforcing continuity conditions on the states as additional path constraints [60]. For this research, phasing is a key problem attribute, since it will be used to enforce trajectory waypoints and other en-route path constraints.

2.5.3.2 Adaptive Mesh.

Originally, PS methods were broken into intervals, where each interval used separate polynomial interpolants, but have the same polynomial degree. To achieve convergence to a solution, h-methods [60] were used to increase the number of intervals over the problem's time domain, in essence increasing the "fineness" of the mesh. As PS methods developed, p-methods were also used to achieve convergence to a solution. In p-methods [60], only a single interval was used but the degree of the polynomial interpolant was increased to achieve convergence. Amongst others, Rao

and Patterson have combined and refined these methods into a *ph*-method collocation scheme that combines both interval and polynomial order refinement, in order to better match dynamics occurring within a problem as well as making the GPOPS algorithm more computationally efficient [64].

2.6 Multi-Objective Optimization Methods

Multi-objective optimization has been studied since the 19th century but did not become an active research area until the 1970s with the more widespread distribution of Pareto's work from the early 1900s [31].

Multi-objective optimization problems can be defined as [47]:

$$\begin{aligned} &\text{Minimize: } \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_k(\mathbf{x})) \\ &\text{Subject to:} \\ &g_i(\mathbf{x}) \leq 0; i = 1 \ldots m \\ &h_j(\mathbf{x}) = 0; j = 1 \ldots n \end{aligned} \tag{6}$$

where $g_i(\mathbf{x})$ and $h_j(\mathbf{x})$ are inequality and equality constraints, respectively.

The predominant aspect in multi-objective optimization is the concept of Pareto optimality, which is defined as: a point $\mathbf{x}^*$ in the feasible design space D is Pareto optimal if and only if there does not exist another point $\mathbf{x}$ in the set S such that $f(\mathbf{x}) \leq f(\mathbf{x}^*)$ with at least one $f_i(\mathbf{x}) < f_i(\mathbf{x}^*)$. Restated, there is no point that can improve an objective function without detriment to any other considered objective function [47].

There are various methods to implement multi-objective optimization. Marler and Arora [47] grouped these methods into several categories. One category, "A Priori Articulation of Preferences", works well with optimal control formulations since this

method transforms the multi-objective optimization problem into a single objective problem by using scalarization methods [19]. While these methods are fast and efficient, they converge to a local solution and do not guarantee a global optimum [47], similar to NLP methods for non-convex problems. While genetic algorithms are better suited to find global solutions, they tend to be computationally intensive [19].

2.7 Sensor Modeling

Little available research exists on the integration of EO sensors on a hypersonic vehicle. Traveling at hypersonic speeds drives a very different flow-field than an aircraft traveling at subsonic or supersonic speeds [51]. The hypersonic flow around a vehicle thickens the boundary layer around the vehicle, potentially resulting in large flowfield gradients that could adversely impact EO sensors. Integration of any sensors would be highly driven by vehicle structure as well as the resultant flow-field at hypersonic speeds, to avoid stressing thermal environments as well as excessive turbulence and flow-field shock structures (density gradients) at the sensors' imaging window [39]. As shown in Figure 9, a sensor model would include the geometries from the vehicle to the target, sensor placement and orientation, as well as effects caused by the boundary layer, shock waves, and thermal phenomena [44].

There has been little experience with relevant EO sensors at hypersonic speeds. The SR-71 was a high-speed reconnaissance aircraft that employed both film and EO sensors and operated at Mach 3+. At that velocity, SR-71 EO sensors would experience similar phenomena as sensors operating at hypersonic velocities, such as sensor window and bay temperatures, albeit at less severe levels. There has also been development and operational experience of EO sensors used in missile defense interceptors. These sensors operate primarily in the Long-wave Infrared (LWIR) band and track their targets against a cold background (sky or space) [44].

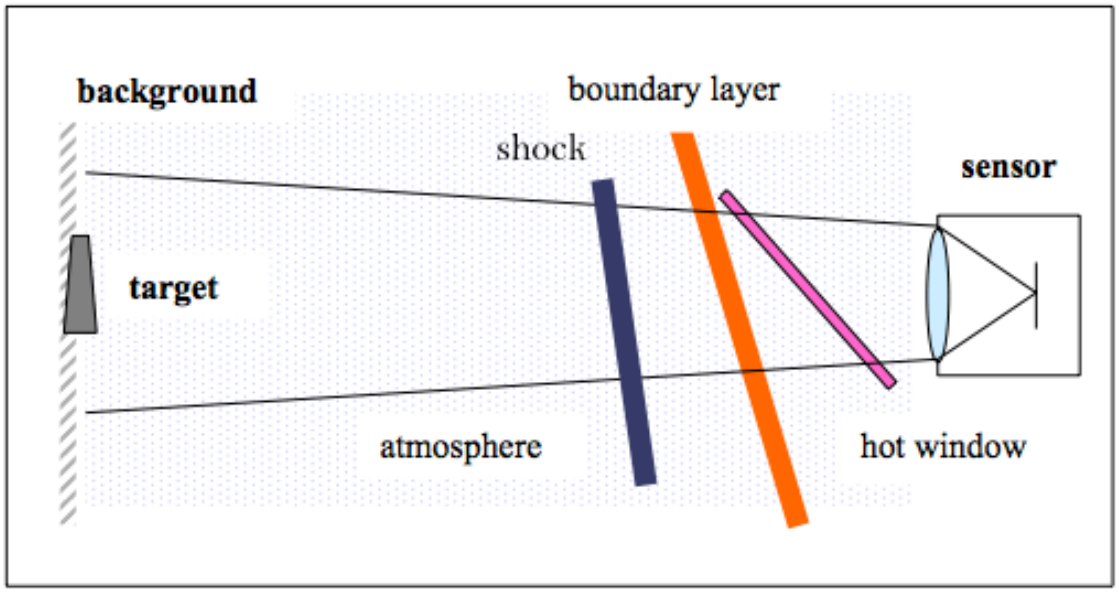


Figure 9. Notional Optical Radiometry Model [44]

In the mid-2000s, AFRL undertook the Hypersonic Aerospace Sensor Technology (HAST) study to evaluate a sensor suite that could perform reconnaissance, targeting, and terminal guidance functions for a hypersonic air vehicle, focusing on unpowered boost-glide reentry vehicles, such as a Common Aero Vehicle (CAV).

> A radiometric model was generated to calculate Signal to Noise (SNR) using parameters and geometry from the scenario for the HAST mission. The Field of View (FOV) versus slant range and the tradeoffs on field of regard were then studied. Further, the radiometry starting from the ground, through the atmosphere, through the shock layer, through the hot boundary layer, through the window system, and finally into the sensor was investigated. At each stage choices were made to eliminate spectral bands due to opaque transmission, background flux, or other phenomenology that would preclude low SNRs [44].

The study looked at two specific scenarios: a penetrator mission against deeply buried targets, and a submunition dispensing mission against soft, wide-area targets. While both of these missions are not reconnaissance missions, the vehicle and its sensors will experience similar environmental conditions as the hypersonic reconnaissance vehicle

considered in this research. The main difference is that the HAST CAV will operate at Mach 10–15+, which are significantly higher velocities than the hypersonic ISR vehicle considered in this research. Besides the increased severity of the thermal and flow field effects, the CAV's sensor will also have to compensate for plasma effects encountered at these high Mach numbers [26].

Based on the work done to meet the above objectives, the study identified several key drivers that influence sensor design and operations [44].

- Thermal Effects
  - High vehicle surface temperature, drives sensor aperture window materials
  - Thermal loads from boundary layer, increases background noise (decreases SNR)
  - Thermal flux from flow field, increased thermal load on sensor bay equipment
- Aero-optics Effects
  - EO distortions due to flow field and shock layer, perturbs EO signal in the form of Wavefront Errors (WFE)
- Trajectory Dynamics Effects
  - Operations at high Mach numbers could drive high angular rates of change, limits sensor dwell time to minimize image smearing
- Atmospherics Effects
  - EO propagation thru the atmosphere is affected by wavelength and transmission distance, drives sensor wavelength selection and vehicle trajectories

Since the research in this document is focused on trajectory optimization and not hypersonic aerodynamics or EO sensor development, this research incorporates only the effects of trajectory dynamics on sensors, resulting in only using distances and angles between the sensor and the collection target. Thermal effects due to thermal loads on the vehicle structure will also be considered, but the research will not include thermal and flowfield effects on the sensor. The other sensor effects are difficult to model and are beyond the scope of this research but could be candidates for future research topics.

2.8 Summary

This research investigates gaps that were found in the above literature search. Specifically, this research effort develops a methodology for determining optimal trajectories for hypersonic vehicles that uses an optimal control formulation incorporating several path constraints including vehicle skin temperature. This research adds to the current body of knowledge because it integrates the use of a hypersonic vehicle model, a sensor model, and an aerothermal effects model. The technique will allow hypersonic researchers to quickly generate optimal flight profiles that can be used in trade studies and systems engineering architecture assessments. Researchers will have the ability to incorporate flight path characteristics such as temperature constraints, originating/terminating locations, waypoints, keep out zones, and varied sensor targets.

III. Modeling

THIS chapter describes the environmental models (gravity and atmosphere), the vehicle models (aerodynamic and propulsion), and the aerothermal models used in this research. These models describe the system under study, an air-breathing hypersonic vehicle. How these models are appropriately employed is key to gaining confidence in the results and conclusions of the research.

3.1 Gravitational Model

This research uses a simple gravitational model where the earth is a smooth sphere with a constant radius. At any point along a candidate trajectory, the (local) gravity (g) can be expressed as an inverse square law, a function of the distance from the center of the earth (r_e) to the vehicle (r), with h as the altitude above the earth's surface and μ as the geocentric gravitational constant ($1.40764e16 ft^3/s^2$).

$$g = \frac{\mu}{r^2} = \frac{\mu}{(h + r_e)^2} \tag{7}$$

3.2 Atmospheric Model

The atmospheric model used in this research is based on the 1976 Standard Atmosphere [3]. This commonly used reference provides the atmospheric properties of temperature, speed of sound, density, and pressure as a function of geometric altitude. The model is limited to altitudes up to approximately 53 miles (86km) which is much higher than the expected operational altitudes of the vehicle in this research. In this research, only atmospheric density and speed of sound are needed.

A MATLAB function for a standard atmosphere function based on the 1976 Standard Atmosphere was considered to calculate the atmospheric density and speed of

sound for a given geometric altitude [68]. Since this function implements the lookup tables for the speed of sound from the 1976 Standard Atmosphere reference tables without any smoothing, the resulting data has discontinuities in its derivatives (shown in Figure 10) and therefore cannot be used "as is" with the Non-Linear Programming (NLP) solvers implemented in this research.

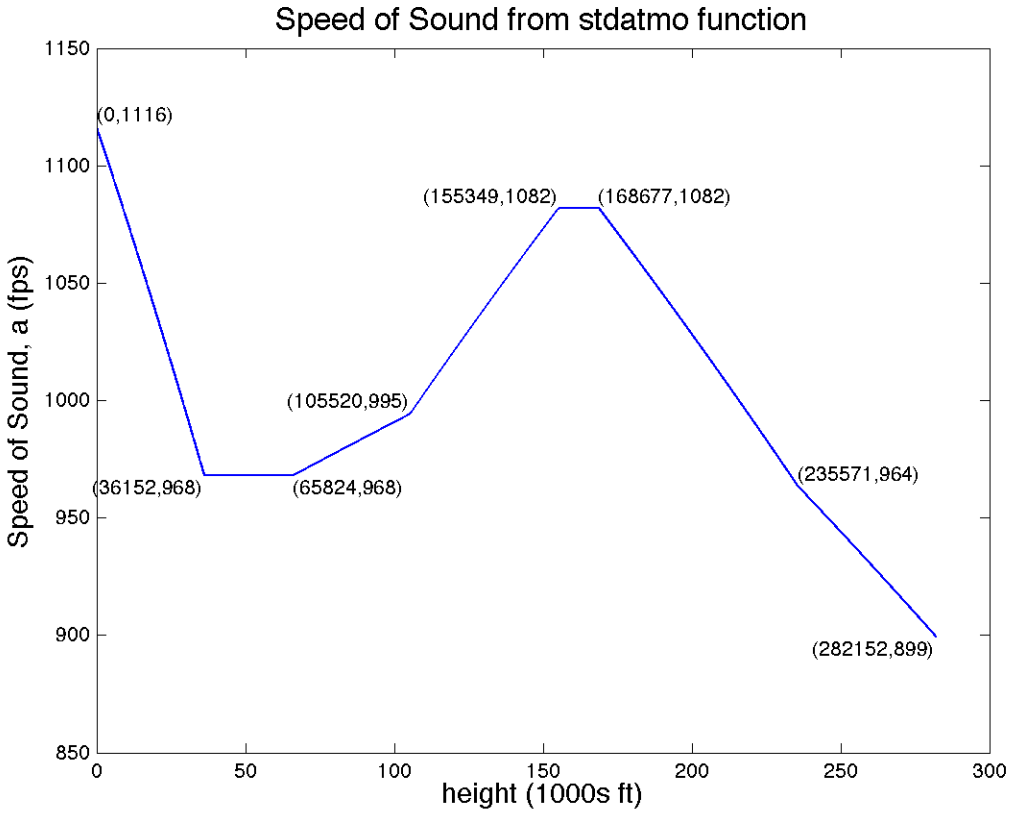


Figure 10. 1976 Standard Atmosphere Speed of Sound (a) Values

For the speed of sound calculations, the standard atmosphere data needs to be modified so that the first and second derivatives are continuous. As shown in Figure 10, the original data has discontinuous first derivatives. Eventually, a higher-order polynomial curve fit was used that included points between each "sharp corner" in Figure 10 but did not include the corners themselves. The resulting curve fit shown in Figure 11 shows an accurate fit with the original data, with an average relative

error of 0.013%.

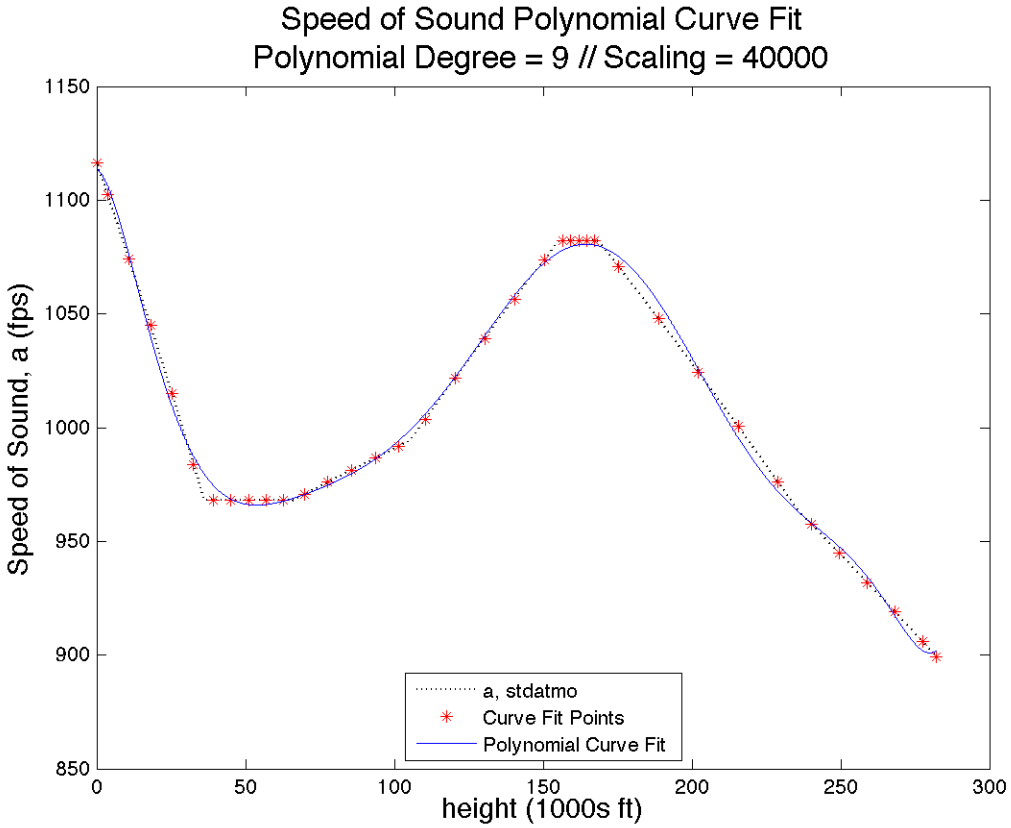


Figure 11. 1976 Standard Atmosphere Speed of Sound (a) Curve Fit

For the density calculations, polynomial and exponential curve fits were considered but not implemented since they introduced significant errors at altitudes where this vehicle may operate. For example, both the polynomial and exponential curve fits had average relative errors of 116% and 91%, respectively, over the range of typical operating altitudes. As seen in Figure 12, the exponential curve fit is significantly different from the tabular data and even though the polynomial curve fit appears to be accurate, the polynomial values oscillate between positive and negative values when the vehicle operates at higher altitudes (shown in Figure 13), which is not representative of the actual values. Eventually, a spline-based "griddedInterpolant" function was used to ensure continuity in the first and second derivative calculations.

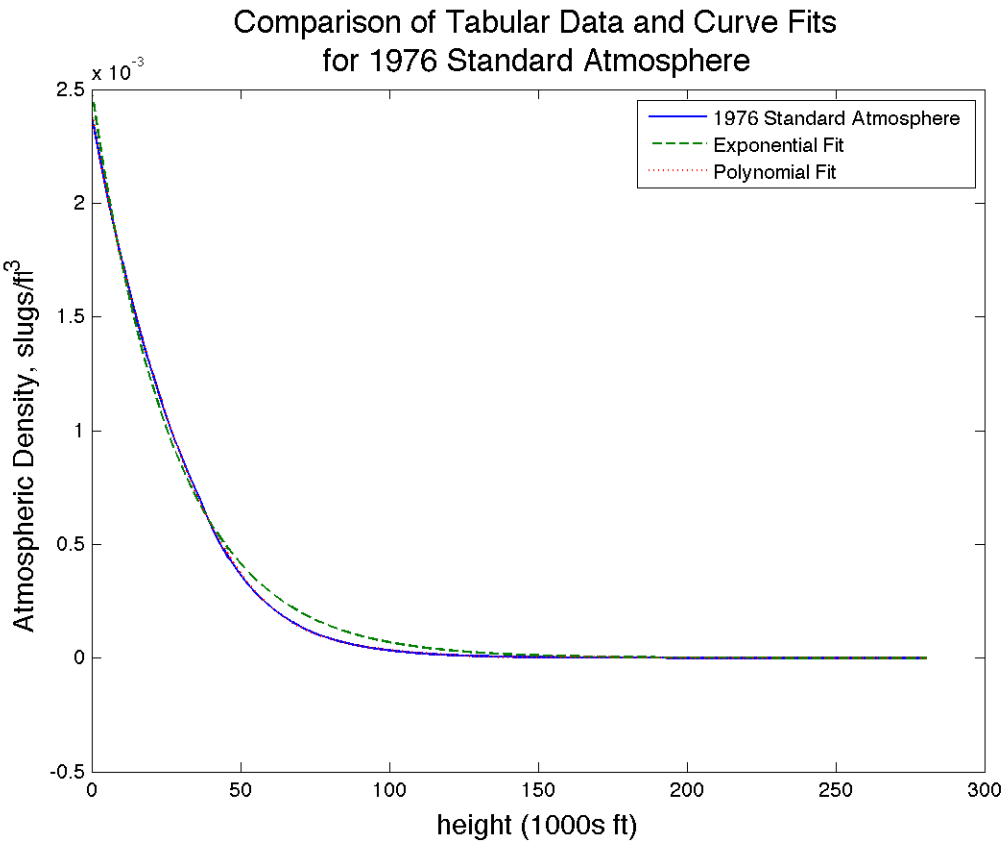


Figure 12. 1976 Standard Atmosphere Density (ρ) Curve Fits

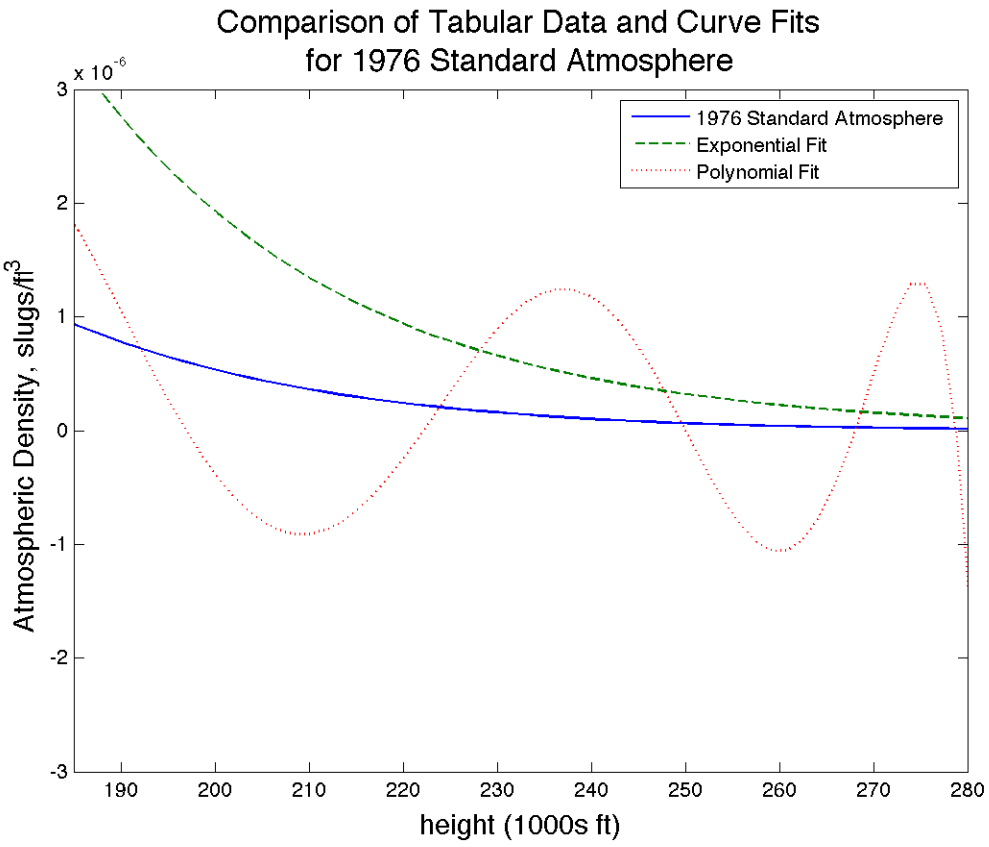


Figure 13. 1976 Standard Atmosphere Density (ρ) Curve Fits at High Altitude

3.3 Vehicle Model

For this research, a remote sensing mission is used. A nominal mission planning profile is considered and may include: runway takeoff, climb to cruise altitude, aerial refueling, avoidance of no-fly zones, descent from cruise altitude, and runway landing.

To perform this mission, a hypersonic air-breathing vehicle is considered. As stated previously, the modeling of a hypersonic vehicle is different from a subsonic or supersonic vehicles because the hypersonic vehicle models need to be enhanced to incorporate the unique modeling of aerodynamic, propulsion, and aerothermodynamic aspects.

The Generic Hypersonic Aerodynamic Model Example (GHAME) hypersonic vehicle model used in this research is taken from Bowers [13] and White [84]. GHAME is a hypothetical aircraft to provide simulation models for design activities, such as trajectory optimization. It was developed to provide aerodynamic, aerothermodynamic and propulsion data representative of a Single Stage to Orbit (SSTO)-class air-breathing hypersonic vehicle using hydrogen fuel. The shape of the vehicle was based on a composite of simple geometric shapes that resemble an aircraft[1]. Table 1 lists the aerodynamic and propulsion reference data for the GHAME vehicle. The values for the minimum and maximum fuel flow rates were incorporated from Air Force Research Laboratory's (AFRL) GHAME implementation in their DOF36 model [82] [83].

GHAME simulates a SSTO Turbine Based Combined Cycle (TBCC) vehicle that takes off horizontally, accelerates to near-hypersonic velocities using a turbine engine, transitions to a scramjet for hypersonic velocities, cruises at hypersonic velocities or accelerates to orbital velocities, and then returns to earth via a horizontal landing [13].

[1] A cylinder modeled the fuselage, two cones modeled the nose and boattail, thin triangular plates modeled the winds and vertical tail, another cone modeled the inlet, another cylinder modeled the engine, and strakes were added for the engine nozzle [84].

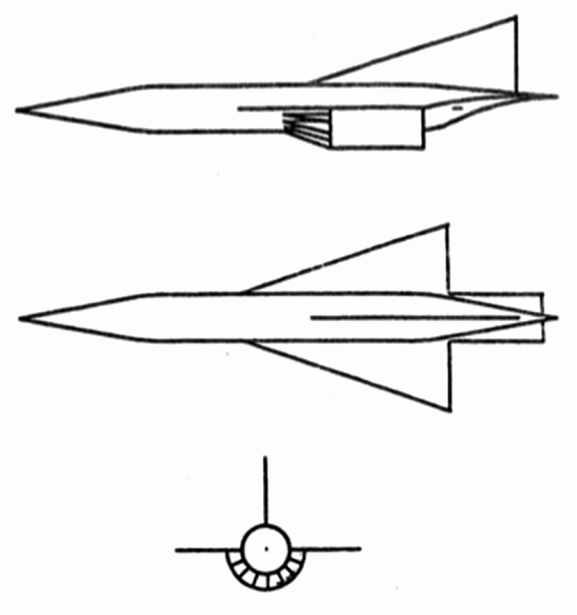


Figure 14. GHAME Aircraft Configuration [84]

Since this research is only going to consider operations at lower hypersonic velocities (Mach 5 to 8), this model is still valid in this limited flight regime. This models a hypersonic vehicle traveling in the atmosphere using derived equations of motion and empirical data for both the aerodynamic force and the engine thrust models.

For this research, the aerodynamic forces and engine thrust models are adapted from the GHAME data sets, which are based on empirical data from previous operational systems and wind tunnel testing [13] [84]. The lift, drag, and propulsion forces are computed from force and thrust coefficients in GHAME data sets, which vary based on vehicle flight conditions. The coefficients will vary with Mach number (M) from Mach 0.4 to (approximately) Mach 8 (although the GHAME data goes to Mach 24), angle of attack (α) $-3^\circ \leq \alpha \leq 21^\circ$, and fuel-air equivalence ratio (φ) $0 \leq \varphi \leq 2$.

Table 1. GHAME Reference Data [13] [84] [83]

GHAME Attribute	Values
aerodynamic reference area, A_{ref}	6000 ft^2
length	233.4 ft
reference chord	75 ft
reference span	80 ft
takeoff gross mass, $m_{\max}$	300000 lbm
zero fuel mass, $m_{\min}$	120000 lbm
inlet capture area, A_{ic}	300 ft^2
minimum fuel flow rate, $\dot{m}_{f_{\min}}$	20 lbm/s
maximum fuel flow rate, $\dot{m}_{f_{\max}}$	360 lbm/s

3.3.1 Aerodynamics Model.

The lift force (L) equation in the aerodynamics model for this vehicle is a standard linear equation for the lift coefficient (C_L) with A_{ref} given in the GHAME data set and modified by Zipfel[2], defined as, with q_∞ as dynamic pressure:

$$L = C_L q_\infty A_{ref} \tag{8}$$

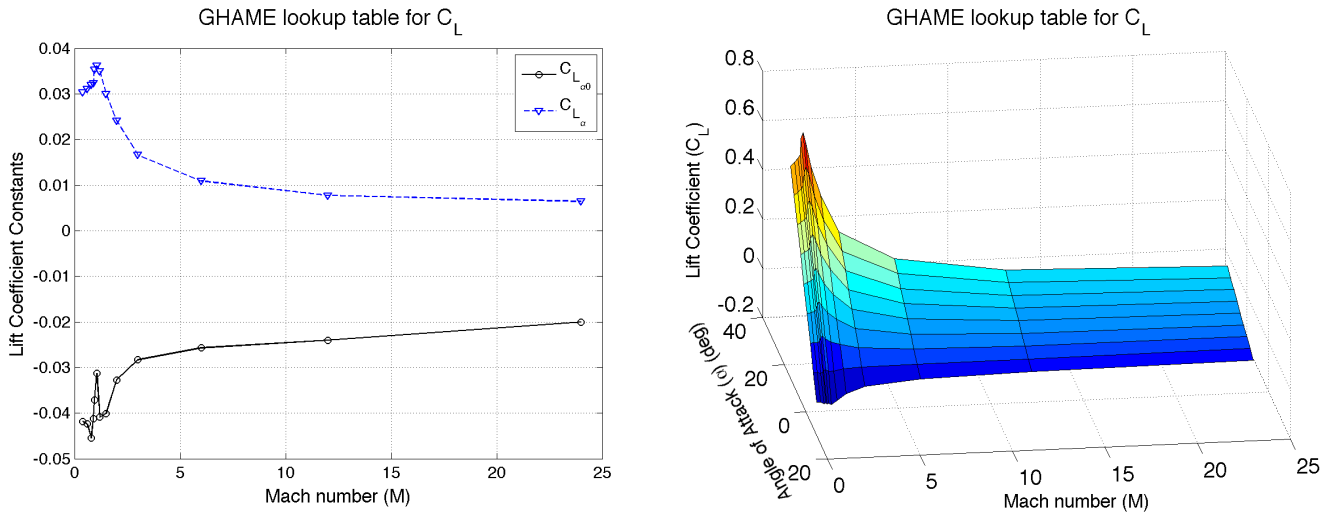
$$q_\infty = \frac{\rho V^2}{2} \tag{9}$$

$$C_L = C_{L_{\alpha_0}} + C_{L_\alpha}\alpha \tag{10}$$

The terms coefficient of lift due to alpha ($C_{L_{\alpha_0}}$) and coefficient of lift due to alpha

[2]The lift coefficient formulation used here is given by Zipfel [92], simplifying the GHAME lift coefficient equation by integrating the elevator deflection and pitch rate dependencies into the remaining terms.

(C_{L_α}) are functions of Mach number and are defined in the GHAME data tables. In Figure 15 below, both the 3-D plot of C_L and 2-D plot for C_L equation terms ($C_{L_{\alpha_0}}$ and C_{L_α}) are shown.



(a) 2-D Equation terms (slope and intercept coefficients)

(b) 3-D Curve as a function of M and α

Figure 15. GHAME Lift Coefficient Curves [92] expressed as either (a) a linear function with slope and intercept coefficients as a function of M and (b) as a 3-D function of M and α

While C_D and C_L are vehicle frame force coefficients, it is useful to define the resulting stability frame axial and normal force coefficients, C_A and C_N. These force coefficients are useful to visualize vehicle forces as well as factors used in additional path constraints.

$$\begin{bmatrix} C_A \\ C_N \end{bmatrix} = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} C_D \\ C_L \end{bmatrix} \tag{11}$$

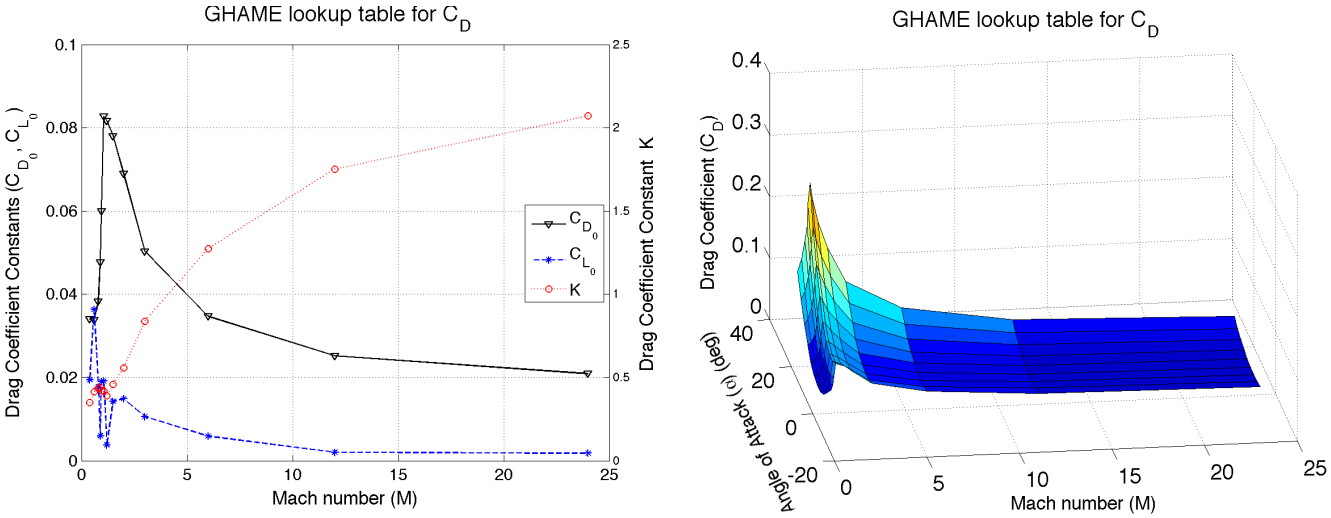
The drag force (D) equation in the aerodynamics model for this vehicle is a standard offset parabolic drag polar equation for the drag coefficient (C_D)[3] with A_{ref} given in the GHAME data set, defined as:

[3]The drag coefficient formulation used here is given by Zipfel [92], modifying the GHAME drag coefficient equation by transforming it into a parabolic equation.

$$D = C_D q_\infty A_{ref} \tag{12}$$

$$C_D = C_{D_0} + k(C_L - C_{L_0})^2 \tag{13}$$

The terms for minimum drag coefficient (C_{D_0}), drag coefficient constant (k), and zero-coefficient lift force (C_{L_0}) are functions of Mach number and are defined in the GHAME data tables. In Figure 16, both the 3-D plot of C_D and 2-D plot for C_D equation terms (C_{D_0}, k, and C_{L_0}) are shown.



(a) 2-D Equation terms (slope and intercept coefficients)

(b) 3-D Curve as a function of M and α

Figure 16. GHAME Drag Coefficient Curves [92] expressed as either (a) a linear function with slope and intercept coefficients as a function of M and (b) as a 3-D function of M and α

3.3.2 Propulsion Model.

The GHAME thrust equation in the propulsion model for this vehicle is a standard thrust equation, defined in Equation 14, where T is thrust, g_0 is the gravitational acceleration on the earth's surface, $\dot{m}_f$ is the propellant mass flow rate, and I_{sp} is the

specific impulse.

$$T = g_0 \dot{m}_f I_{sp} \tag{14}$$

I_{sp} is provided in the GHAME data set and is a function of Mach number and fuel-air equivalence ratio, φ, which is defined in Equation 15, and where Φ is the fuel-air ratio, $\dot{m}_a$ is the air mass flow rate, and Φ_{st} is the fuel stoichiometric fuel-air ratio[4].

$$\varphi = \frac{\Phi}{\Phi_{st}} = \frac{\frac{\dot{m}_f}{\dot{m}_a}}{\Phi_{st}} = \frac{\frac{\dot{m}_f}{\rho V A_0 / A_c A_{ic}}}{\Phi_{st}} \tag{15}$$

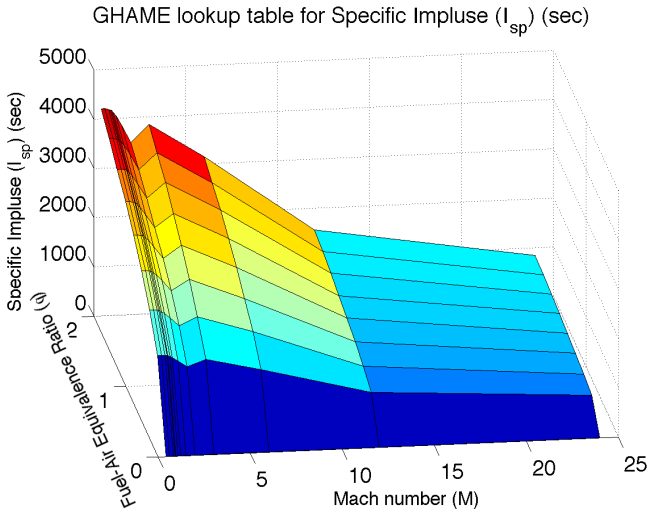


Figure 17. GHAME Specific Impulse (I_{sp}) [84]

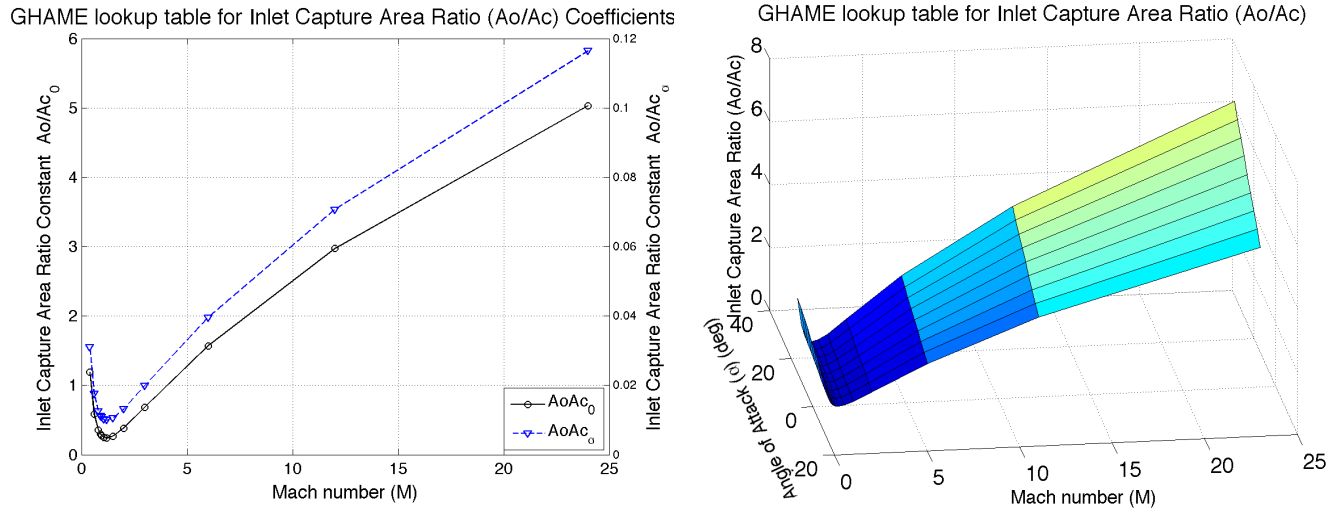
The GHAME model simulates a variable-geometry engine inlet which necessitates a lookup table for the inlet capture area ratio[5]. A_0/A_c is the inlet capture area ratio and is a function of Mach number and α. In the referenced GHAME documentation [84], A_0/A_c is provided as a lookup table based on Mach number and α, but it can be reduced to a linear equation that varies with α where the coefficients inlet

[4]For hydrogen fuel, $\Phi_{st} = .0292$

[5]For smaller engines, the inlet has a fixed geometry. The inlet would be designed for a specific cruise altitude and velocity, also driving flight at a specific dynamic pressure range.

capture area ratio at zero alpha (A_0/A_{c_0}) and inlet capture area ratio due to alpha (A_0/A_{c_α}) are only based on Mach number.

$$A_0/A_c = A_0/A_{c_0} + A_0/A_{c_\alpha}\alpha \tag{16}$$



(a) 2-D Equation terms (slope and intercept coefficients)

(b) 3-D Curve as a function of M and α

Figure 18. GHAME Inlet Capture Area Ratio Curves [84] expressed as either (a) a linear function with slope and intercept coefficients as a function of M and (b) as a 3-D function of M and α

3.4 State Dynamics

For this research, the standard three degrees of freedom (3DOF) equations of motion as derived by Vinh for a point mass [79] is used. Symmetric flight is assumed, thus sideslip angle β is zero and not included in any equations.

A 3DOF model is used since this tool is intended to do responsive trajectory optimization analysis for mission planners and systems engineers. The 3DOF model used in this research is also similar to a trajectory model developed by Watson and Liston for AFRL [82] [83]. In the original AFRL 3DOF model, vehicle control inputs were given as control surface deflections for pitch, roll, and yaw. As a result, the

original model also included vehicle control loops [54]. Since this research is focused on a mission planning and systems engineering tool, the model in this research will simplify the AFRL model by removing the vehicle control loops and use stability and body angles for control inputs. This simplification is common in controls and trajectory research and is commonly called "inertialess" control [12].

The research model is a modification of the AFRL 3DOF model since this research uses an optimal control formulation while the AFRL model uses a second-order Runge-Kutta method to solve the equations of motion. Since solving the equations of motion is built into the formulation given to General Pseudospectral Optimal Control Software (GPOPS), Runge-Kutta methods are not used in this research [54] [53]. This 3DOF model assumes the vehicle is a rigid body. The model also uses empirically-based tabular data for aerodynamic and propulsive forces to compute equations of motion [82]. When implemented in MATLAB, these data sets will be interpolated using spline or cubic interpolation methods to provide smooth data points when computing the equations of motion.

3.4.1 Reference Frames.

Several reference frames (and coordinates) are needed and are given in Figure 19.

The transformation from the inertial reference frame to the earth reference frame is defined as $\mathbf{T_i^e}$, where θ is latitude north of the equator and Φ is the longitude east of an arbitrary reference, as shown in Figure 20:

$$\mathbf{T_i^e} = \begin{bmatrix} -\sin\Phi & \cos\Phi & 0 \\ -\sin\theta\cos\Phi & -\sin\theta\sin\Phi & \cos\theta \\ \cos\theta\cos\Phi & \cos\theta\sin\Phi & \sin\theta \end{bmatrix} \tag{17}$$

The transformation from the earth reference frame to the velocity reference frame

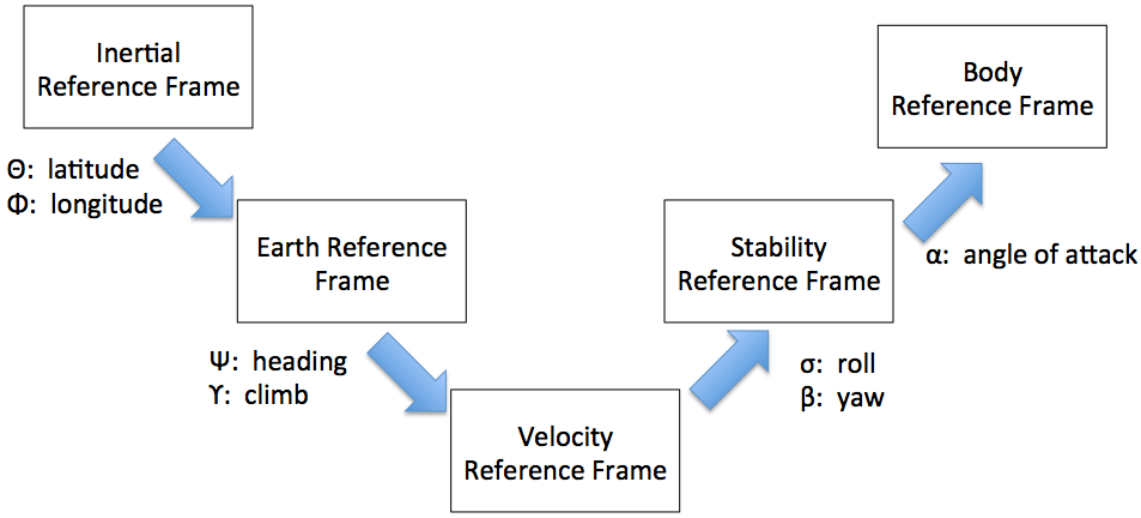


Figure 19. Reference Frames

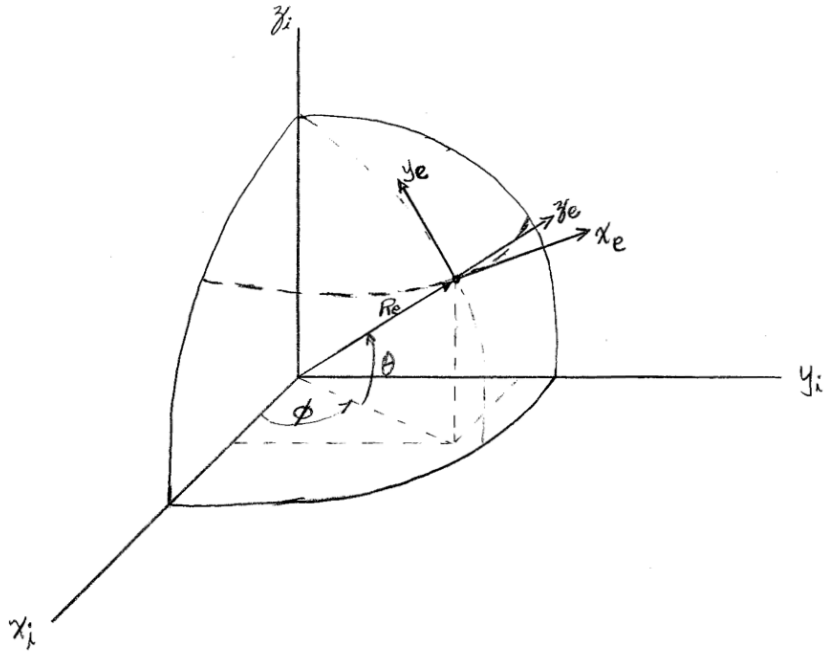


Figure 20. Earth Reference Frame [82]

is defined as $\mathbf{T_e^v}$, where Ψ is the heading angle and γ is the climb angle, as shown in Figure 21:

$$\mathbf{T_e^v} = \begin{bmatrix} \cos\gamma\cos\Psi & \cos\gamma\sin\Psi & \sin\gamma \\ -\sin\Psi & \cos\Psi & 0 \\ -\sin\gamma\cos\Psi & -\sin\gamma\sin\Psi & \cos\gamma \end{bmatrix} \tag{18}$$

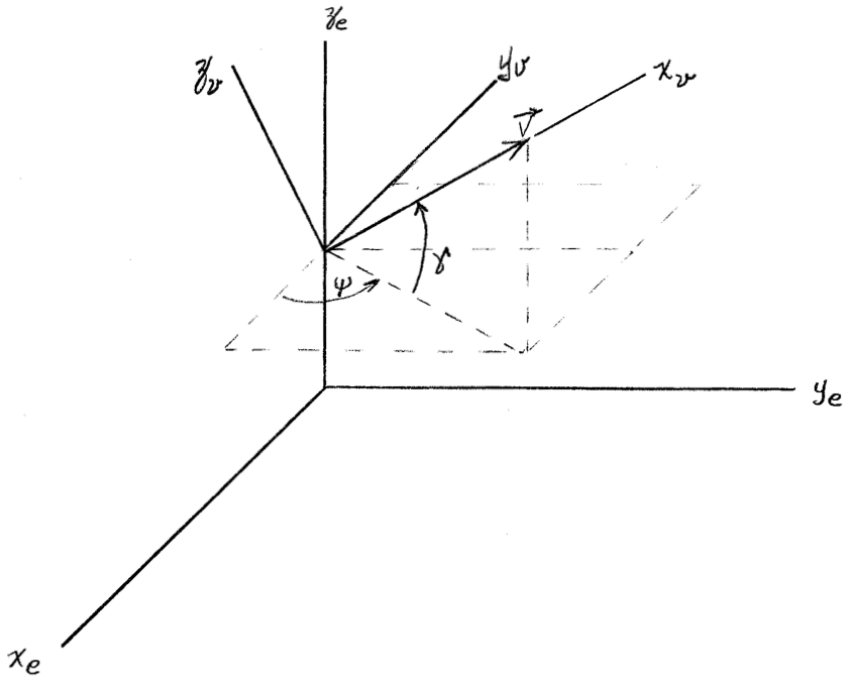


Figure 21. Velocity Reference Frame [82]

The transformation from the velocity reference frame to the stability reference frame is defined as $\mathbf{T_v^s}$, where σ is the roll angle about the velocity vector and β is the yaw angle, as shown in Figure 22:

$$\mathbf{T_v^s} = \begin{bmatrix} \cos\beta & \sin\beta\cos\sigma & \sin\beta\sin\sigma \\ -\sin\beta & \cos\beta\cos\sigma & \cos\beta\sin\sigma \\ 0 & -\sin\sigma & \cos\sigma \end{bmatrix} \tag{19}$$

The transformation from the stability reference frame to the body reference frame

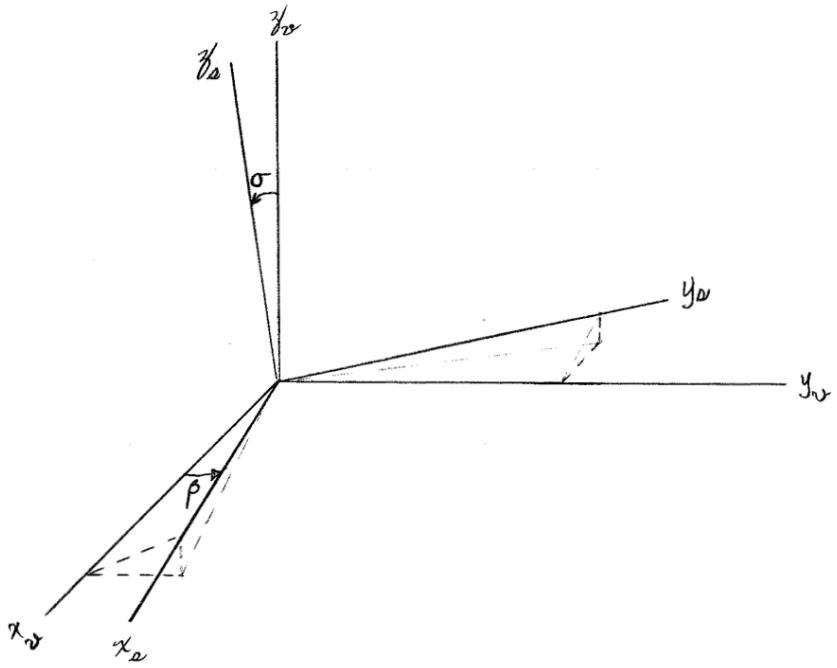


Figure 22. Stability Reference Frame [82]

is defined as $\mathbf{T_s^b}$, where α is the angle of attack, as shown in Figure 23:

$$\mathbf{T_s^b} = \begin{bmatrix} \cos\alpha & 0 & \sin\alpha \\ 0 & 1 & 0 \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix} \tag{20}$$

Since these rotations are between orthogonal reference frames, these transformation matrices are orthonormal transformations, and the inverse of these transformations is just the transpose of the given matrix [49]. For example, $\mathbf{T_b^s} = \mathbf{T_s^b}^{\mathrm{T}}$. Transformation matrices can also be expressed across multiple reference frames to shorten nomenclature. For example, the transformation from the earth to body reference frames is $\mathbf{T_e^b} = \mathbf{T_s^b T_v^s T_e^v}$ and, similarly, the transformation from body to earth reference frames is the transpose of this transformation, $\mathbf{T_b^e} = \mathbf{T_v^e T_s^v T_b^s}$.

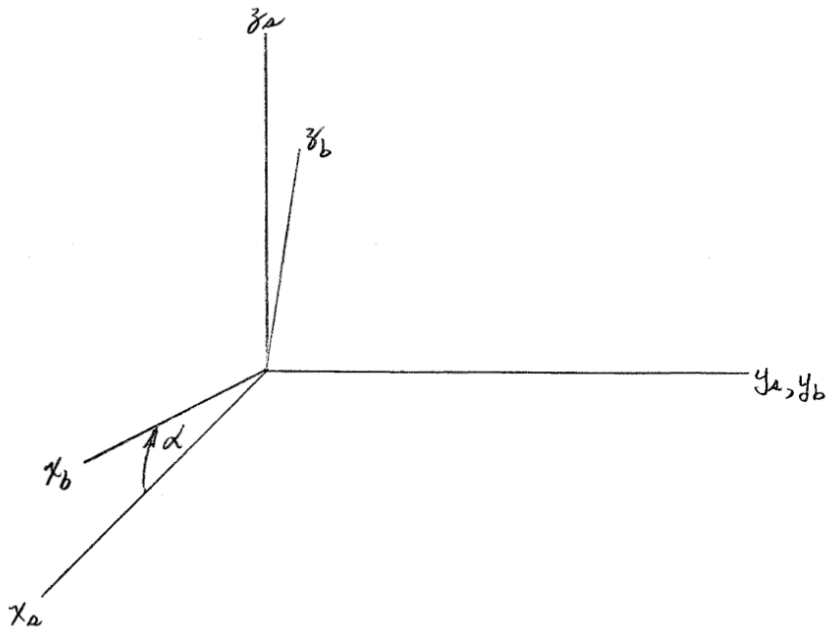


Figure 23. Body Reference Frame [82]

3.4.2 States and Controls.

When angular controls are used, the vehicle model will use seven states elements that include the vehicle position in spherical coordinates, velocity, attitude, and mass. They are the radial distance from vehicle to earth center of mass (r), latitude (θ), longitude (ϕ), velocity (V), climb angle (γ), heading angle (ψ), and mass (m). The vehicle model uses three controls: angle of attack (α), roll angle (σ), and propellant mass flow rate ($\dot{m}_f$). When angular rate controls are used for "with control penalty" in the objective functional, the vehicle model uses nine states by adding the following two states to the above seven states, α and σ, and the control states become angle of attack rate ($\dot{\alpha}$), roll angle ($\dot{\sigma}$), and propellant mass flow rate ($\dot{m}_f$).

3.4.3 Equations of Motion.

The following set of equations are the equations of state (or dynamic constraints). The first three equations are kinematic equations, the second three equations are force

equations and the last equation is the mass flow rate equation. As previously stated, these equations of motion assume flight over a spherical, rotating earth and coordinated turns with no sideslip angle. These equations are also considered "inertialess" since they do not include command delays, from using control surface deflections, and body moments [12].

$$\frac{dr}{dt} = V \sin\gamma \tag{21}$$

$$\frac{d\theta}{dt} = \frac{V \cos\gamma \cos\psi}{r \cos\phi} \tag{22}$$

$$\frac{d\phi}{dt} = \frac{V \cos\gamma \sin\psi}{r} \tag{23}$$

$$\frac{dV}{dt} = \frac{F_T}{m} - g \sin\gamma + \omega^2 r \cos\phi \left(\sin\gamma \cos\phi - \cos\gamma \sin\psi \sin\phi\right) \tag{24}$$

$$\frac{d\gamma}{dt} = \frac{F_N \cos\sigma}{mV} - \cos\gamma \left(\frac{V}{r} - \frac{g}{V}\right) + 2\omega \cos\psi \cos\phi + \frac{\omega^2 r \cos\phi}{V} \left(\cos\gamma \cos\phi + \sin\gamma \sin\psi \sin\phi\right) \tag{25}$$

$$\frac{d\psi}{dt} = \frac{F_N \sin\sigma}{mV \cos\gamma} - \frac{V}{r} \cos\gamma \cos\psi \tan\phi + 2\omega \left(\tan\gamma \sin\psi \cos\phi - \sin\phi\right) - \frac{\omega^2 r}{V \cos\gamma} \cos\psi \sin\phi \cos\phi \tag{26}$$

$$\frac{dm}{dt} = -\frac{T}{I_{sp} g_0} \tag{27}$$

For the equations of motion in Equations 21 thru 27, the states and controls can

be expressed as vectors.

$$\mathbf{x} = \begin{bmatrix} r(t) \\ \theta(t) \\ \phi(t) \\ V(t) \\ \gamma(t) \\ \psi(t) \\ m(t) \end{bmatrix} \tag{28}$$

$$\mathbf{u} = \begin{bmatrix} \alpha(t) \\ \sigma(t) \\ \dot{m}_f(t) \end{bmatrix} \tag{29}$$

As described in Section 3.4.2, when angular rate controls are used for "with control penalty" objectives, the vehicle model will use two additional states, two additional equations of motion, and two angular rate controls instead of angular controls. The new states and controls and additional equations of motion are:

$$\mathbf{x} = \begin{bmatrix} r(t) \\ \theta(t) \\ \phi(t) \\ V(t) \\ \gamma(t) \\ \psi(t) \\ m(t) \\ \alpha(t) \\ \sigma(t) \end{bmatrix} \tag{30}$$

$$\mathbf{u} = \begin{bmatrix} \dot{\alpha}(t) \\ \dot{\sigma}(t) \\ \dot{m}_f(t) \end{bmatrix} \tag{31}$$

$$\frac{d\alpha}{dt} = \dot{\alpha} \tag{32}$$

$$\frac{d\sigma}{dt} = \dot{\sigma} \tag{33}$$

3.5 Aerothermal Modeling

For an aircraft traveling at subsonic or low supersonic speeds, aerodynamics forces and vehicle dynamics are the dominant effects that need to be modeled to determine optimal trajectories. As an aircraft travels faster, aerothermal heating effects become significant and need to modeled as well. As described in Chapter II, MINIVER (Miniature Version) is National Aeronautics and Space Administration (NASA)-developed engineering tool to assess the aerothermal environment around a vehicle [88]. It is a version of the JA70 General Aerodynamic Heating Computer Code, with development starting in the late 1960s [46]. MINIVER provides several design analysis capabilities.

- Assessment of aerothermal environment over critical regions of the vehicle
- Engineering-level aeroheating thermal analysis capability
- Conceptual/preliminary design analyses
- Interactive use with other vehicle design tools

MINIVER is composed of two software packages, LANMIN (Langley MINIVER Code) and Explicit Interactive Thermal Structures Code (EXITS). LANMIN requires a small set of inputs for the vehicle's trajectory (altitude, velocity, and angle of

attack as functions of time) and calculates the flowfield's aeroheating parameters, including heat transfer coefficients, recovery enthalpy, laminar and turbulent flow conditions, heat rates/loads, local and wall flow conditions and flow field transition information. EXITS uses LANMIN outputs, plus the vehicle's geometry and Thermal Protection System (TPS) configuration, to calculate time-dependent temperatures at selected points on the vehicle's surface as well as the accompanying TPS cross-section temperatures [88]. While MINIVER is used extensively for aerothermal analysis, it is typically run off-line given a vehicle geometry and free stream flowfield.

As written, MINIVER is not compatible with MATLAB-based framework in this research since MINIVER is comprised of about 12500 lines of Fortran 66 source code. In addition, MINIVER uses American Standard Code for Information Interchange (ASCII) output files to transfer data from LANMIN to EXITS and also to return the final MINIVER results to the user. This use of ASCII-based transfer limits MINIVER's precision since ASCII file output truncates the data to a precision well than less than a computer's single-precision floating-point precision. To make the MINIVER code operable with this research, the Fortran code was modified such that binary files are now used to input/output data when using both LANMIN and EXITS. As a result, truncation error from using ASCII data files is eliminated but MINIVER is still executing only as single-precision code. In addition to using binary files, the source code was modified to use batch files to input user data, instead of interacting with the user via terminal windows. After making changes to the source code, the Intel Fortran Compiler for OS X and Windows were used to compile the source code into executable application for OS X and Windows-based machines, respectively.

Table 2 shows the significant settings for MINIVER. The settings were selected based on their applicability to this research. For example, for the nose analysis position, the Fay-Riddell Stagnation Point heat transfer method is commonly used [88] [21],

while for the body analysis position, the Boeing $\rho\mu$ flat plate heat transfer method is commonly used [88] [45].

Table 2. MINIVER Settings

MINIVER Option	Selection
aerothermal analysis points	2 (nose and body)
nose flowfield	"sharp wedge shock angle"
nose pressure option	"tangent wedge pressure coefficient"
nose heat transfer method	"Fay-Riddell Stagnation Point [27]"
wedge (nose) angle	2.5 °
nose radius	0.5 inches
body position	windward side, fuselage mid-point
body flowfield	"oblique and normal shock"
body pressure option	"oblique shock"
body heat transfer method	"Boeing $\rho\mu$ flat plate [56]"
normal (body) shock angle	90 °
transition criteria	Reynolds and Mach numbers
running length	50% of fuselage length
LANMIN time spacing	4 sec
EXITS time spacing	5 sec
nose material and slab thickness	tantalum 0.8 inches
body material and slab thickness	molybdenum 2.4 inches

3.6 Modeling Summary

This section described the environmental, vehicle, dynamics, and aerothermal models used in this research. It provided the detailed implementation of these models,

including modeling modifications to enable their use with NLP solvers. The following chapter provides the methodology to develop the optimal control formulation with objective functionals and path constraints for several different scenarios. The subsequent chapter will provide the results and analysis for these scenarios, demonstrating the effectiveness of the research methodology.

IV. Methodology

4.1 Problem Statement

THE overall objective of this research is to develop a mission planning and systems engineering analysis methodology by rapidly developing optimal trajectories for a scramjet-based hypersonic reconnaissance vehicle.

As stated in Chapter I, the mission objective of a hypersonic reconnaissance system is to fly a reconnaissance mission over contested or denied regions, from an initial point to a final point to collect sensor data against specified collection targets. The vehicle is an air-breathing reconnaissance aircraft that has specified takeoff/landing locations, locations for air-to-air refueling, no-fly zones, and specified target for sensor data collections. Mission objectives are specified, such as minimizing a system parameter or a combination of multiple parameters, such as time of flight, fuel expended, or even a sensor parameter such as ground sample distance and/or sensor look angle. Along the path, waypoints can be defined such that the vehicle must pass thru a specified location, possibly at a specified time. Additional constraints can be imposed as well. No-fly zones can be specified to avoid locations or areas. A minimum level of survivability can also be specified. In addition, constraints related to vehicle performance, such as maximum dynamic pressure, can be imposed as well as constraints related to data sensor performance, such as sensor resolution.

4.2 Optimal Control Problem Formulation

In this research, three common types of optimal control problems will be considered: maximum range, minimum time, and minimum time with control penalty. Each of these problems are common design and mission planning scenarios.

4.2.1 Maximum Range Problem.

Maximum range problems are a direct application of the general optimal control problem. This formulation is useful for computing the maximum range of the vehicle for a given fuel load. The maximum range formulation is:

$$\min_{u^* \in U} J = -\int_{t_o}^{t_f} V dt \tag{34}$$

where V is the vehicle speed, given by $V = \left\| \begin{array}{c} \dot{h}(t) \\ \dot{\theta}(t) \\ \dot{\phi}(t) \end{array} \right\|$, and subject to Equation 4, with:

$$\text{Path Constraints } (\mathbf{C}), \forall \mathbf{t} \in [\mathbf{t_0}, \mathbf{t_f}]: \tag{35}$$

$$\varphi_{\min} \leq \varphi(\mathbf{x}(t), \mathbf{u}(t), t) \leq \varphi_{\max}$$

$$q_\infty(\mathbf{x}(t), \mathbf{u}(t), t) \leq q_{\infty \max}$$

4.2.2 Minimum Time Problem.

Minimum time problems are one of the simplest applications of the general optimal control problem. This formulation is useful for computing the fastest possible time to transit between two fixed points. For this research, this formulation will be used to determine the fastest possible time to execute a given mission profile, with prescribed mission parameters and constraints. The minimum time formulation of the cost functional is:

$$\min_{u^* \in U} J = \int_{t_o}^{t_f} dt = t_f - t_0 \tag{36}$$

subject to Equation 4 and with path constraints ($\mathbf{C}$) the same as in Equation 35.

4.2.3 Minimum Time with Control Penalty Problem.

Minimum time with control penalty problems are a variation to minimum time problems. In addition to the minimum time contribution to the objective functional, there is also a component for the control usage. The term ϵ is a convex combination coefficient to give the user control on the relative weighting of the time and control contributions. The minimum time with control penalty formulation of the cost functional is:

$$\begin{aligned} \min_{u^* \in U} J &= \epsilon(t_f - t_0) + (1-\epsilon) \int_{t_o}^{t_f} \|\mathbf{u}(t)\|^2_{\mathbf{R},\mathbf{2}}\, dt \\ &= \epsilon(t_f - t_0) + (1-\epsilon) \int_{t_o}^{t_f} \mathbf{u}^T(t)\, \mathbf{R}\mathbf{u}(t)\, dt \end{aligned} \tag{37}$$

subject to Equation 4 and with path constraints ($\mathbf{C}$) the same as in Equation 35. $\mathbf{R}$ is a real symmetric positive definite ($\mathbf{z}^T\mathbf{R}\mathbf{z} > 0 \;\forall\; \mathbf{z} \neq \mathbf{0}$) weighting matrix used in the weighted vector norm. $\mathbf{R}$ is used to "weight the relative importance of the different components of the control vector and to normalize the numerical values of the controls." [43]

Assuming that all the controls have the same relative importance and the numerical values of the controls are already normalized, then the weighting matrix $\mathbf{R}$ becomes the identity matrix and the minimum energy formulation becomes:

$$\min_{u^* \in U} J = \epsilon(t_f - t_0) + (1-\epsilon) \int_{t_o}^{t_f} \mathbf{u}^T(t)\, \mathbf{u}(t)\, dt \tag{38}$$

subject to Equation 4 and with path constraints (**C**) the same as in Equation 35.

4.2.4 Minimum Time with Control Penalty, No-Fly Zone and g-limits Problem.

This formulation is similar to Section 4.2.3 but it adds a path constraint for a minimum distance from fixed point(s) (center(s) of no-fly[1] zone(s)) and two other path constraints for the maximum g's (axial and normal) the vehicle can experience.

$$\min_{u^* \in U} J = \epsilon(t_f - t_0) + (1-\epsilon) \int_{t_o}^{t_f} \mathbf{u}^T(t)\, \mathbf{u}(t)\, dt \tag{39}$$

subject to Equation 4, with:

Path Constraints $(\mathbf{C}), \forall \mathbf{t} \in [\mathbf{t_0}, \mathbf{t_f}]$: (40)

$$\varphi_{\min} \leq \varphi(\mathbf{x}(t), \mathbf{u}(t), t) \leq \varphi_{\max}$$

$$q_\infty(\mathbf{x}(t), \mathbf{u}(t), t) \leq q_{\infty\max}$$

$$g_{a\min} \leq \mathbf{g_a}(\mathbf{x}(t), \mathbf{u}(t), t) \leq g_{a\max}$$

$$g_{n\min} \leq \mathbf{g_n}(\mathbf{x}(t), \mathbf{u}(t), t) \leq g_{n\max}$$

$$\left\|\mathbf{x}(t) - \mathbf{x}_{nofly_j}\right\|_2 \geq r_{nofly_j}$$

$$(j = 1, \ldots, N_z \text{ (number of no-fly zones)})$$

4.2.5 Minimum Time with Control Penalty, No-Fly Zone and Sensor Constraints Problem.

This formulation is similar to Section 4.2.4 but instead of having g limits, it adds sensor constraints.

Sensor designs (with corresponding technical specifications) are driven by the in-

[1] As described in Section 1.3, no-fly zones can be implemented for many reasons, such as avoiding high-threat areas or prohibited airspace.

tended mission, such as data collection against a fixed or mobile target. Many different design choices have to be used, such as: active vs. passive sensors, still imagery vs. motion imagery, imaging vs. non-imaging, and electromagnetic operational band selection. In this research, the geometric constraints for a passive Electro-Optical (EO) sensor will be modeled, since the focus of this research is developing a methodology for trajectory optimization and not sensor modeling.

Besides incorporating vehicle or trajectory parameters, the optimal control formulation can also incorporate sensor-related parameters, such as slant (look) angle/range, sensor ground resolution, Field of View (FOV), collection time (against a specific target), or coverage area. Most of these sensor parameters would be incorporated as additional path constraints, but some could be incorporated as additional terms in the objective functional.

$$\min_{u^* \in U} J = \epsilon(t_f - t_0) + (1 - \epsilon) \int_{t_o}^{t_f} \mathbf{u}^T(t)\, \mathbf{u}(t)\, dt \tag{41}$$

subject to Equation 4, with:

$$\text{Path Constraints } (\mathbf{C}), \forall \mathbf{t} \in [\mathbf{t_0}, \mathbf{t_f}]: \tag{42}$$

$$\varphi_{\min} \leq \varphi\left(\mathbf{x}(t), \mathbf{u}(t), t\right) \leq \varphi_{\max}$$

$$q_\infty\left(\mathbf{x}(t), \mathbf{u}(t), t\right) \leq q_{\infty\max}$$

$$\left\|\mathbf{x}(t) - \mathbf{x}_{nofly_j}\right\|_2 \geq r_{nofly_j}$$

$$(j = 1, \ldots, N_z \text{ (number of no-fly zones)})$$

$$\mathbf{S_{SR}}\left(\mathbf{x}(\mathbf{t}), \mathbf{u}(\mathbf{t}), \mathbf{t}\right) \leq \mathbf{S_{SR}}_{\max}$$

$$S_{LOS\min} \leq \mathbf{S_{LOS}}\left(\mathbf{x}(t), \mathbf{u}(t), t\right) \leq S_{LOS\max}$$

$$S_{az\min} \leq \mathbf{S_{az}}\left(\mathbf{x}(t), \mathbf{u}(t), t\right) \leq S_{az\max}$$

$$S_{el\min} \leq \mathbf{S_{el}}\left(\mathbf{x}(t), \mathbf{u}(t), t\right) \leq S_{el\max}$$

4.3 Sensor Modeling as Constraints

As discussed in Section 2.7, this research is focused on trajectory optimization and not hypersonic aerodynamics or EO sensor development. Thus, this research incorporates only the effects of trajectory dynamics on sensors, specifically geometric constraints related to the sensor and the target.

As described by Gundlach [37], sensor Field of Regard (FOR) parameters can be calculated for sensor azimuth and elevation look angles. For each trajectory point, these look angles can be determined by determining the vector from the vehicle to the sensor target (inertial coordinate system), converting the resulting vector into cartesian coordinates, and then translating these cartesian coordinates into the body reference frame. For each trajectory point's resulting x, y, z coordinates $(x_{b_{sensor}}, y_{b_{sensor}}, z_{b_{sensor}})$, the sensor azimuth and elevation look angles can be calculated using the following equations.

$$\underset{az}{\delta} = \tan^{-1}\left(\frac{y_{b_{sensor}}}{x_{b_{sensor}}}\right) \tag{43}$$

$$\underset{el}{\delta} = \tan^{-1}\left(\frac{z_{b_{sensor}}}{\sqrt{{x_{b_{sensor}}}^2 + {y_{b_{sensor}}}^2}}\right) \tag{44}$$

In both of these equations, $\tan^{-1}$ is the four-quadrant inverse tangent ($atan2$). Use of this version of the inverse tangent returns an angle that is in the appropriate quadrant, with a range $(-\pi, \pi]$.

4.4 Thermal Modeling as Constraints

One of the major differences that sets hypersonic aerodynamics apart from lower speed flight are the much higher operating temperatures for a hypersonic vehicle. These temperatures must be accounted for in the development of the vehicle. In

the optimal control formulation, this environmental condition is incorporated as skin temperature limits in the path constraints. Given a vehicle geometry, Thermal Protection System (TPS) configuration and flight profile (altitude, velocity, angle of attack), LANMIN (Langley MINIVER Code)/EXITS (Explicit Interactive Thermal Structures Code) can compute vehicle surface temperatures as specified points along the vehicle. These surface temperatures are operationally limited by the vehicle's materials and design for the structure and TPS. Thus, the optimal control formulations in Section 4.2 would have an additional path constraint:

$$\text{Path Constraints } (\mathbf{C}), \forall \mathbf{t} \in [\mathbf{t_0}, \mathbf{t_f}]: \tag{45}$$

$$\varphi_{\min} \leq \varphi\left(\mathbf{x}(t), \mathbf{u}(t), t\right) \leq \varphi_{\max}$$

$$q_{\infty}\left(\mathbf{x}(t), \mathbf{u}(t), t\right) \leq q_{\infty\max}$$

$$T_k\left(\mathbf{x}\left(t\right)\right) \leq T_{k\max}$$

$$k = 1, \ldots, N_t \text{ (number of temperature locations)}$$

where $T_k\left(\mathbf{x}\left(t\right)\right)$ is computed surface temperature (for a given position on the vehicle) for the entire trajectory and $T_{k\max}$ is the maximum surface temperature at the kth position on the vehicle, specified by the user.

In this research, LANMIN/EXITS will be integrated with General Pseudospectral Optimal Control Software (GPOPS) such that, at selected vehicle locations, surface temperatures can be constrained and used as path constraints. Figure 24 shows how LANMIN/EXITS will be integrated with GPOPS. For a given candidate trajectory, LANMIN will use the candidate state to compute the flow type, heating rate/load, and local pressure for specified vehicle locations over the entire candidate trajectory. LANMIN then passes this information to EXITS, which calculates, with the user-specified vehicle TPS configuration, the skin temperatures for specified vehicle

locations over the entire candidate trajectory. This information is passed back to GPOPS which uses this data as path constraints.

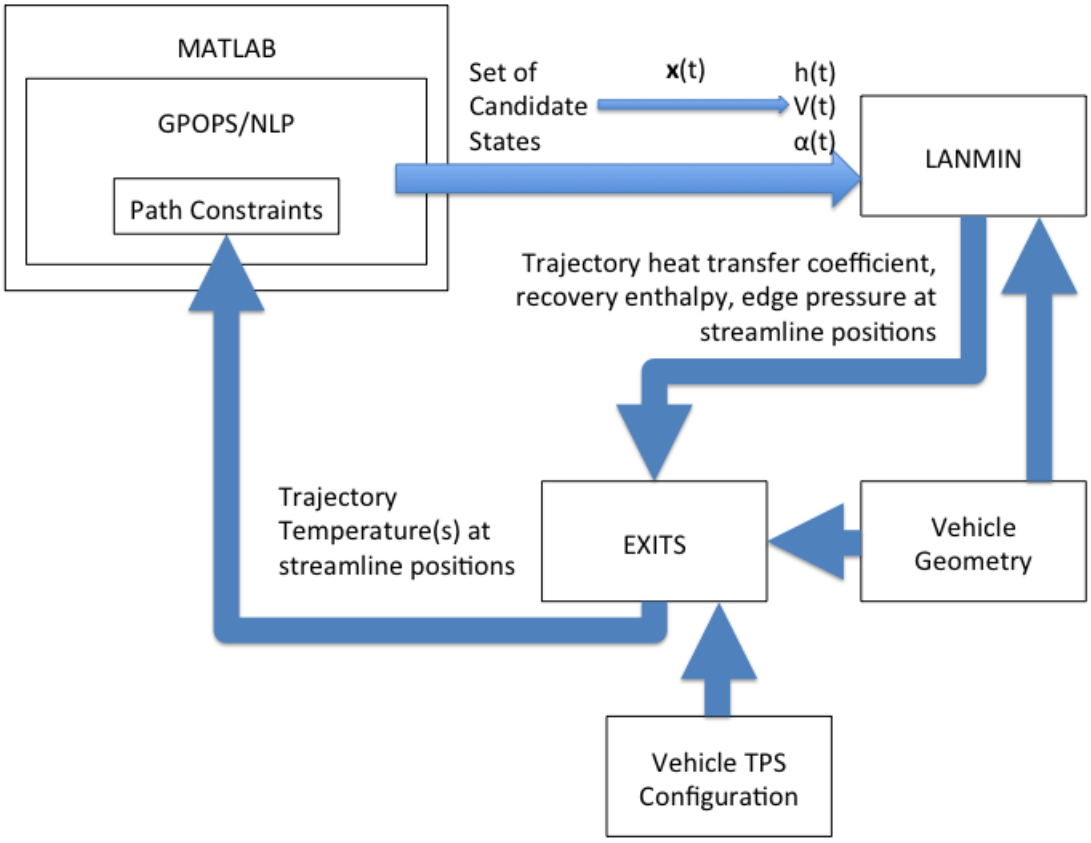


Figure 24. LANMIN/EXITS Integration with MATLAB/GPOPS

4.5 Methodology Summary

In this chapter, the optimal control formulation for three performance functionals in five different formulations were presented and discussed. As part of the discussion, the incorporation of aerothermal path constraints and sensor parameter path constraints and objective functions were described, to include the integration of MINIVER into the problem methodology. The next chapter provides the numerical solutions and detailed analysis for these formulations. The results include discussions of scenarios' flight mechanics with insight trajectory behavior. As a result, the

methodology and models used in this research are demonstrated to provide a suitable and effective mission planning capability.

V. Scenarios Results and Analysis

This chapter documents the results and analysis applying the methodology and models discussed in previous chapters. The goal of the analysis is to demonstrate results for each of the scenario specific settings, generating solutions that resemble typical flight profiles and constraints. By completing these analyses, the research objectives are met: to develop a methodology to calculate optimal trajectories for an air-breathing hypersonic vehicle with different optimality goals and path constraints, especially TPS temperature constraints.

5.1 Analysis Overview

This section provides an overview of the system used to generate the results. Also listed are the software settings used to run the scenarios.

The scenarios were run using in Matlab® version 2014a on a Mac Pro (MacPro5,1) desktop computer (24GB of memory, 3.2GHz quad-core Intel® Xeon W3565 quad-core processor) running OS X 10.9.5. General Pseudospectral Optimal Control Software-II (GPOPS-II) version 2.0 (hp-adaptive Radau Pseudospectral method) was run within Matlab®. Unless stated otherwise, the GPOPS-II settings listed in Table 3 were used to run the scenarios.

The state and control limits used in this analysis are located in Table 4. They were derived from dynamic constraint limits, limits on lookup tables, geometric constraints, and Generic Hypersonic Aerodynamic Model Example (GHAME) vehicle properties.

The following results are given for the optimal control formulations described in Chapter IV and summarized in Table 5. Unless explicitly stated, all the solutions provided met both the Non-Linear Programming (NLP) and the mesh error tolerances specified in Table 3.

Table 3. Scenario GPOPS-II settings

GPOPS-II option	Setting
setup.nlp.solver	IPOPT (v3.11.0)
setup.nlp.options.ipopt.linear_solver	ma57
setup.derivatives.supplier	sparseCD (sparse Center Difference)
setup.derivatives.derivativelevel	second
setup.derivatives.dependencies	sparseNaN (sparse Not a Number)
item setup.method	RPMIntegration
setup.nlp.options.tolerance	10^{-6}
setup.nlp.options.maxiterations	250
setup.mesh.method	hpPattersonRao
setup.mesh.tolerance	10^{-3}
maximum mesh iterations	10
min num of collocation points per interval	3
max num of collocation points per interval	10
scaling	auto (hybrid update)
measurement system	US (foot-slug-second-Rankine)

Table 4. State and Control Limits

		minimum	maximum	units
States	height	0	282000	feet
	θ	-180	180	degrees
	ϕ	-89	89	degrees
	velocity	379	8931.6	feet/sec
	γ	-89	89	degrees
	ψ	-180	180	degrees
	mass	120000	300000	lb mass
Controls	α	-3	21	degrees
	σ	-45	45	degrees
	$\dot{m}_f$	20	360	lb mass/sec
Rate Controls (if used)	$\dot{\alpha}$	-6	6	degrees/sec
	$\dot{\sigma}$	-6	6	degrees/sec

Table 5. Scenario Attributes Summary

Formulation Attributes		Sec 5.2.1	Sec 5.2.2	Sec 5.2.3	Sec 5.2.4	Sec 5.2.5	Sec 5.2.6	Sec 5.2.7	Sec 5.2.8	Sec 5.2.9	Sec 5.2.10
OBJECTIVE FUNCTION	Maximum Range	x									
	Minimum Time		x		x					x	
	Minimum Time with Control Penalty			x		x	x	x	x		x
BOUNDARY CONDITIONS	Start Fixed (Takeoff)	x	x	x	x	x	x	x	x	x	x
	End Variable (Overflight)	x									
	End Fixed (Overflight)		x	x					x	x	
	End Fixed (Landing)				x	x	x	x			x
	Waypoint Variable (Refuel)						x	x	x		x
	Waypoint Variable (sensor on)								x		
PATH CONSTRAINTS	Fuel-Air Ratio	x	x	x	x	x	x	x	x	x	x
	Dynamic Pressure	x	x	x	x	x	x	x	x	x	x
	Mach Number									x	
	g-limits							x			
	No-fly zone							x	x		
	Sensor								x		
	Temperature									x	x

As described in previous chapters, the vehicle under consideration in this research is specified to have a maximum Mach number around Mach 8. While this Mach number can be a path constraint, it drives a sine wave-like behavior in the cruise altitude and/or velocity. To remove this behavior, a height (altitude) limitation (around 130,000 feet) could be introduced but can constrain the optimal solution. Instead, a maximum velocity is determined by using a desired approximate maximum Mach number and a speed of sound from an approximate cruise altitude. Using this approach, the vehicle does not have a direct constraint on cruise altitude; instead, it has a constrained maximum velocity. The only scenarios which do not use this approach are Sec 5.2.1 and Sec 5.2.9. The first scenario still uses the height constraint since removing the height constraint resulted in a height profile with severe cyclic fluctuations. The second scenario uses Sparse Nonlinear Optimizer (SNOPT) as the NLP solver and uses Mach number as a state instead of velocity. With these changes, the scenario converges significantly faster than using Interior Point OPTimizer (IPOPT).

In each scenario, only a subset of the scenario result figures are included in this chapter. All of the result figures for each scenario are provided in Appendix C.

In each of the figures in the subsequent sections, the results are plotted against relevant independent variables, which is typically time. The vertical hash marks in each of the plot curves represent the collocation points in the optimal solution. Most of the figures are self explanatory except for two charts. The first one is Mesh Tolerance and Maximum Relative Error chart, which shows the maximum relative error of the NLP solution for each mesh iteration (e.g., Figure 31). The trend over mesh iterations shows how quickly the ultimate solution converges to meet the user-specified mesh tolerance. The other chart is the Mesh Interval and Collocation Point History chart (e.g., Figure 43), which shows, for each mesh iteration, the mesh interval and collocation point history as a stem plot. It shows the distribution of the mesh

intervals with each stem point defining an interval end point (except the first point that shows zero collocation points). The height of each stem point shows the number of collocation points in that interval. Each mesh interval shows the interval solution, but since each solution could have a different total time, the times are normalized such that the sum of the intervals in each phase is always unity.

In each of the scenarios, there are transients in the optimal trajectory solutions, especially evident in the control subfigures. These transients occur when the vehicle is around Mach 2 and again at around Mach 6. This corresponds to the propulsion system transition from turbine to ramjet modes and then from ramjet to scramjet modes. At these Mach numbers, the lookup tables are relatively sparse, and even though spline fits are used with the lookup tables, this sparsity results in quickly varying trajectory parameters. If the GHAME lookup tables had more refined data points in these regimes, this phenomena would be reduced.

5.2 Scenario Results and Discussion

5.2.1 Maximum Range (Climb and Cruise) Scenario.

This scenario is used to determine the nominal maximum range for this vehicle under specific conditions using the optimal control formulation defined by Equation 34. It assumes that the entire flight will occur along a latitude line, so that measuring the ground path distance is a function of the initial and final longitudes, as seen in Figure 25. The vehicle is flown starting at the equator and given an initial heading along the equator so that the vehicle does not have to turn. As a result, the limits for the bank angle control (σ) were set to 0, as well as the heading angle (ψ). In this scenario, the minimum propellant mass flow rate was set to $40 lb_{mass}/\,\mathrm{sec}$ to achieve a smooth trajectory. The vehicle flew over the equator until all the fuel was spent. The resulting maximum range was 5584 miles.

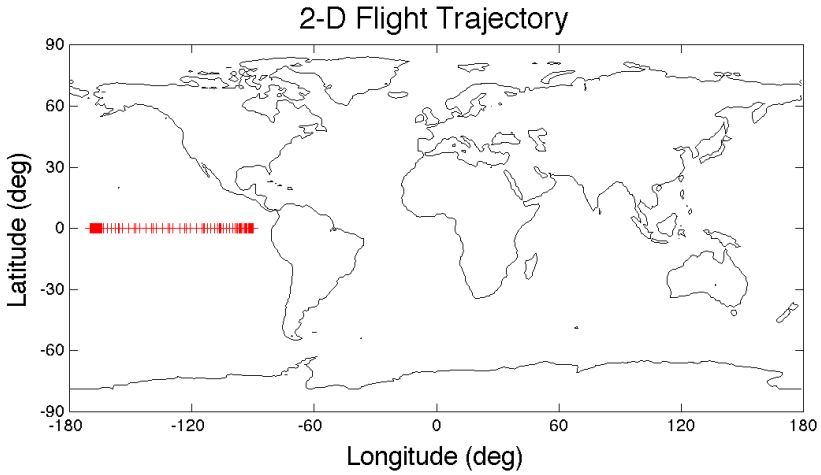


Figure 25. Optimal Trajectory: Maximum Range, 2-D Flight Trajectory

The top half of Figure 26 shows the height and velocity history for the solution and Figure 27 shows the ascent portion of the trajectory in more detail. Like many other scenarios in this chapter, the vehicle quickly climbs and then cruises until it overflies a specified location. There is an oscillation in the height curve which coincides with the Mach 6 scramjet transition ($t \approx 172$ sec and $V \approx 6114$ mph or ≈ 5313 kts) and acceleration to Mach 8, which has been seen in several scenarios and results from use of the GHAME data set. The height reduction at the end of the scenario results when the vehicle approaches its dry weight and to maintain altitude, cannot reduce the fuel-air ratio below a specified lower limit. It instead reduces altitude so that it operates at the minimum specified fuel-air ratio. This behavior can be eliminated by assuming a lower minimum propellent flow rate. Since the minimum propellent flow rate was not specified in the GHAME source documents, there is some flexibility in setting this parameter.[1]

[1]The numbers used are derived from the AFRL's DOF36 trajectory tool [82].

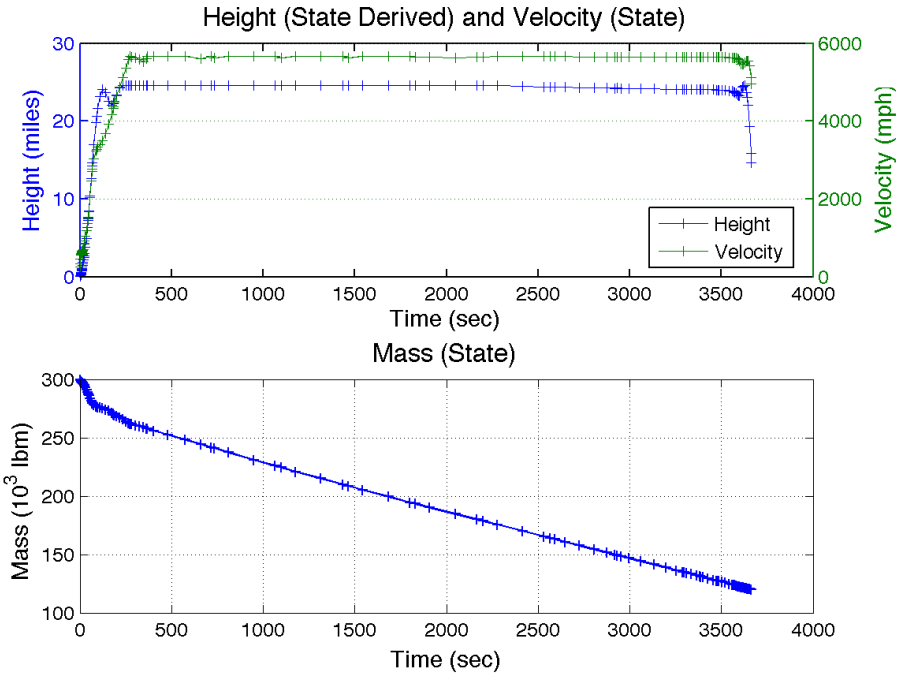


Figure 26. Optimal Trajectory: Maximum Range, States

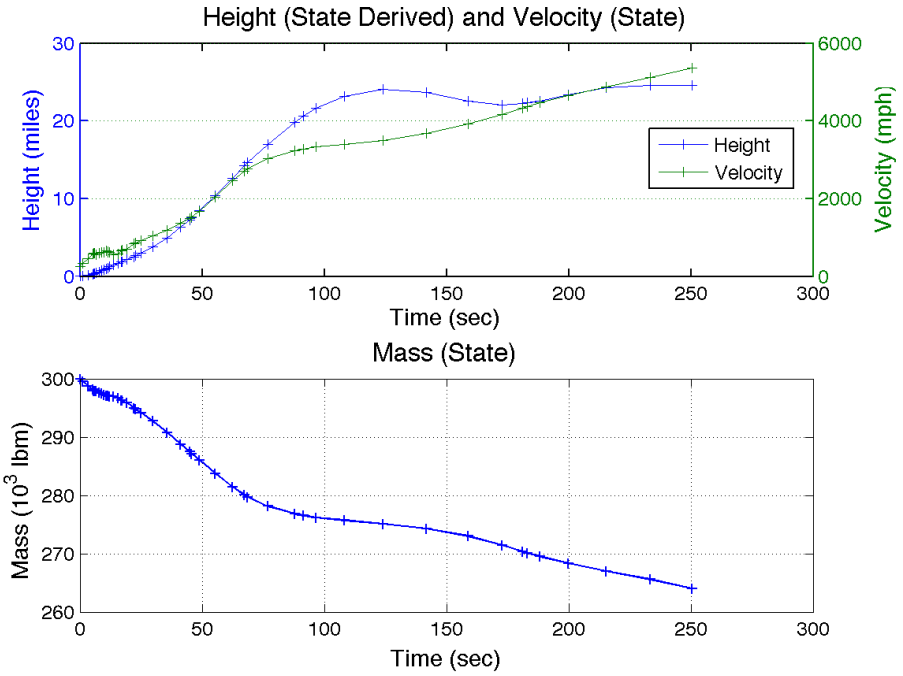


Figure 27. Optimal Trajectory: Maximum Range, States (Ascent Portion)

Figure 28 shows the oscillations in the α control that results in oscillations in the height state described above and Figure 29 ascent portion of the trajectory in more detail. As shown in the propellant mass flow rate control and path constraint plots (Figures 28 and 30), the vehicle operates at near maximum flow rate for the ascent and then during cruise, the vehicle operates at near minimum values. The upper plot in Figure 30 shows the path constraints for this solution. The fuel-air ratio $\varphi_{\max} = 2$ becomes an active constraint during the ascent phase only and exhibits oscillations after the vehicle ascent due to the vehicle wanting to operate at a propellent mass rate less than than the lower control limit. Like all the scenarios used in this chapter, the dynamic pressure $q_{\infty\,\max} = 2000$psf never becomes an active constraint. As seen in Chapter VI, dynamic pressure limits are experienced only for very short duration flights.

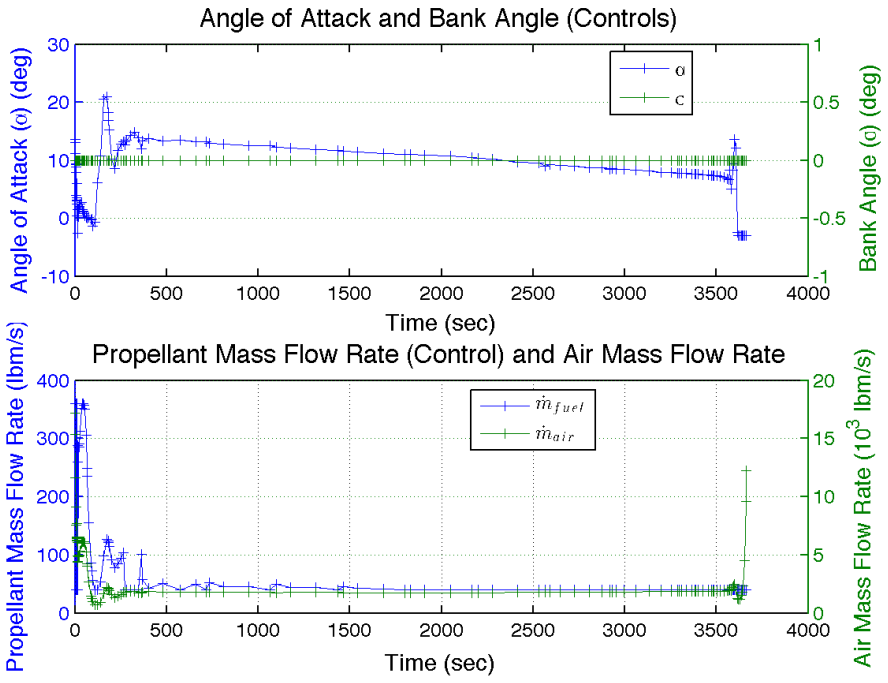


Figure 28. Optimal Trajectory: Maximum Range, Controls

As seen in Figure 31, the Mesh Maximum Relative Error[2] default (10^{-3}) is not

[2]As discussed by Darby [20], the mesh relative error is an measurement of the approximation

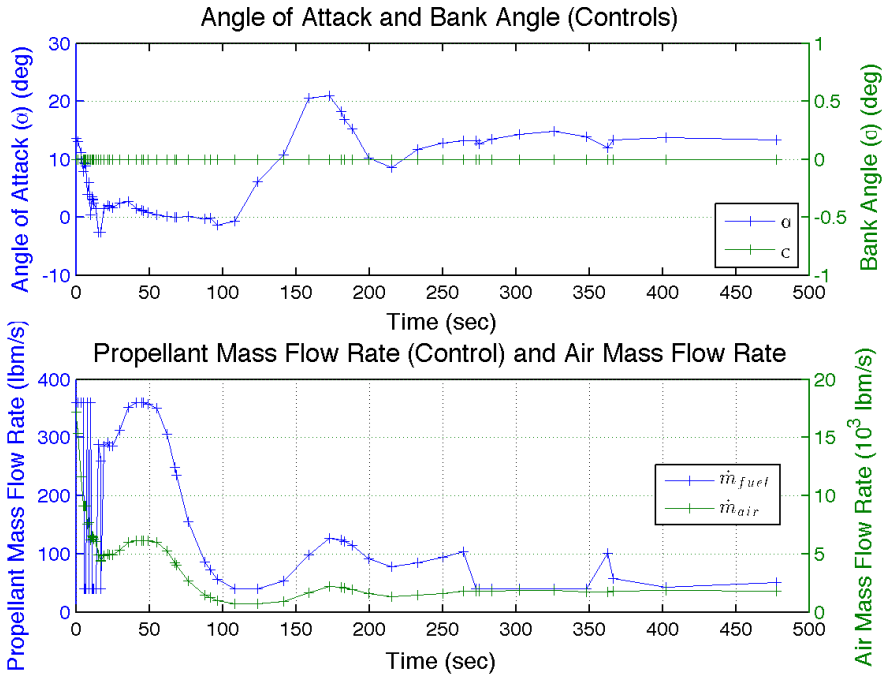


Figure 29. Optimal Trajectory: Maximum Range, Controls (Ascent Portion)

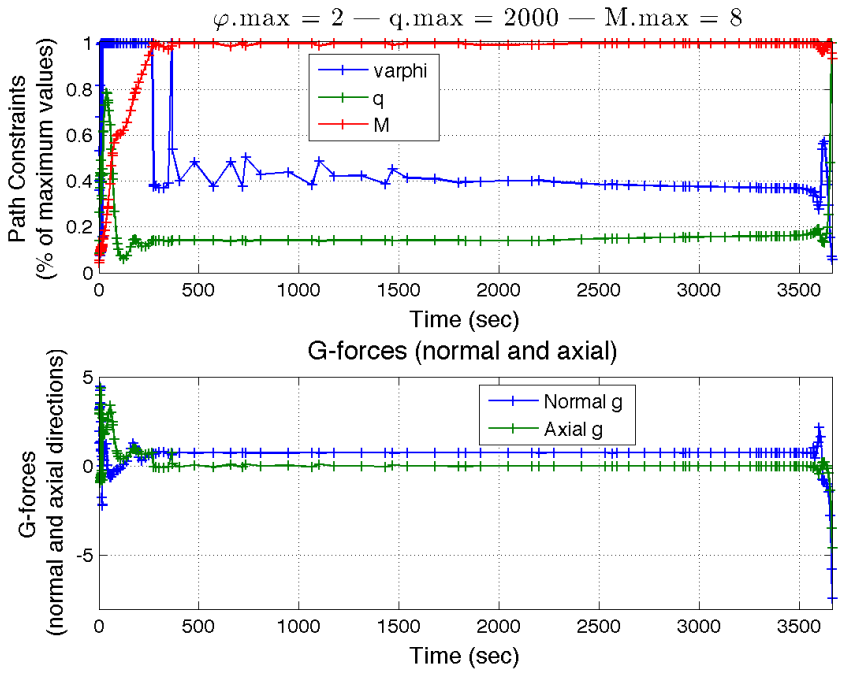


Figure 30. Optimal Trajectory: Maximum Range, Path Constraints and g's

met but the general trend is leading towards meeting this value. While the maximum number of mesh iterations can be increased, the chance that error will not decrease further (or even increase) becomes more prevalent because each mesh iteration increases the number of collocation points and/or intervals. As seen during test runs, having too many collocation points can result in non-convergence. This is the consequence of using lookup tables instead of smoother polynomial curve fits for the aerodynamic and propulsion models. The trend of the Maximum Relative Error illustrates the exponential convergence quality of a spectral method solution.

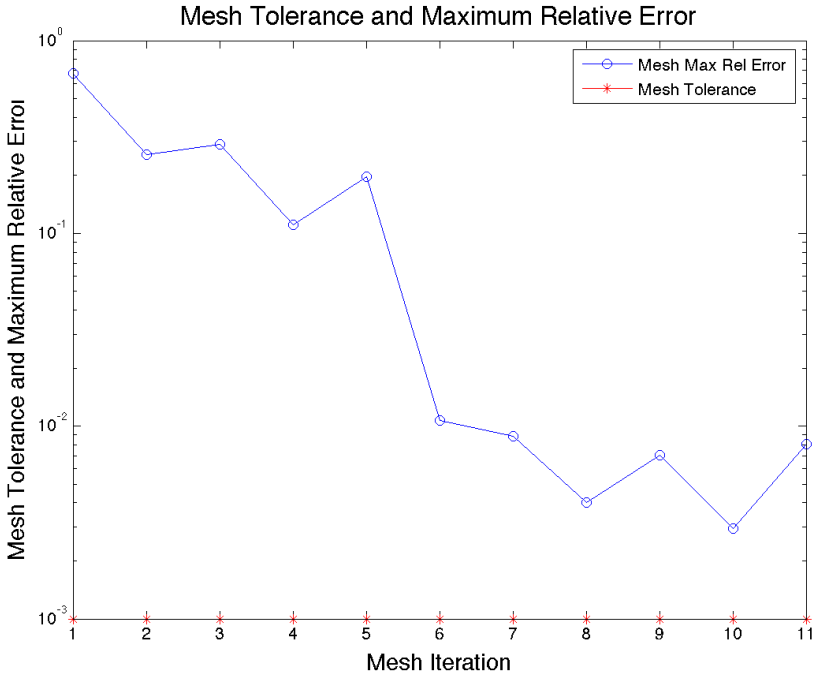


Figure 31. Optimal Trajectory: Maximum Range, Maximum Relative Error

error in each interval in the optimal solution. To measure this error, GPOPS determines how well the dynamic constraints (equations of motion) are satisfied between collocation points in the optimal solution. To interpolate between collocation points for each state, the optimal solution's state Lagrange polynomials are used. To interpolate between collocation points for each control, Lagrange polynomials are approximated for each control. The dynamic constraints are computed at these interpolation points and the maximum value of the dynamic constraints is the mesh maximum relative error.

5.2.2 Minimum Time (Climb and Cruise) Scenario.

This scenario is for a minimum time trajectory from California to Maine. The vehicle takes off from a California runway, climbs and turns to its destination, then cruises until it overflies the Maine runway, as seen in Figure 32. Since this is a minimum time problem, Figure 33 shows that after the vehicle leaves the runway, it quickly gains altitude and turns towards the destination. Once it reaches cruise altitude, it cruises at its maximum velocity until it overflies the destination (unlike the takeoff from a California runway, there is no enforced altitude at which the vehicle overflies the Maine runway).

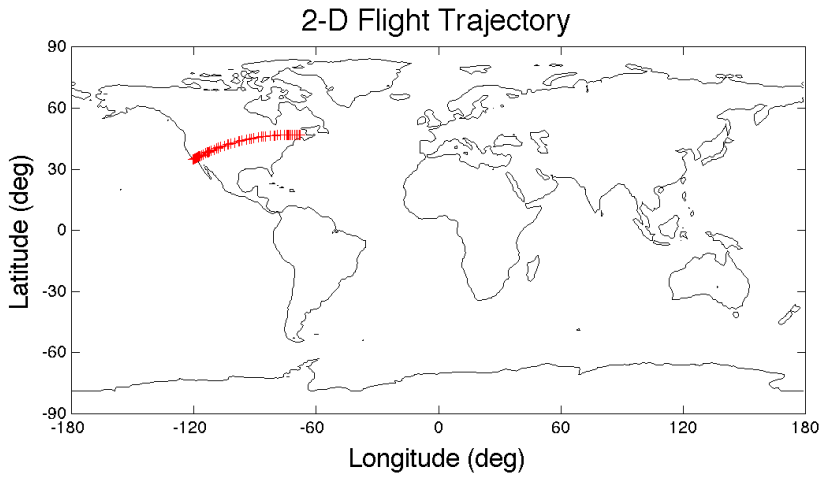


Figure 32. Optimal Trajectory: Minimum Time (Climb and Cruise), 2-D Flight Trajectory

Figure 34 shows a typical profile for a climb and cruise scenario. After takeoff, both the height and velocity rapidly increase until they transition to a roughly level profile for the remainder of the flight. The vehicle mass quickly reduces due to the large fuel usage during ascent but once maximum velocity is reached, the vehicle

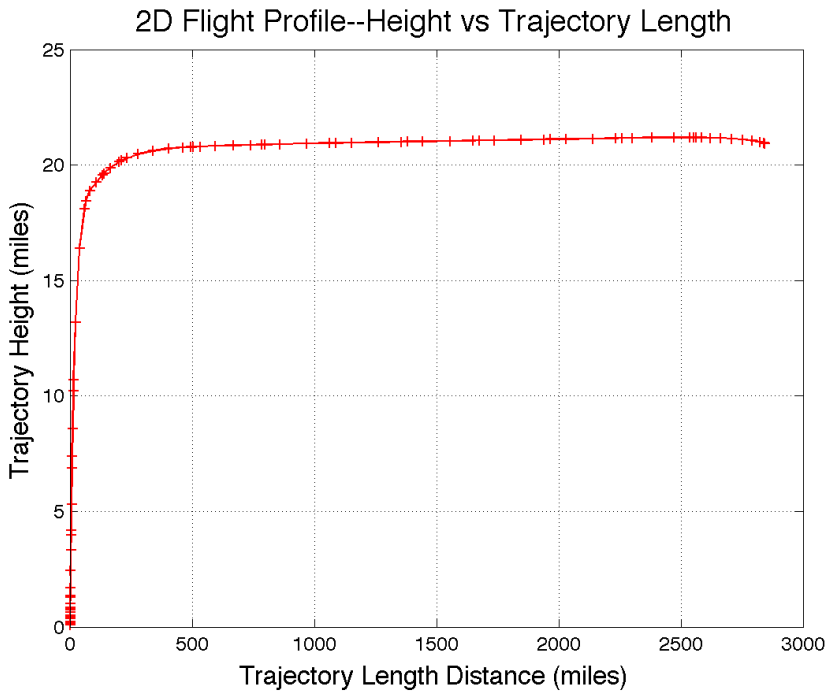


Figure 33. Optimal Trajectory: Minimum Time (Climb and Cruise), 2-D Flight Profile

throttles back to a constant fuel usage for the remainder of the flight, such that all the fuel (180,000 lbm) is used by the end of the scenario.

Figure 35 shows that the majority of control usage occurs at the start of the scenario to get the vehicle to cruise. Again, as seen in the previous section, there is a small α transient around the Mach 6 transition point ($t \approx 76$ sec), as shown in Figure 36.

The top of Figure 37 shows that for a majority of the trajectory, except during the initial climb, there are no active path constraints. During the rapid ascent, the active path constraint is φ due to the rapidly decreasing air mass flow, as well as the lookup table limit on φ. The bottom of Figure 37 shows a typical g's profile for a climb and cruise scenario, where the highest g's are experienced in the climb flight portion. There are some small transients that occur around the Mach 6 transition point ($t \approx 41$ sec), similar to α control.

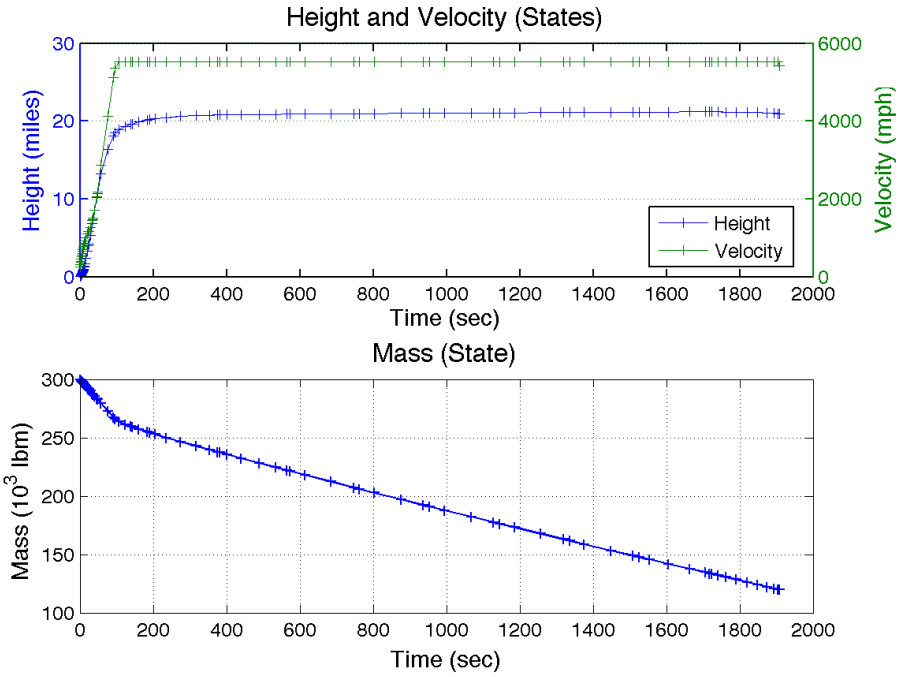


Figure 34. Optimal Trajectory: Minimum Time (Climb and Cruise), States

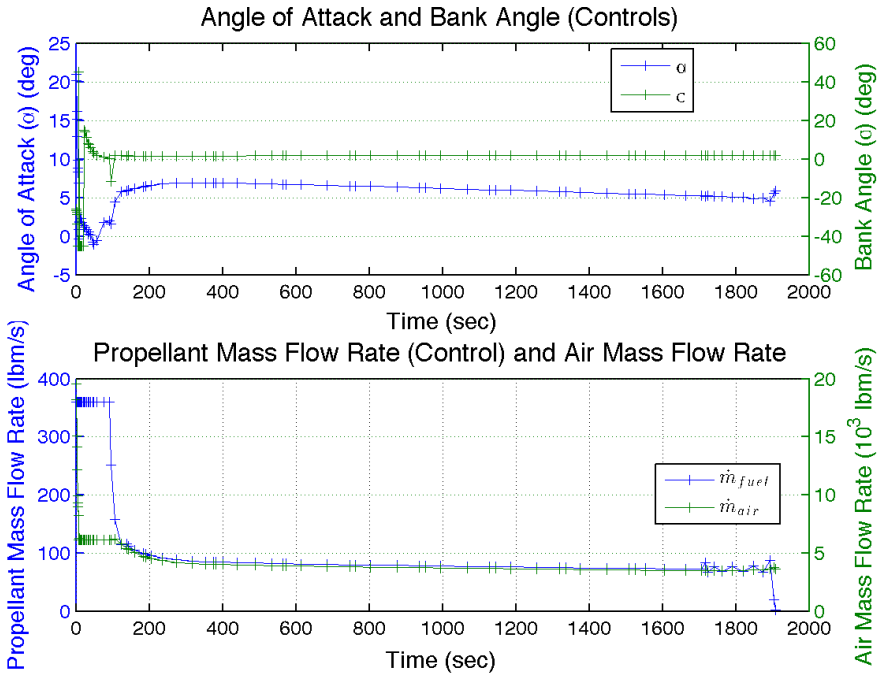


Figure 35. Optimal Trajectory: Minimum Time (Climb and Cruise), Controls

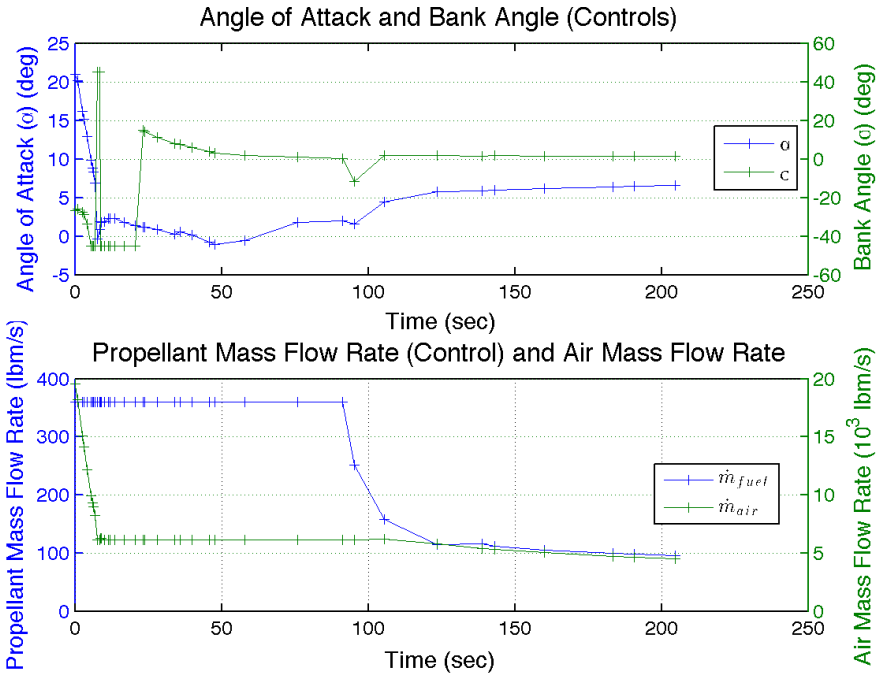


Figure 36. Optimal Trajectory: Minimum Time (Climb and Cruise), Controls (Ascent Portion)

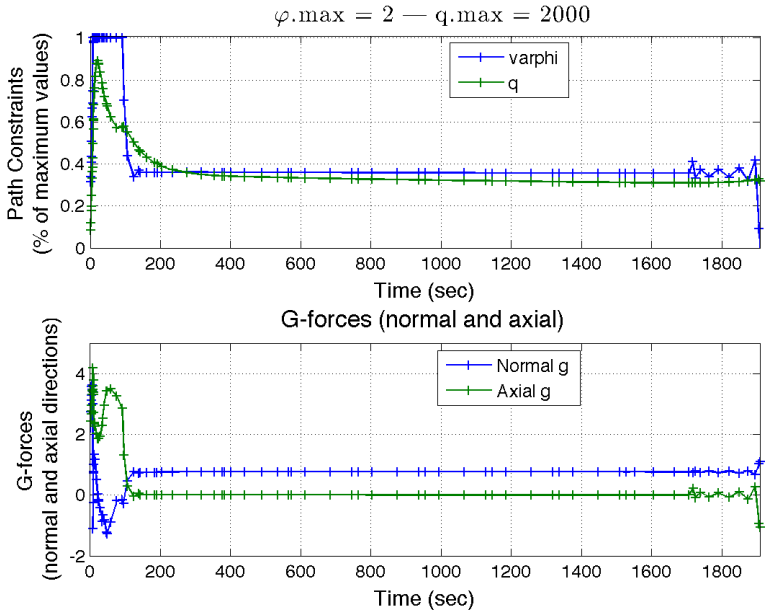


Figure 37. Optimal Trajectory: Minimum Time (Climb and Cruise), Path Constraints and g's

Figure 38 shows the heating rates for the scenario. Three heating rates are shown: body, nose (MINIVER (Miniature Version) calculated), and nose (Chapman's Equation, Equation 1 calculated). Typical for this vehicle, the body (windward fuselage, half way down vehicle) heating is much lower than at the nose. The maximum values for nose heating rate occur when the vehicle reaches cruise velocity but is still climbing. Once cruise altitude is reached, the heating rate reduces to near steady state, but slightly decreasing during the flight due to the slightly increasing cruise altitude, resulting in decreasing atmospheric density. For this scenario, the difference between the nose heating rate from MINIVER and Chapman's equation is around 25 percent for level flight, but can be much higher in quickly changing flight regimes.

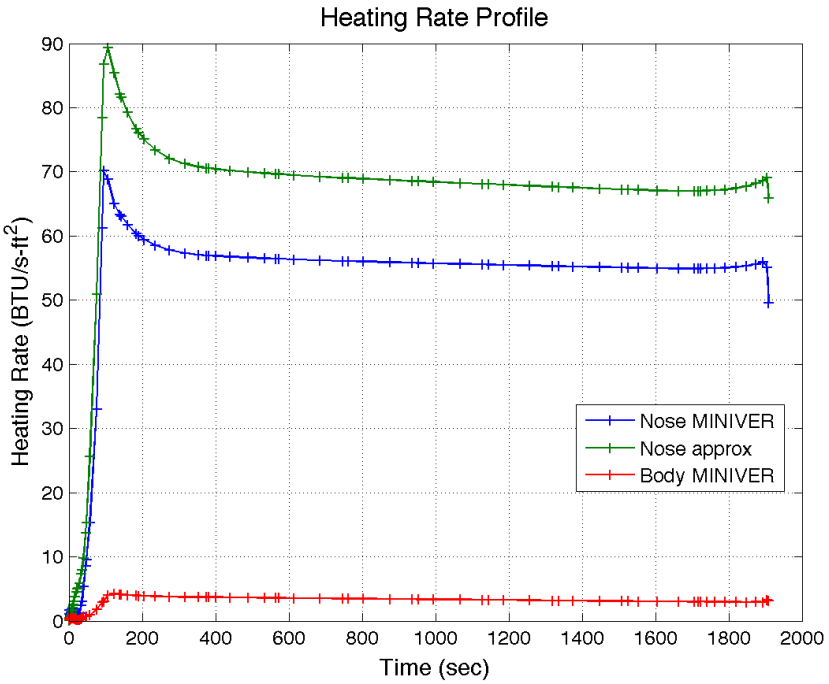


Figure 38. Optimal Trajectory: Minimum Time (Climb and Cruise), Heating Rates

Figure 39 shows the heating loads for the flight profile. Since heating loads are just the integrals of heating rates (Equation 2), these curves are relatively linear since heating rates are nearly constant for most of the flight profile.

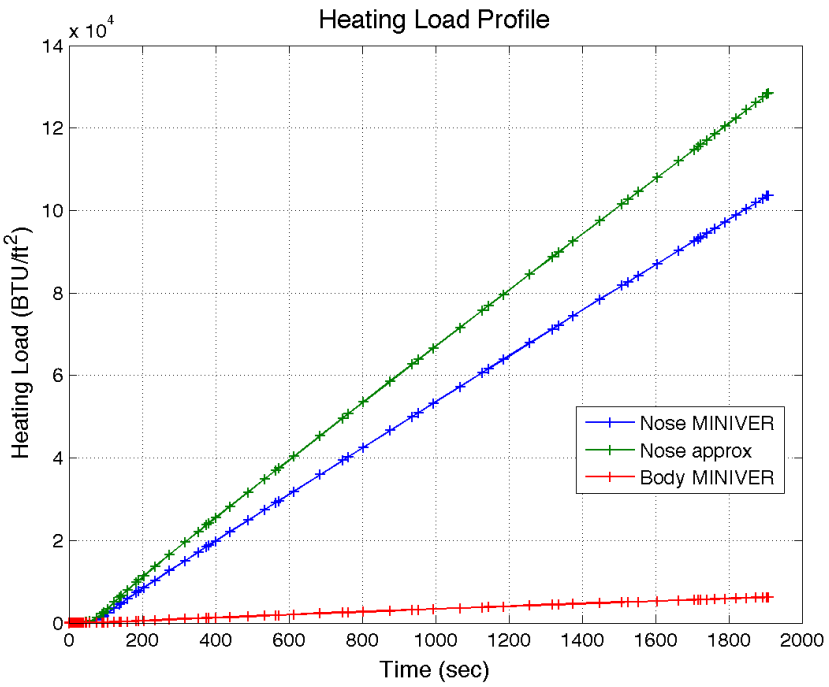


Figure 39. Optimal Trajectory: Minimum Time (Climb and Cruise), Heating Loads

Figure 40 shows the temperature profile at the nose and body, calculated by MINIVER. For each fuselage position, two values are given, the outer and inner surfaces of the vehicle skin. Since the body temperatures increase more slowly, the outer and inner surfaces at the body position tend to be roughly equivalent. The nose, on the other hand, has drastic temperature increases during the ascent and early cruise phase, until thermal equilibrium is reached. Once this equilibrium is reached, it is maintained until the end of flight.

As seen in Figure 41, Maximum Relative Error default (10^{-3}) is not met but the general trend is leading towards meeting this value. As mentioned previously, the number of mesh iterations can be increased at the risk of the solution diverging.

Figure 42 shows the distribution of collocation point intervals for each mesh iteration, using a normalized time. This figure shows how the collocation point mesh evolves over time, where GPOPS-II adds either intervals and/or collocation points

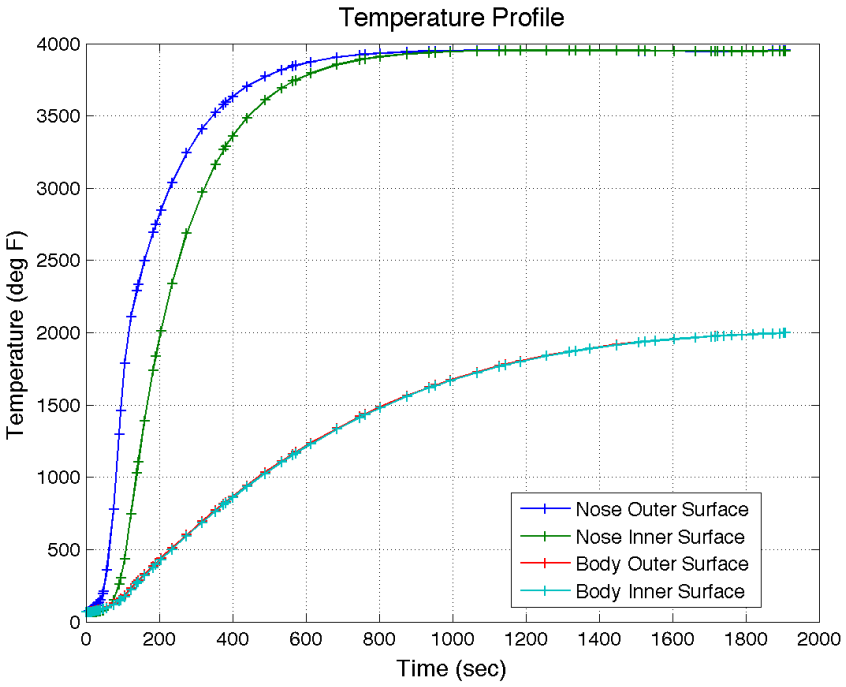


Figure 40. Optimal Trajectory: Minimum Time (Climb and Cruise), Temperatures

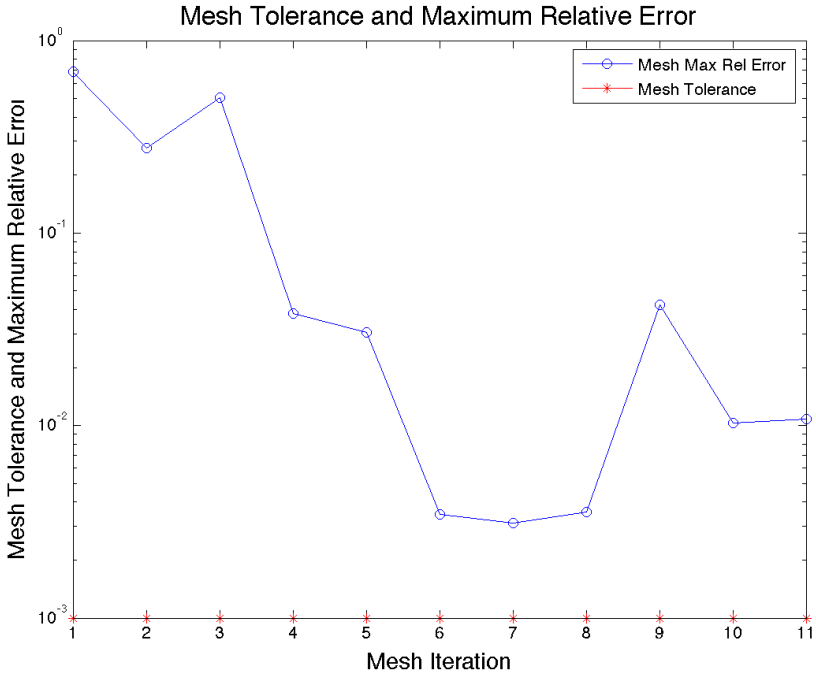


Figure 41. Optimal Trajectory: Minimum Time (Climb and Cruise), Maximum Relative Error

between successive mesh iterations. The blue circles are the collocation point interval boundaries and the red squares are the phase boundaries (only 1 phase in this scenario). As seen in the figure, additional intervals are added mainly at the beginning of the scenario, in the climb phase, since this is where most of the dynamics are occurring.

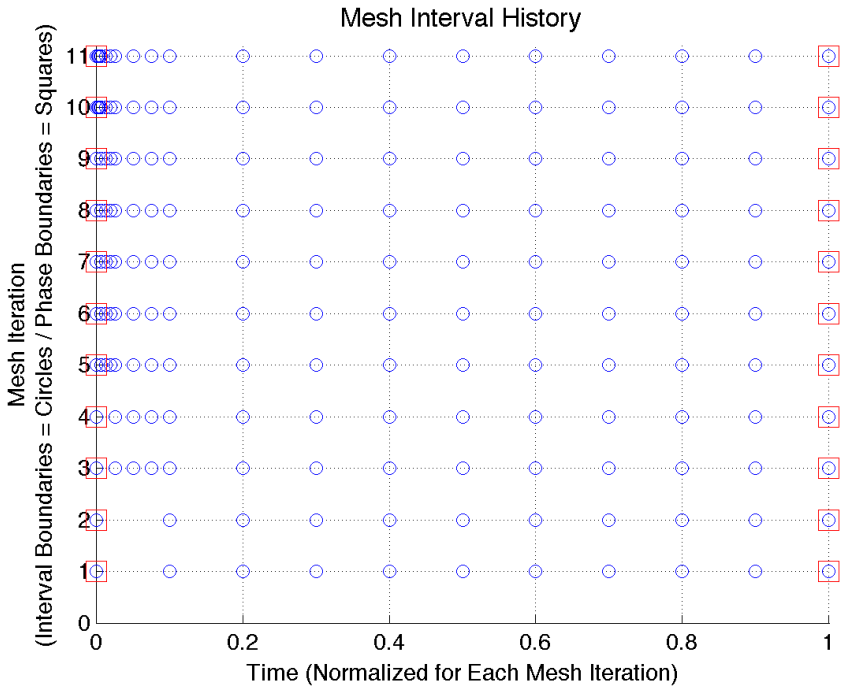


Figure 42. Optimal Trajectory: Minimum Time (Climb and Cruise), Mesh Interval History

Figure 43 shows the evolution of both the mesh interval sizes and the number of collocation points per interval, using normalized time for all the mesh solutions. This is similar to Figure 42 but this figure also includes the number of collocation point for each interval. In this scenario, additional intervals and collocation points are added during the climb portion of the profile, where most of the dynamics occur. During most of the cruise portion, the intervals and collocation points remain static but at the end of the phase additional collocation points are added to reflect the dynamics at the end of flight where there are some transients in propellant mass flow and α

control.

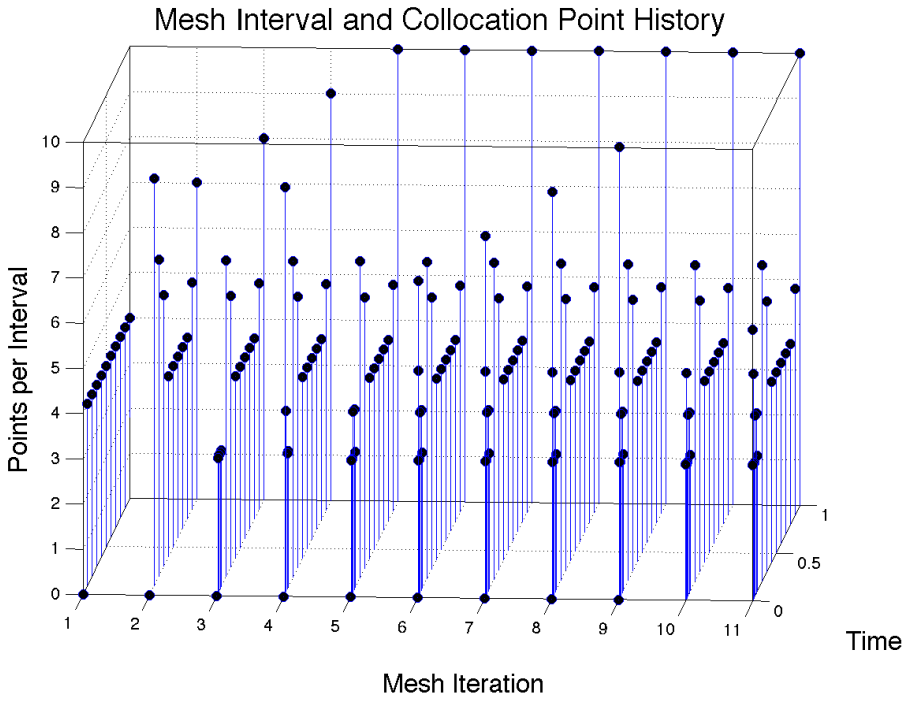


Figure 43. Optimal Trajectory: Minimum Time (Climb and Cruise), Mesh Interval and Collocation Point History

5.2.3 Minimum Time with Control Penalty (Climb and Cruise) Scenario.

This scenario is very similar to the previous minimum time scenario (Section 5.2.2) except that a control usage penalty is added and the number of maximum mesh iterations is increased to 15 (this scenario does not converge to the default mesh tolerance within 10 iterations). As shown in Section 4.2.3, ϵ is the convex combination coefficient that is used to specify the contributions of both the time and control usage for the objective function. In this scenario, the angular rate controls have an equal contribution to the control penalty and ϵ is set to 10^{-1}, favoring the control penalty over the flight time. Since for this scenario the flight time is much larger than the control penalty, ϵ makes them four orders of magnitude apart.

Like the previous scenario, the vehicle takes off from a California runway, climbs and turns to its destination, then cruises until it overflies the Maine runway. Since this is a minimum time problem, Figure 44 shows that after the vehicle leaves the runway, it quickly gains altitude.

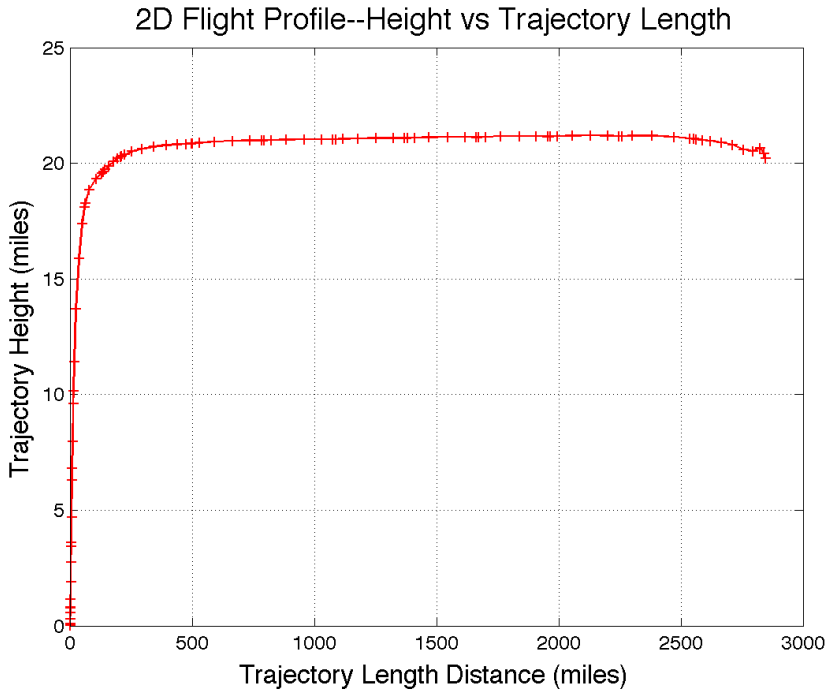


Figure 44. Optimal Trajectory: Minimum Time with Control Penalty (Climb and Cruise), 2-D Flight Profile ($\epsilon = 10^{-1}$)

With the selected ϵ, the resulting trajectory is relatively smooth. If ϵ is reduced to 10^{-3}, the resulting optimal trajectory has an oscillation in the altitude during cruise, as seen in Figure 45, resulting from oscillations in the $\dot{\alpha}$ control. This is a similar behavior to that shown by Zipfel in his vehicle simulation using GHAME data sets [92].

With the addition of the control penalty, the α and σ profiles are fairly similar to the scenario with no control penalty except during the ascent where the control penalty α and σ profiles are smoother, as shown in Figures 46 and 35.

In this scenario, the angular values α and σ are added as to states and their

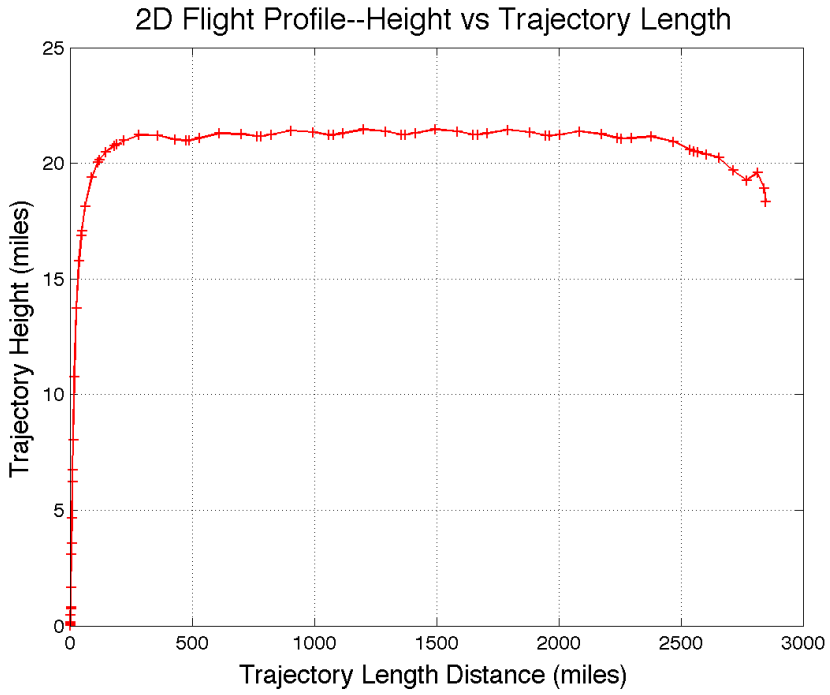


Figure 45. Optimal Trajectory: Minimum Time with Control Penalty (Climb and Cruise), 2-D Flight Profile with $\epsilon = 10^{-3}$

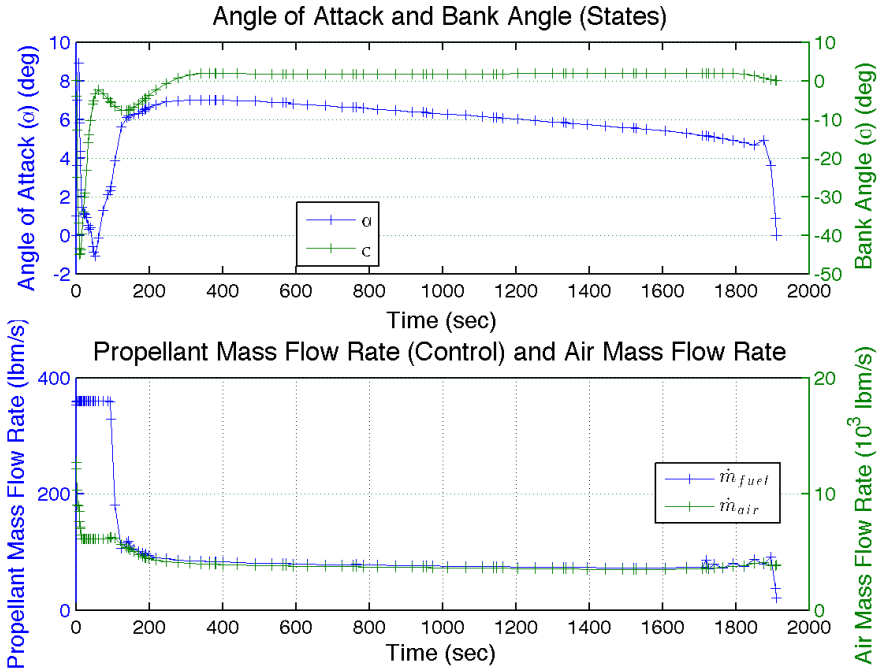


Figure 46. Optimal Trajectory: Minimum Time with Control Penalty (Climb and Cruise), Controls

derivatives $\dot{\alpha}$ and $\dot{\sigma}$ are now the controls, with propellant mass flow rate. As seen in Figure 47, there is significant control action at the beginning of the scenario but minimal control action during the cruise phase.

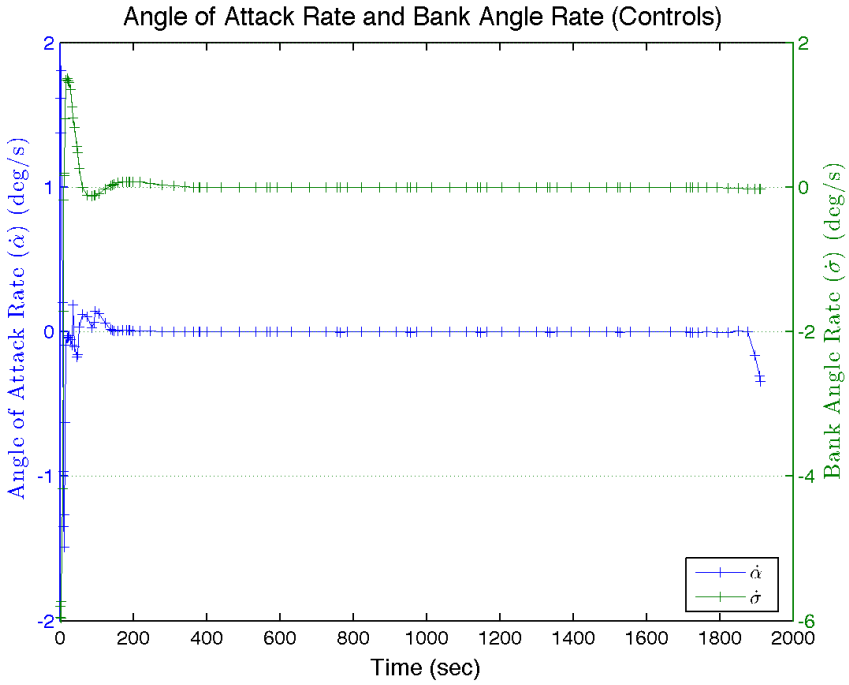


Figure 47. Optimal Trajectory: Minimum Time with Control Penalty (Climb and Cruise), Controls (angular rate)

As seen in Figure 48, the nose heating rates are similar to the scenario without the control penalty.

5.2.4 Minimum Time (Climb, Cruise, and Land) Scenario.

This scenario is similar to Section 5.2.2 but instead of overflying the runway in Maine, it lands on the runway, as seen in Figure 49. Since this is a minimum time problem, this figure shows that after the vehicle leaves the runway, it quickly gains altitude and turns towards the destination. Once it reaches cruise altitude, it cruises at its maximum velocity until it descends to land at the destination runway. At both the departure and destination runways, runway location, height and heading

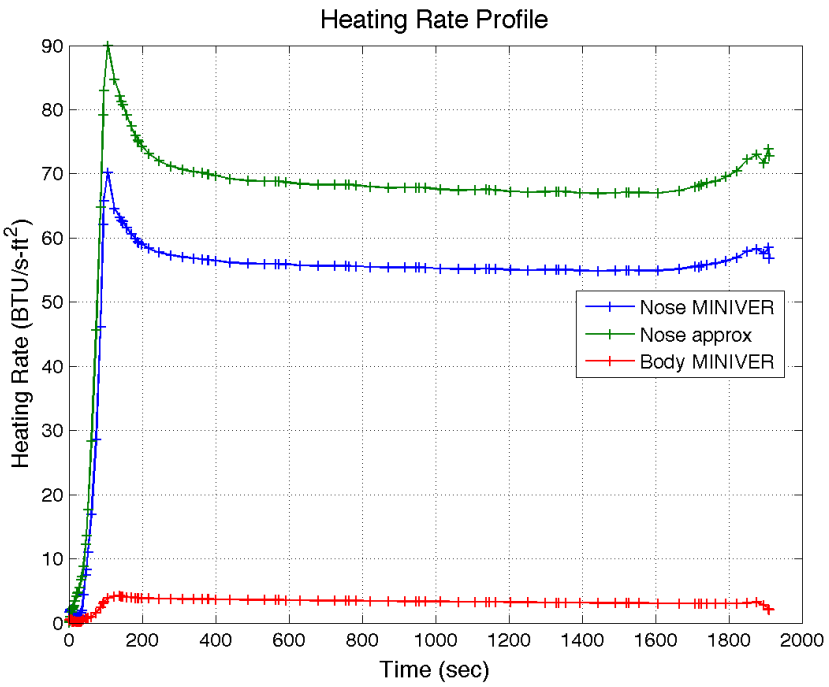


Figure 48. Optimal Trajectory: Minimum Time with Control Penalty (Climb and Cruise), Heating Rates

are enforced, in addition to vehicle takeoff velocity and mass. The climb height and velocity profiles appear to be somewhat similar but as will be shown later, the control profiles are very different.

Figure 50 shows the angular state profiles. During the climb portion of the scenario, there are significant changes in the γ and ψ states, to reach cruise altitude and heading quickly. Again, at the destination, there are significant changes since boundary conditions for γ and ψ are enforced.

In Figure 51, there is significant control action during the climb, which has been seen in all the previous scenarios. During the descent to landing, there is even a more significant 'bang-bang' control action as the vehicle generates more drag to reduces both kinetic and potential energy. This control action, called chattering [86] [85], allows the vehicle to reduce energy by not changing its aerodynamic configuration (e.g., deploying speed brakes).

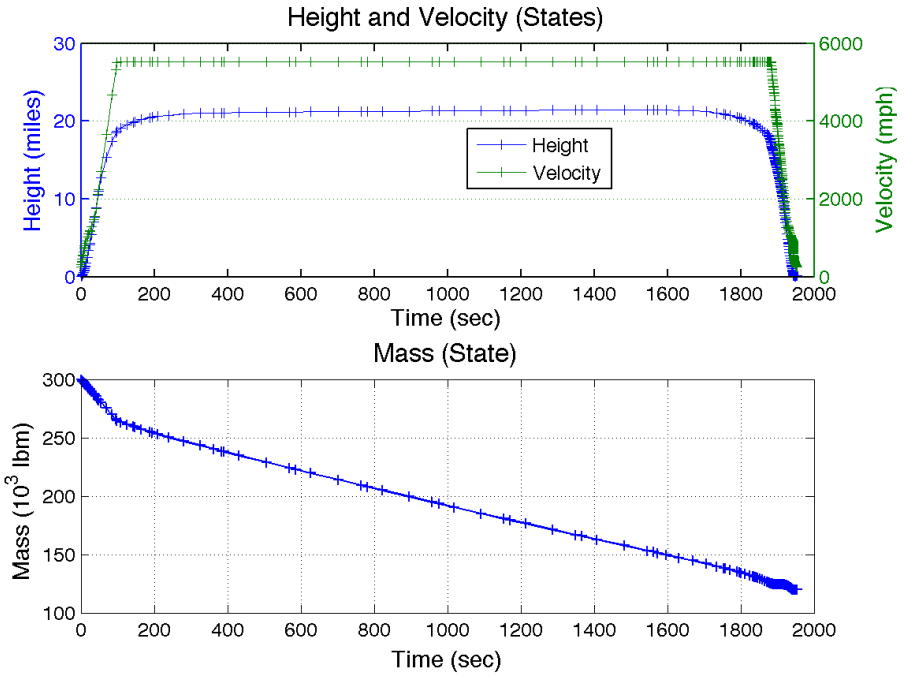


Figure 49. Optimal Trajectory: Minimum Time (Climb, Cruise, and Land), States

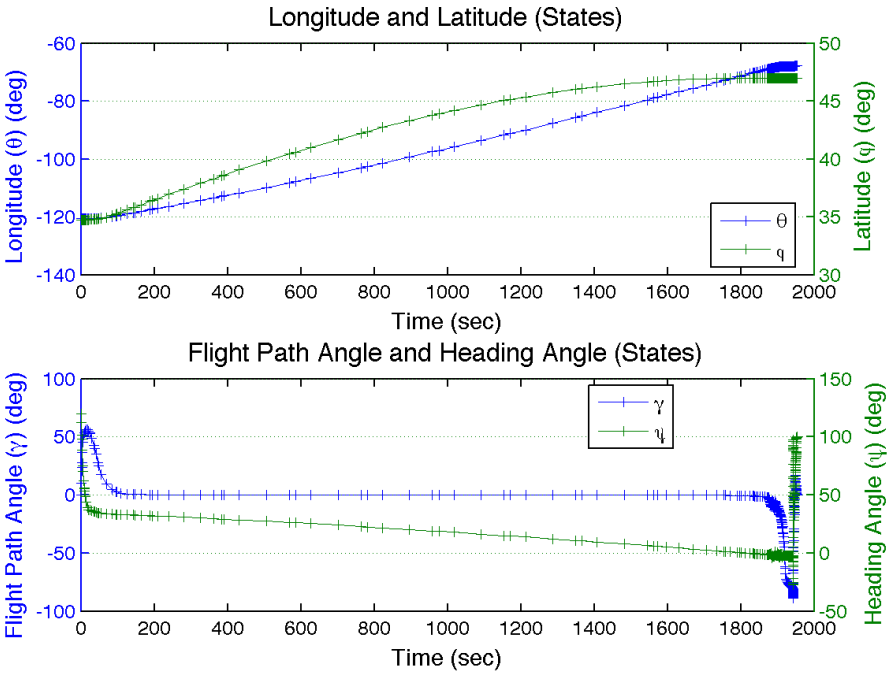


Figure 50. Optimal Trajectory: Minimum Time (Climb, Cruise, and Land), States (angular)

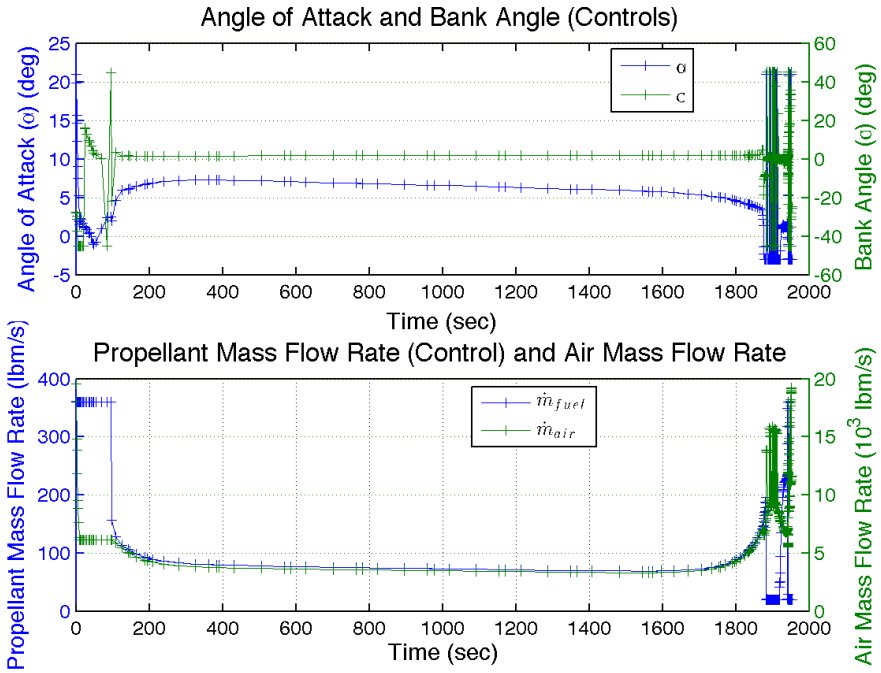


Figure 51. Optimal Trajectory: Minimum Time (Climb, Cruise, and Land), Controls

In Figure 52, as seen previously, φ is an active constraint during the climb portion. Now, during the descent portion, the dynamic pressure becomes a path constraint. Even with the constraint, the (unconstrained) g's experienced during the descent are an order of magnitude larger than previously experience. For this scenario, a g's path constraint would have to be added.

In Figure 53, the heating rates are similar to previous scenarios until the descent portion, where the heating quickly increases for a short period until the vehicle starts slowing down from its Mach cruise number as it continues its descent into the atmosphere. Maximum MINIVER-calculated heating rate occurs at $M = 8.2$ and $h = 91000$ft. This scenario also exposes one of MINIVER's limitations. LANMIN (Langley MINIVER Code), MINIVER's front end to calculate the flowfield for a given trajectory, is limited to 1000 trajectory points (up to 3 evenly spaced groups), so there is a limit to the fidelity of LANMIN calculation. In this descent scenario, the

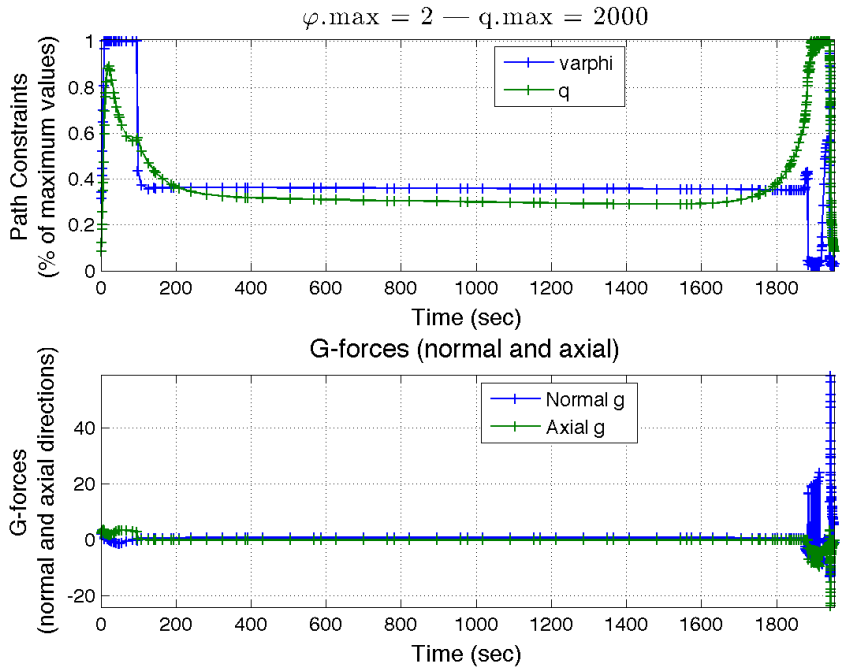


Figure 52. Optimal Trajectory: Minimum Time (Climb, Cruise, and Land), Path Constraints and g's

LANMIN grid points are large enough that with the dynamics in this flight regime, heating goes below zero which is not possible. Since this only happens for a very short time period, it does not affect the final temperatures very much.

In Figure 54, the heat loads are similar to previous scenarios except that during the descent as the heat rates increase the heating load curve slope increases. Once the heating rates decrease, the heating loads level out until landing.

In Figure 55, the temperature profiles are similar to previous scenarios except that during the descent as the heat rates increase the temperatures increase for a short time. Once the heating rates decrease, the temperatures start to decrease since the structure is radiating more heat than it takes in from the flowfield.

As seen in Figure 56, Maximum Relative Error default (10^{-3}) is not met but the general trend is leading towards meeting this value. As mentioned previously, the number of mesh iterations can be increased at the risk of the solution diverging.

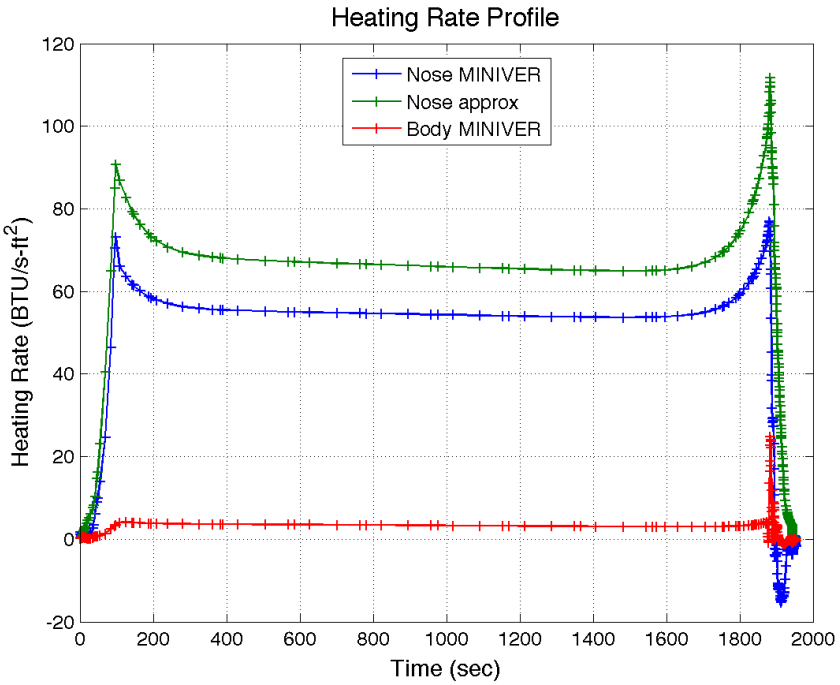


Figure 53. Optimal Trajectory: Minimum Time (Climb, Cruise, and Land), Heating Rates

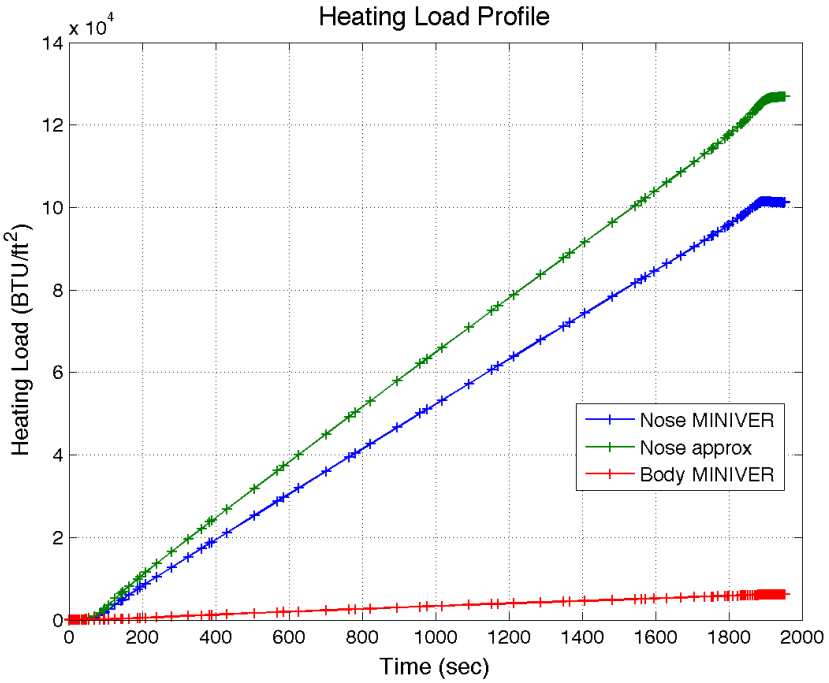


Figure 54. Optimal Trajectory: Minimum Time (Climb, Cruise, and Land), Heating Loads

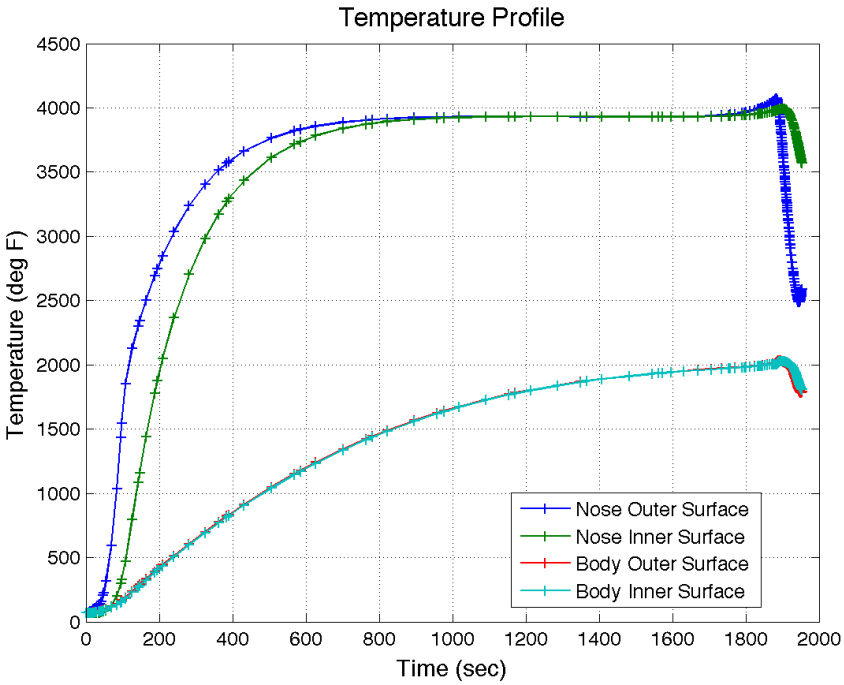


Figure 55. Optimal Trajectory: Minimum Time (Climb, Cruise, and Land), Temperatures

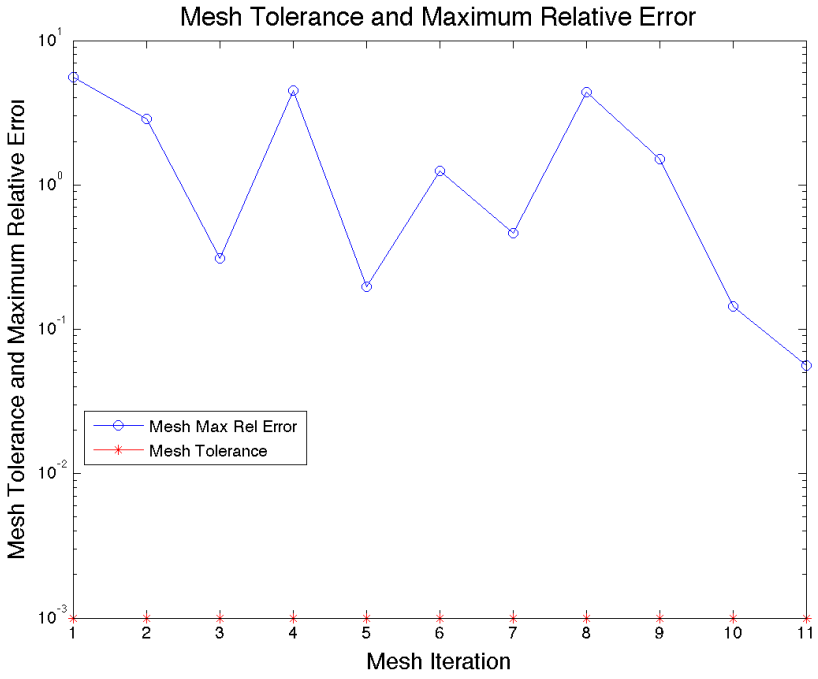


Figure 56. Optimal Trajectory: Minimum Time (Climb, Cruise, and Land), Maximum Relative Error

5.2.5 Minimum Time with Control Penalty (Climb, Cruise, and Land) Scenario.

This scenario is very similar to the previous minimum time scenario (Section 5.2.4) except that a control usage penalty is added and number of maximum mesh iterations is set to 15. Even with the increase of maximum mesh iterations, Figure 57 shows that Maximum Relative Error default (10^{-3}) is not met but the general trend is slowly leading towards meeting this value.

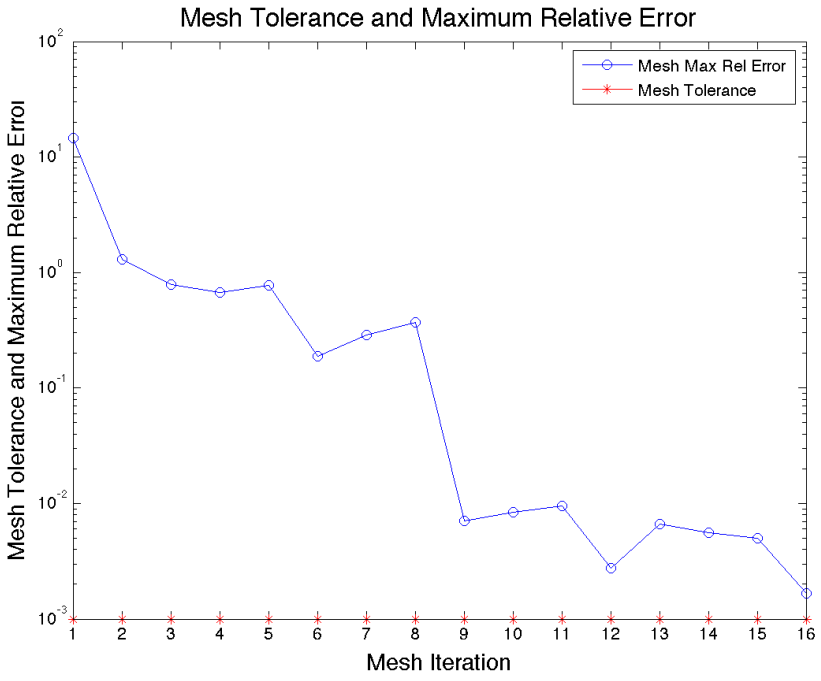


Figure 57. Optimal Trajectory: Minimum Time with Control Penalty (Climb, Cruise, and Land), Maximum Relative Error

As shown in Section 4.2.3, ϵ is the convex combination coefficient that is used to specify the contributions of both the (scaled) time and (scaled) control usage for objective function. In this scenario, the angular rate controls have an equal contribution to the control penalty and ϵ is set to 10^{-1}, favoring the control penalty over the flight time. Since for this scenario the flight time is much larger than control penalty, ϵ makes them four orders of magnitude apart.

Like the previous scenario, the vehicle takes off from a California runway, climbs and turns to its destination, then cruises until it lands on the Maine runway. Since this is a minimum time problem, Figure 58 shows that after the vehicle leaves the runway, it quickly gains altitude then cruises until it descends and lands at its destination.

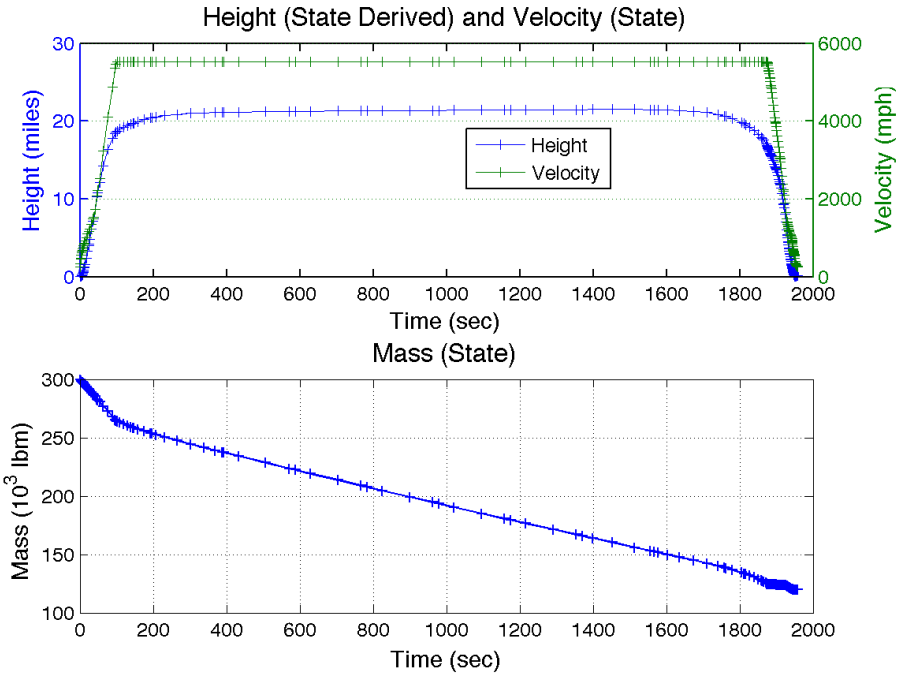


Figure 58. Optimal Trajectory: Minimum Time with Control Penalty (Climb, Cruise, and Land), States

The α and σ state profiles in Figure 59 are similar to the previous scenario but the profile during the descent phase is smoother with no 'bang-bang' controls due to the use of angular rate controls.

In Figure 60, large control actions are also seen in the ascent and descent portion of flight, as seen in previous landing scenarios, but with slightly less 'bang-bang' behavior during descent.

Compared to the same scenario without the control penalty (Figure 52), Figure 61 shows that the dynamic pressure values are similar, and are still active constraints during the descent. The g's experienced are also much lower, but still higher than

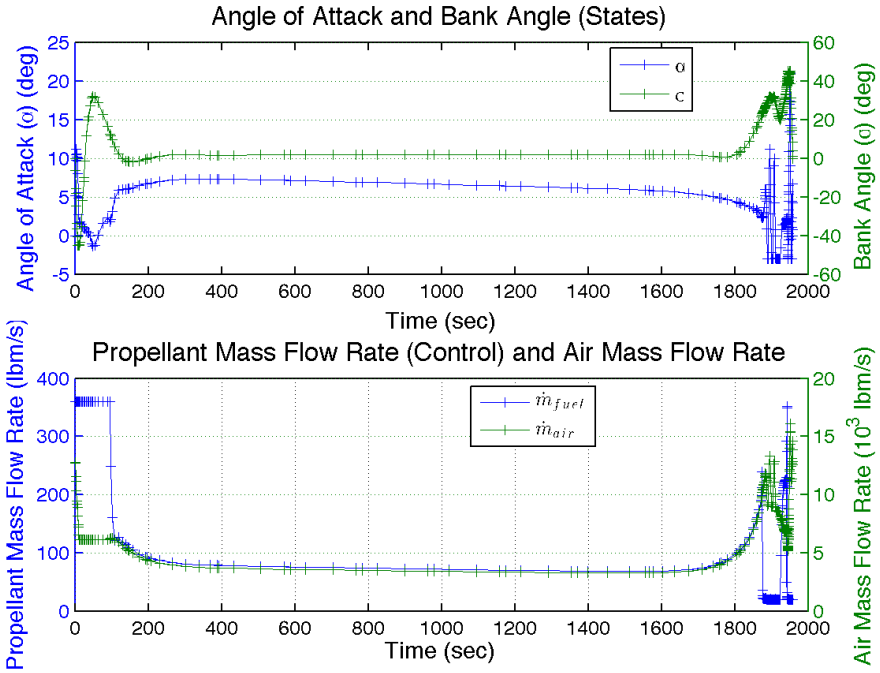


Figure 59. Optimal Trajectory: Minimum Time with Control Penalty (Climb, Cruise, and Land), Controls

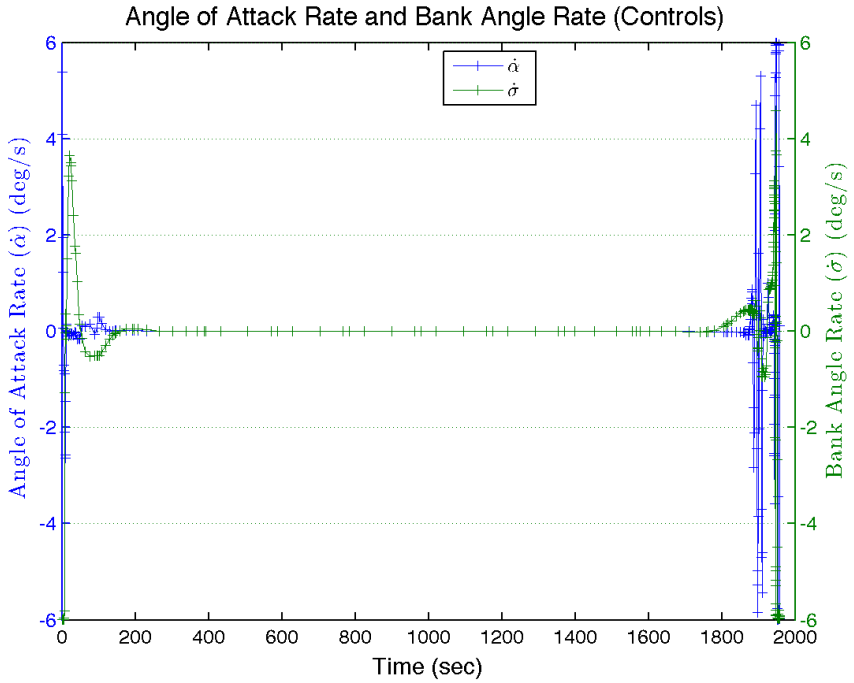


Figure 60. Optimal Trajectory: Minimum Time with Control Penalty (Climb, Cruise, and Land), Controls (angular rate)

the g's experienced in the ascent phase.

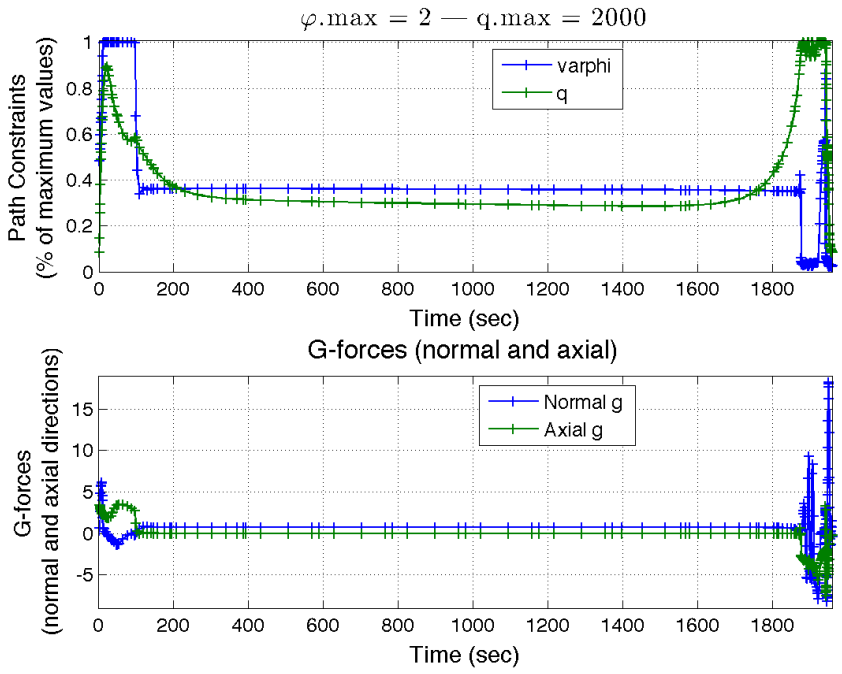


Figure 61. Optimal Trajectory: Minimum Time with Control Penalty (Climb, Cruise, and Land), Path Constraints and g's

As in Figure 53, Figure 62 has heating rates that are similar to previous scenarios until the descent portion, where the heating quickly increases for a short period until the vehicle starts slowing down from its Mach cruise number as it continues its descent into the atmosphere. Maximum MINIVER-calculated heating rate occurs at $M = 7.3$ and $h = 88000$ft. This scenario also demonstrates the previous discussed LANMIN limitation that can result in negative heating.

5.2.6 Minimum Time with Control Penalty (Climb, Cruise, Refuel, Cruise, and Land) Scenario.

This scenario is similar to the minimum time scenario in Section 5.2.5 except that a tanking event along the trajectory is added. The distance between the initial and final locations is at the limit of maximum unrefueled range. In addition, a no-fly zone

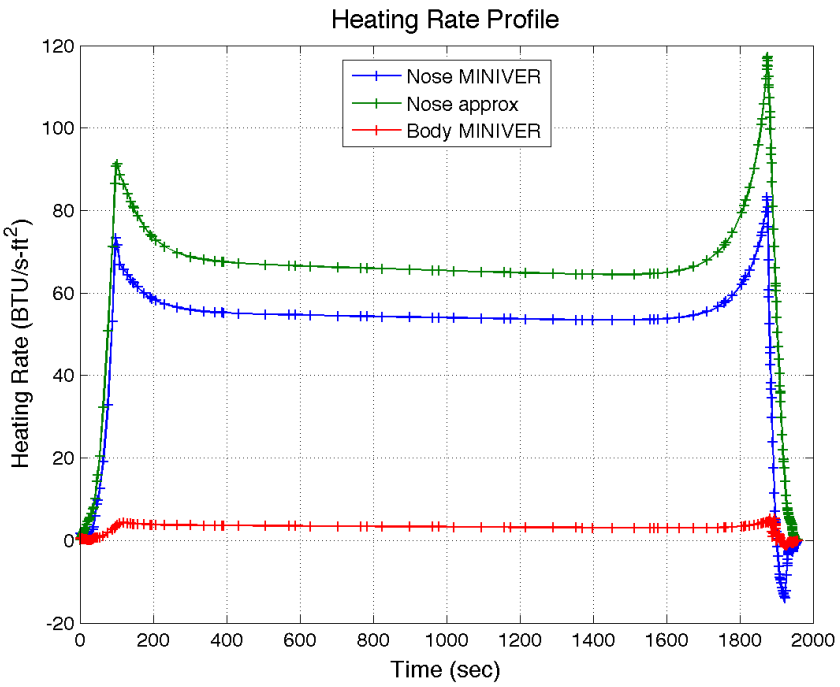


Figure 62. Optimal Trajectory: Minimum Time with Control Penalty (Climb, Cruise, and Land), Heating Rates

was added.

Similar to the previous scenario, the vehicle now takes off from a runway in California, climbs and turns to its destination, then continues until it lands on a European runway as shown in Figures 63 and 64. The refueling event is modeled as the vehicle instantaneously taking on additional fuel at $M = 0.8$ and $h = 30,000ft$. The amount of fuel added is not specified. After being refueled, the vehicle climbs to a cruising altitude until it reaches its destination. The refueling location is not predetermined; however, it is limited to occurring at least 100 (statute) miles from the initial or final locations.

In the optimal control construct, this refueling event drives the scenario to become two separate optimal control problems with state continuity conditions. This formulation is called phasing or pseudospectral knots [66]. This results in both optimal control problems being solved simultaneously while meeting the state continuity

conditions at the phasing boundary between the two problems. In this scenario, time and all states except mass are forced to be continuous at the boundary between the two phases.

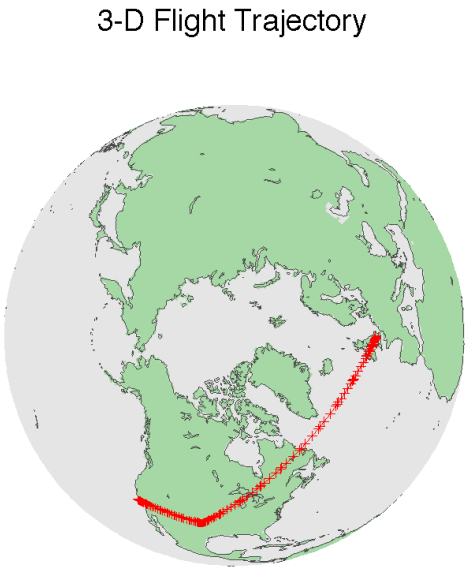


Figure 63. Optimal Trajectory: Minimum Time with Control Penalty (Climb, Cruise, Refuel, Cruise, and Land), 3-D Flight Trajectory

As seen in Figure 65, this scenario has a lower initial cruise altitude than the nominal minimum time scenario but this lower altitude is only maintained for a short duration until the vehicle descends to rendezvous with a tanker. Like in the previous control penalty scenarios, use of an appropriate ϵ (10^{-2}) results in a optimal trajectory without any scalloping.

Figure 66 shows the Mach number profile for this scenario. While the cruise velocities in both phases are at the limit for that state, the Mach number during the first phase is higher than the second phase. Since the second phase cruise altitude is higher than the first phase, the speed of sound in the second phase is greater than the first phase. Thus, since $M = V/a$, the Mach number in the first phase is higher

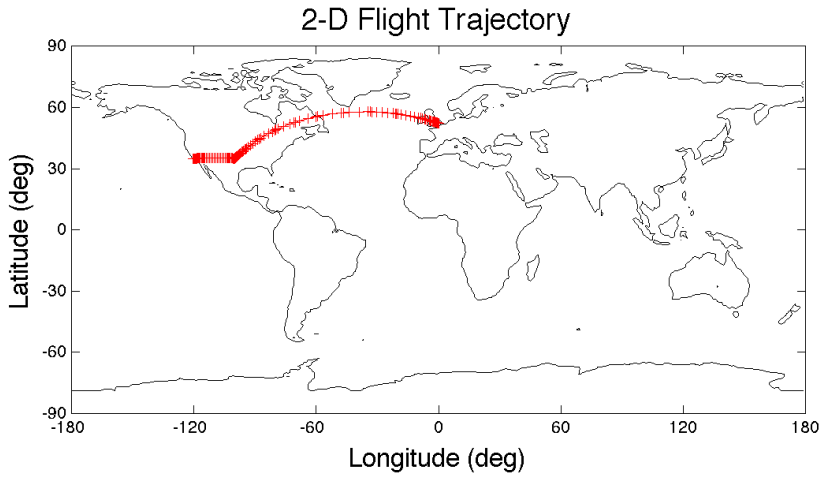


Figure 64. Optimal Trajectory: Minimum Time with Control Penalty (Climb, Cruise, Refuel, Cruise, and Land), 2-D Flight Trajectory

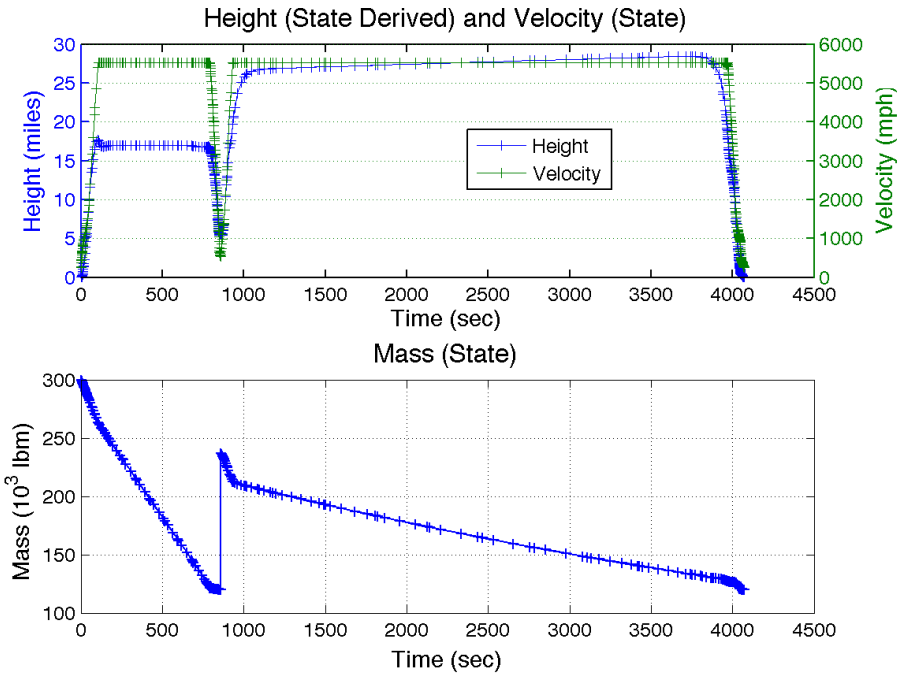


Figure 65. Optimal Trajectory: Minimum Time with Control Penalty (Climb, Cruise, Refuel, Cruise, and Land), States

than the second phase.

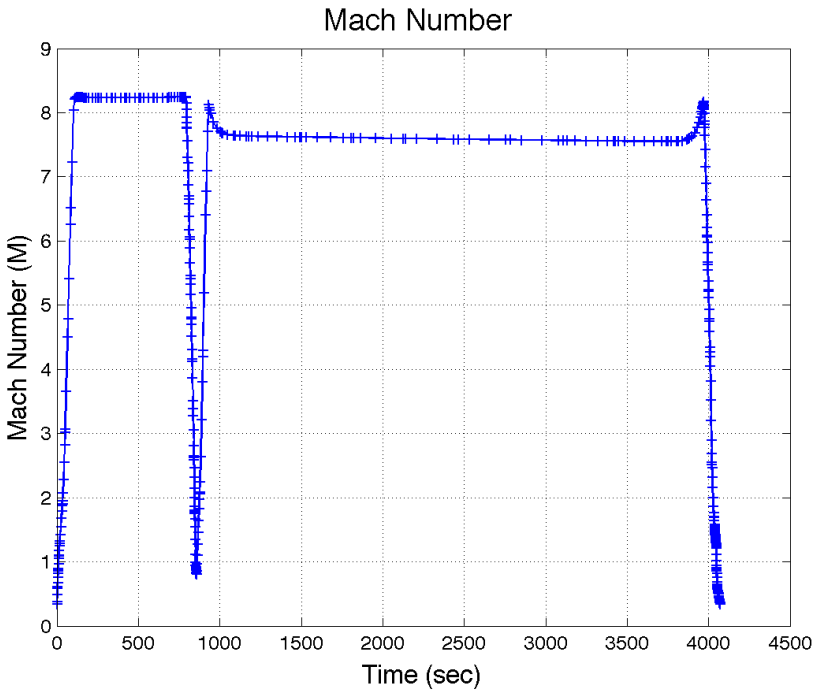


Figure 66. Optimal Trajectory: Minimum Time with Control Penalty (Climb, Cruise, Refuel, Cruise, and Land), Mach Number

Given the minimum time formulation, the vehicle quickly tries to descend to rendezvous with the tanker. Since the GHAME lookup tables have a limited range for both α and φ, the vehicle does not have an effective means of shedding energy from its initial cruise and velocity (unlike the Space Shuttle that flew several high-α S-curves). Thus, the vehicle undergoes several "bang-bang" maneuvers to shed energy to rendezvous with the tanker at the lower altitude/speed as seen in Figures 67 and 68.

As seen in Figure 62, Figure 69 has heating rates that are similar to previous scenarios until the descent portion for both the refueling event as well as landing, where the heating quickly increases for a short period until the vehicle starts slowing down from its Mach cruise number as it continues its descent into the atmosphere. In the first phase of flight, since the vehicle is flying much lower in the atmosphere, the heating rates in this phase are much lower than the post-refueling phase, even

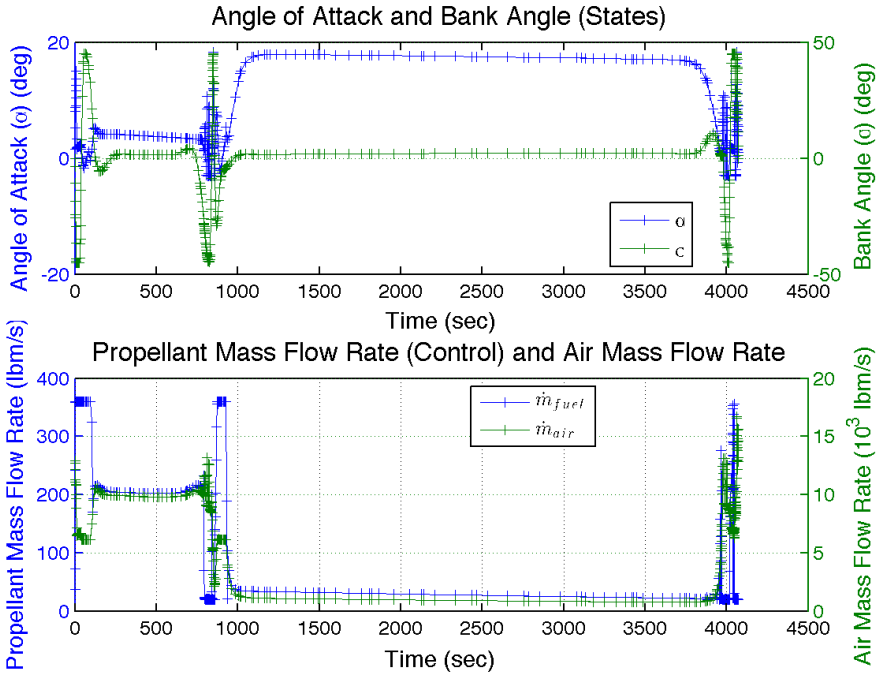


Figure 67. Optimal Trajectory: Minimum Time with Control Penalty (Climb, Cruise, Refuel, Cruise, and Land), Controls

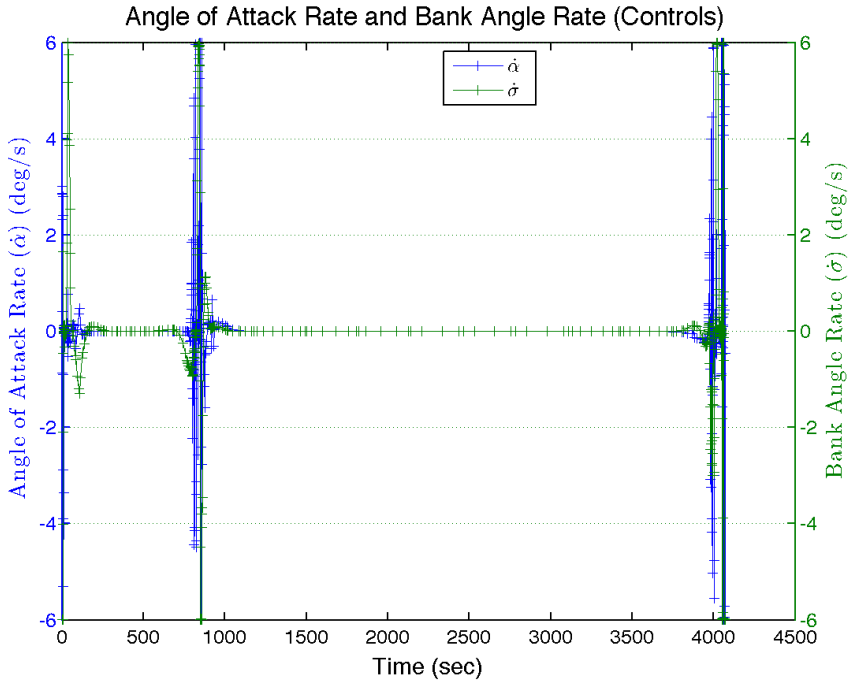


Figure 68. Optimal Trajectory: Minimum Time with Control Penalty (Climb, Cruise, Refuel, Cruise, and Land), Controls (angular rate)

though the velocities in both phases are comparable. This scenario also demonstrates the previous discussed LANMIN limitation that can result in negative heating.

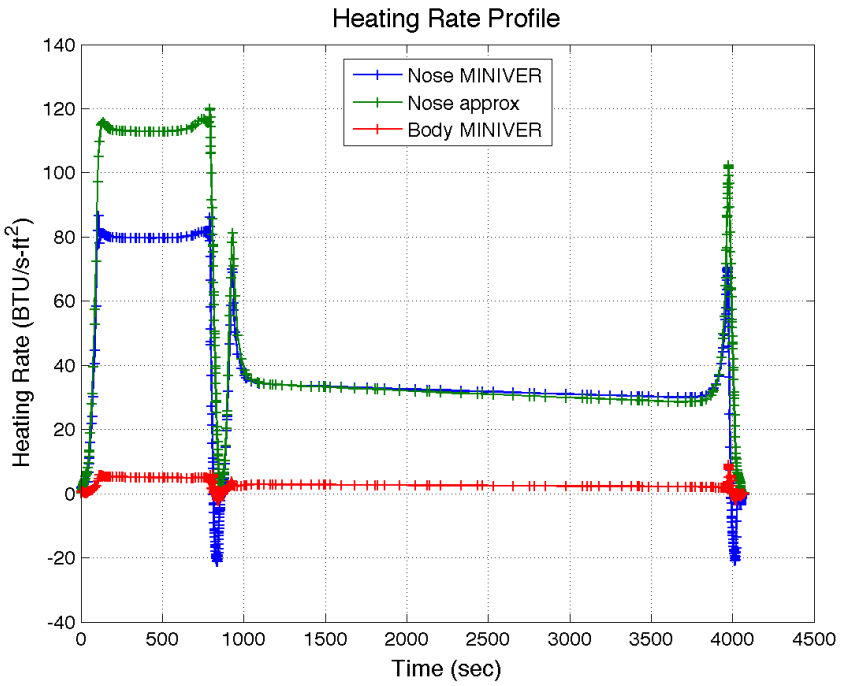


Figure 69. Optimal Trajectory: Minimum Time with Control Penalty (Climb, Cruise, Refuel, Cruise, and Land), Heating Rates

The temperature profile in Figure 70 shows a similar shape in the previous landing scenario, but in this scenario there are two parts to the temperature profile: first phase of takeoff to refueling and the second phase of refueling to landing. In the first phase, the nose temperatures do not get as high as the previous scenario since the vehicle has a short cruise time before descending to refuel. There is a small spike in temperature as the vehicle begins its descent and then the vehicle quickly cools down in the lower atmosphere at a much lower velocity. Prior to refueling, there would have to be some additional temperature reduction near the vehicle's refueling adapter to allow for safe refueling. After the refueling, the vehicle climbs to a cruise altitude which is higher than the previous cruise altitude and flies a little slower, resulting in a lower cruise temperature.

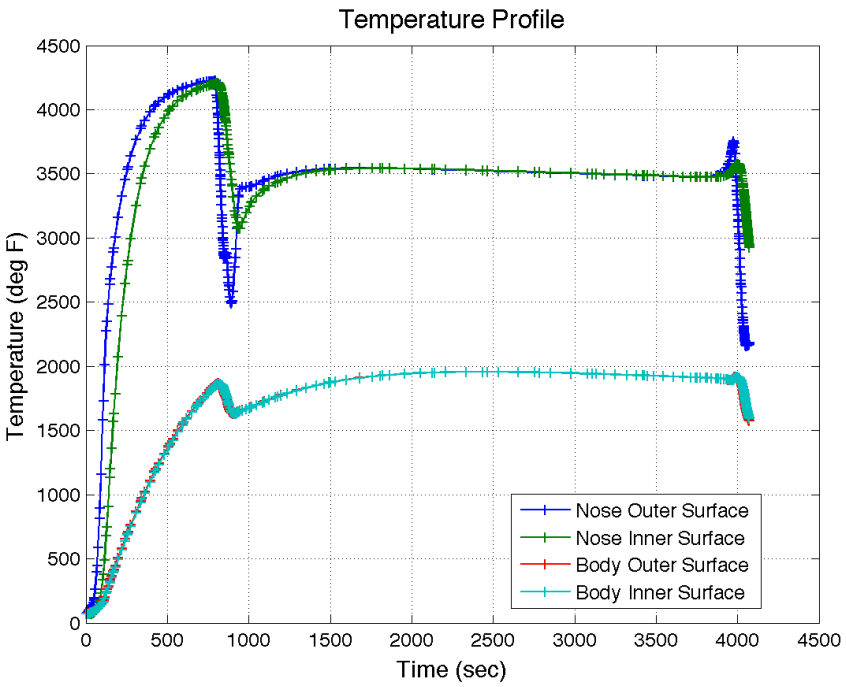


Figure 70. Optimal Trajectory: Minimum Time with Control Penalty (Climb, Cruise, Refuel, Cruise, and Land), Temperatures

Figure 71 shows the Mesh Interval History. An interesting aspect of this figure is the transition between phases occurs at varying normalized times. This is a result of mesh solutions with different final times, driving different phase normalized times. This scenario completed with an optimal solution but the Maximum Relative Error did not meet the specified tolerance.

5.2.7 Minimum Time with Control Penalty, No-Fly Zone and g-limits (Climb, Cruise, Refuel, Cruise, and Land) Scenario.

This scenario is similar to Section 5.2.6 but it includes a no-fly zone (described in Section 1.3), which is a 100 (statute) mile radius hemisphere centered on the earth's surface on a point that would have been near the vehicle's ground track if there was not a no-fly zone. This zone is shown in Figure 72 in the south-central Unites States. This scenario also includes limits on g's

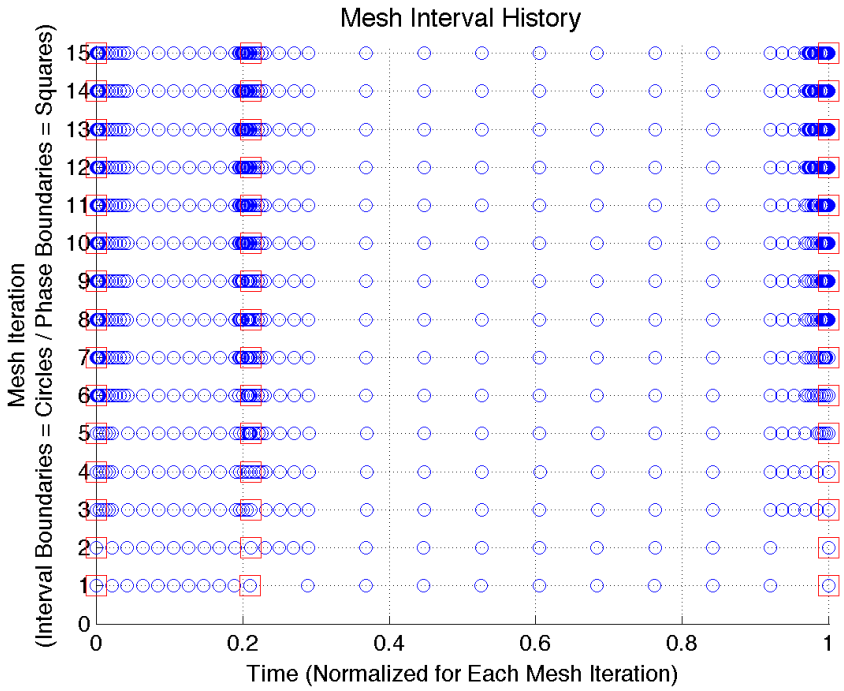


Figure 71. Optimal Trajectory: Minimum Time with Control Penalty (Climb, Cruise, Refuel, Cruise, and Land), Mesh Interval History

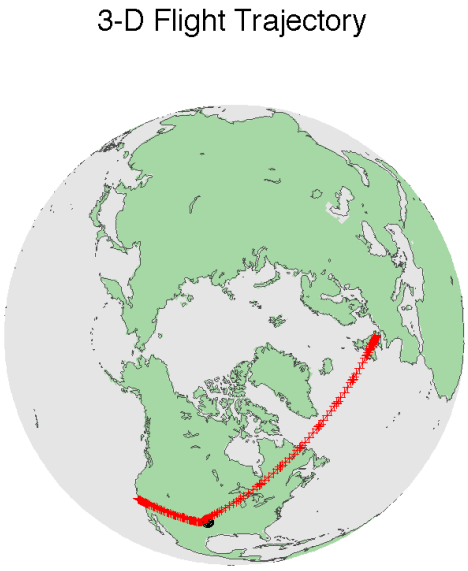


Figure 72. Optimal Trajectory: Minimum Time with Control Penalty, No-Fly Zone and g-limits (Climb, Cruise, Refuel, Cruise, and Land), 3-D Flight Trajectory

Figure 73 shows the vehicle distance from the no-fly zone. To meet the path constraint the distance can be no less than 100 miles, which is represented by the red horizontal line.

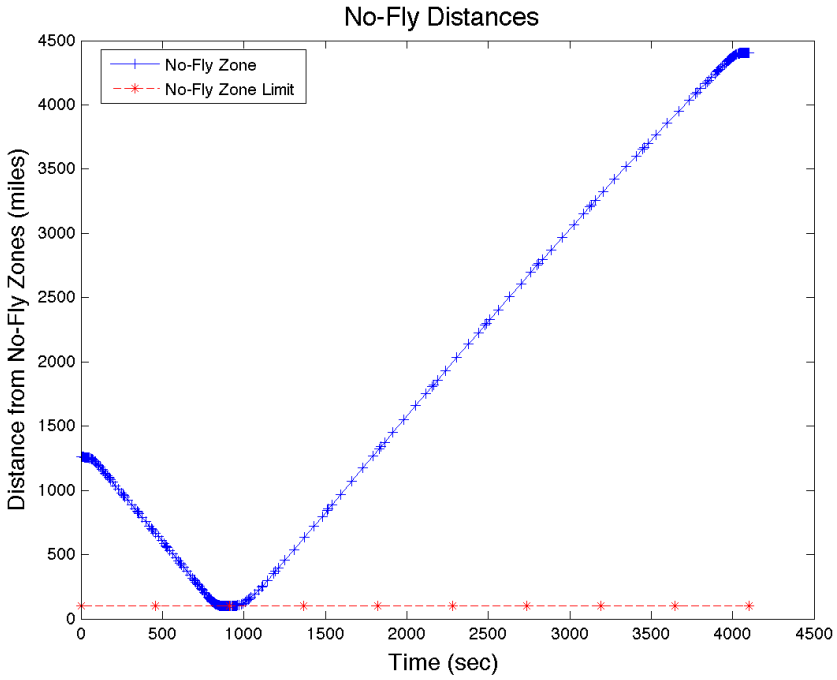


Figure 73. Optimal Trajectory: Minimum Time with Control Penalty, No-Fly Zone and g-limits (Climb, Cruise, Refuel, Cruise, and Land), No-Fly Distance

Figure 74 differs from the previous scenario (Figure 67) in that to avoid the no-fly zone, the vehicle uses a larger control action earlier in the flight. Otherwise the profiles look similar.

5.2.8 Minimum Time with Control Penalty, No-Fly Zone and Sensor Constraints (Climb, Cruise, Sense, and Cruise) Scenario.

This scenario is similar to the scenario in Section 5.2.7 but instead of having a refueling event and g's path constraint, it instead has sensor collection activity. Including a sensor activity drives several scenario attributes. Since the sensor activity is going to occur over 30 seconds when near the target, this part of the trajectory has

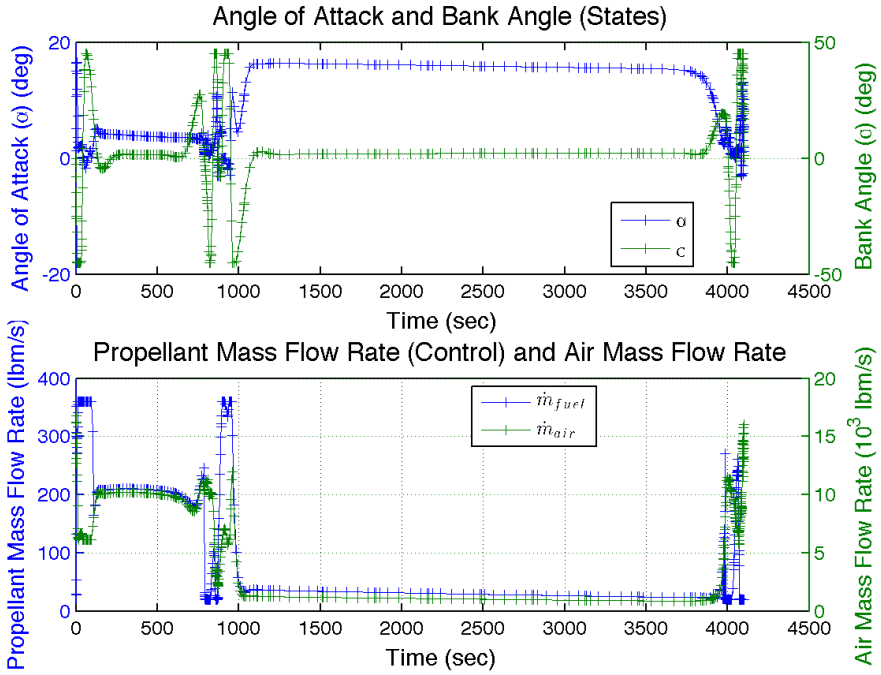


Figure 74. Optimal Trajectory: Minimum Time with Control Penalty, No-Fly Zone and g-limits (Climb, Cruise, Refuel, Cruise, and Land), Controls

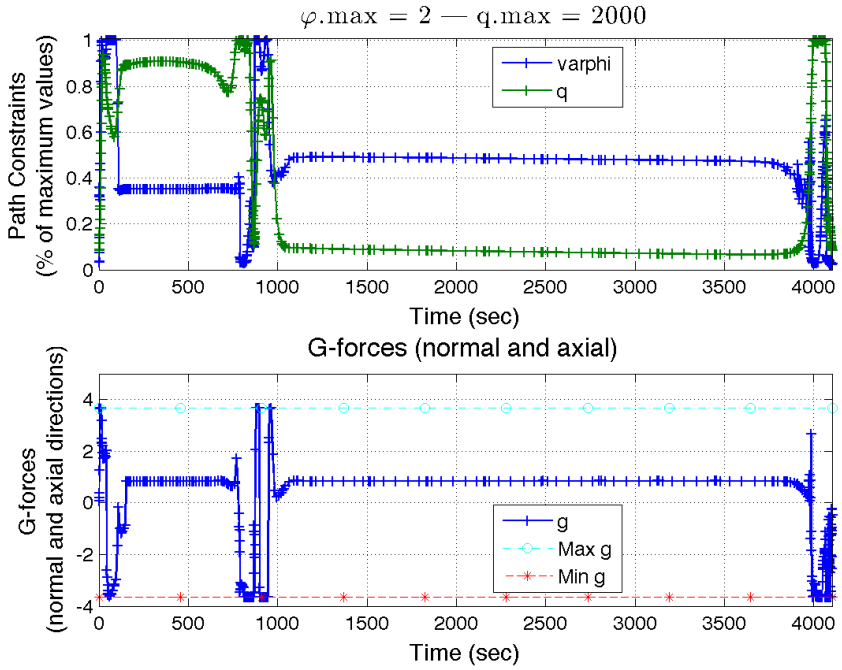


Figure 75. Optimal Trajectory: Minimum Time with Control Penalty, No-Fly Zone and g-limits (Climb, Cruise, Refuel, Cruise, and Land), Path Constraints and g's

a dedicated phase. Including the phases before and after the sensor activity, the total number of phases is 3. The sensor collection drives a minimum (10°) Line-of-Sight (LOS) (elevation above the target's horizon) angle from the target to the vehicle, such that the vehicle will have clear LOS to the target during the sensor collection activity. Finally, the vehicle has to be within 300 (statute) miles of the target during collection activity. For this scenario the no-fly zone is a 100 (statute) mile radius hemisphere one latitude degree north of the sensor target.

Figures 76 and 77 show the keep out zone in relation to the earth and also to the calculated optimal trajectory.

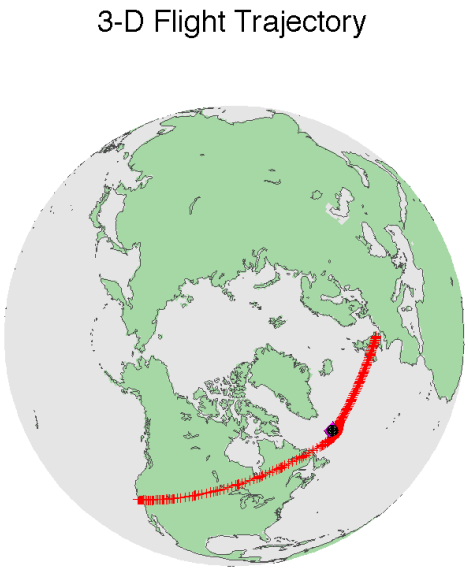


Figure 76. Optimal Trajectory: Minimum Time with Control Penalty, No-Fly Zone and Sensor Constraints (Climb, Cruise, Sense, and Cruise), 3-D Flight Trajectory

Figure 78 shows the sensor target (green diamond) on a Mercator projection and also to the calculated optimal trajectory.

Figure 79 shows the resulting distance from the no-fly zone to each trajectory point. To successfully meet this path constraint, the vehicle needs to be above the

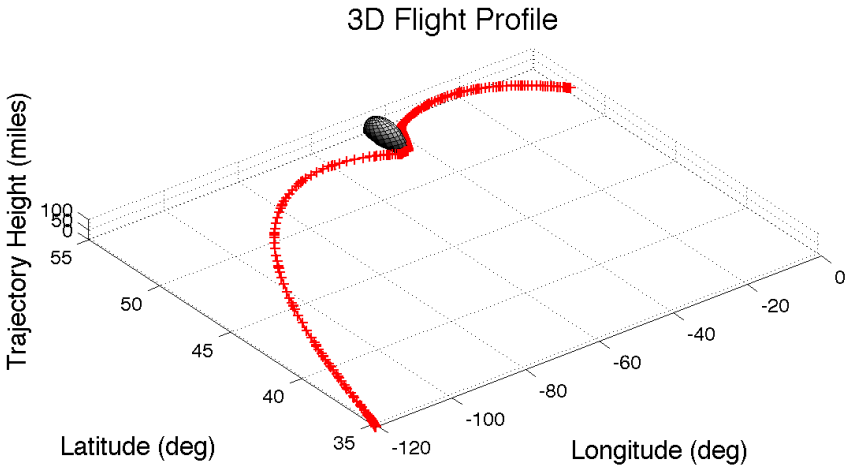


Figure 77. Optimal Trajectory: Minimum Time with Control Penalty, No-Fly Zone and Sensor Constraints (Climb, Cruise, Sense, and Cruise), 3-D Flight Profile

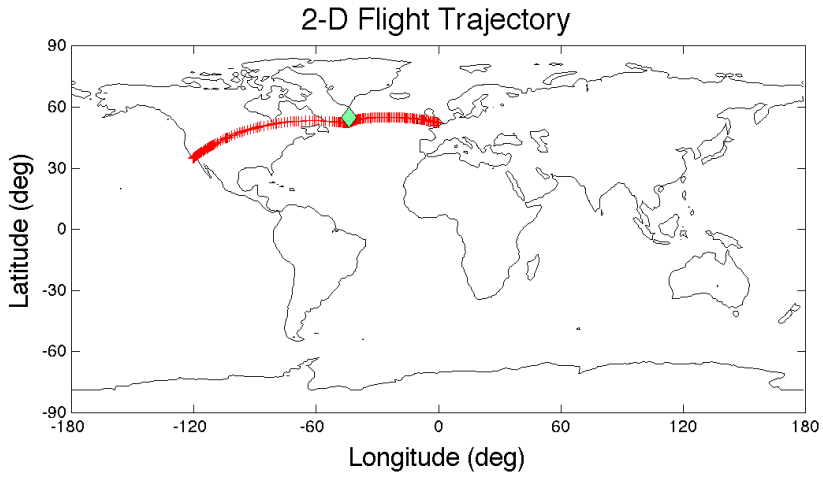


Figure 78. Optimal Trajectory: Minimum Time with Control Penalty, No-Fly Zone and Sensor Constraints (Climb, Cruise, Sense, and Cruise), 2-D Flight Trajectory

red line, at least 100 miles away from the no-fly zone center.

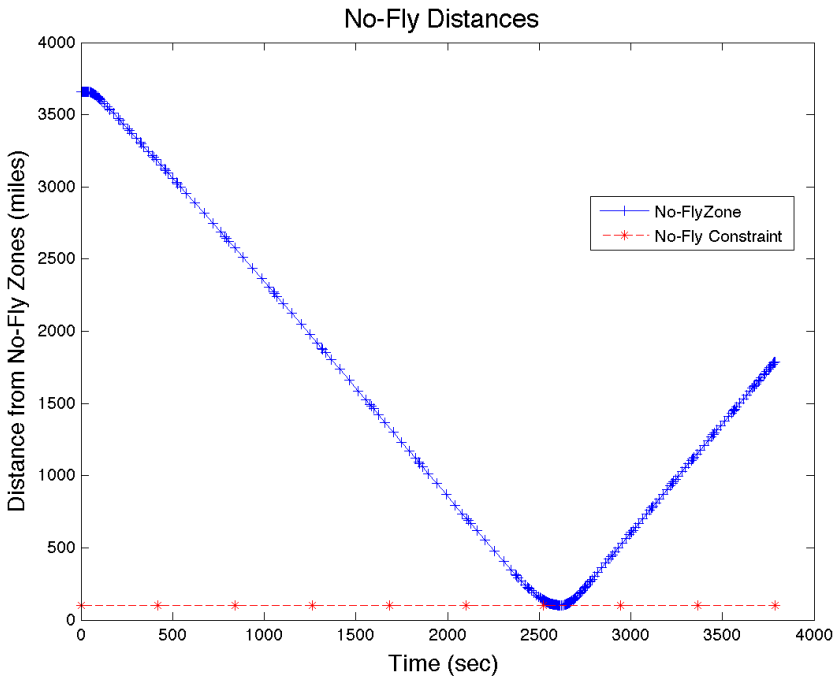


Figure 79. Optimal Trajectory: Minimum Time with Control Penalty, No-Fly Zone and Sensor Constraints (Climb, Cruise, Sense, and Cruise), No-Fly Distance

Figure 80 shows the resulting distance from the sensor target to each trajectory point, with the sensing activity window shown in Figure 81. To successfully meet this path constraint, the vehicle needs to be below the red line, less than 300 miles away from the sensor target for 30 seconds. This figure also shows the elevation angle from the target to the vehicle. To successfully meet this path constraint, the vehicle needs to be above the red line, more than 10 degrees elevation angle for 30 seconds.

Figures 82, 83, 84, and 85 show the vehicle maneuvers to avoid the no-fly zone and conduct a successful sensor positioning relative to the target. When approaching the no-fly zone, the vehicle climbs and turns south to avoid the no-fly zone, then climbs to conduct the sensor activity and finally resumes its heading to its destination as seen in Figure 86.

For the scenario in this section, the resulting azimuth and elevation angles (during

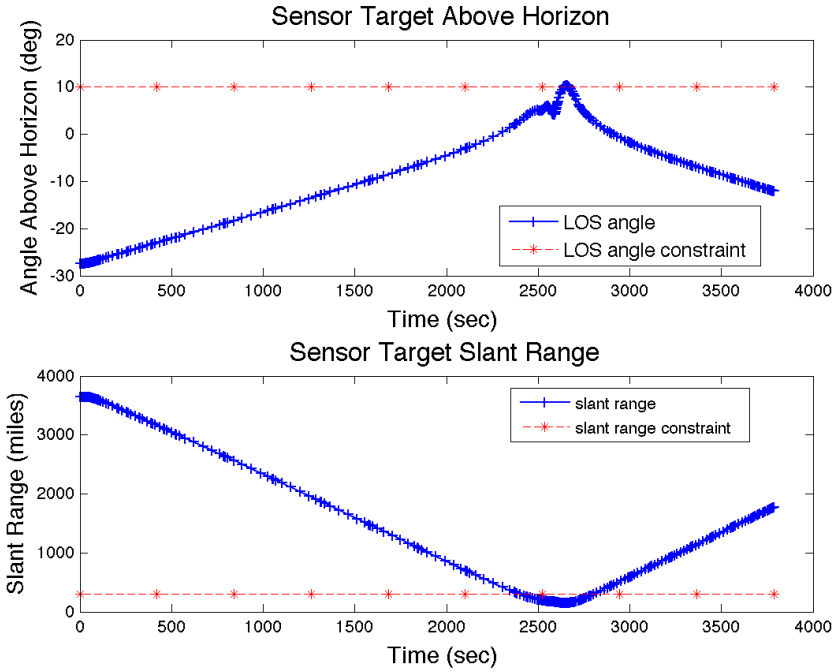


Figure 80. Optimal Trajectory: Minimum Time with Control Penalty, No-Fly Zone and Sensor Constraints (Climb, Cruise, Sense, and Cruise), Sensor Constraints

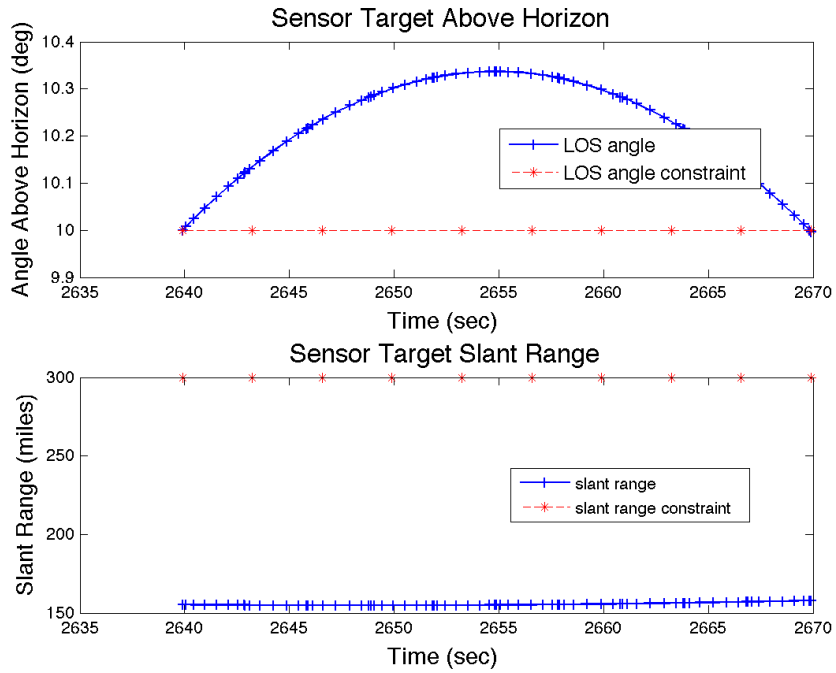


Figure 81. Optimal Trajectory: Minimum Time with Control Penalty, No-Fly Zone and Sensor Constraints (Climb, Cruise, Sense, and Cruise), Sensor Constraints (sensor activity)

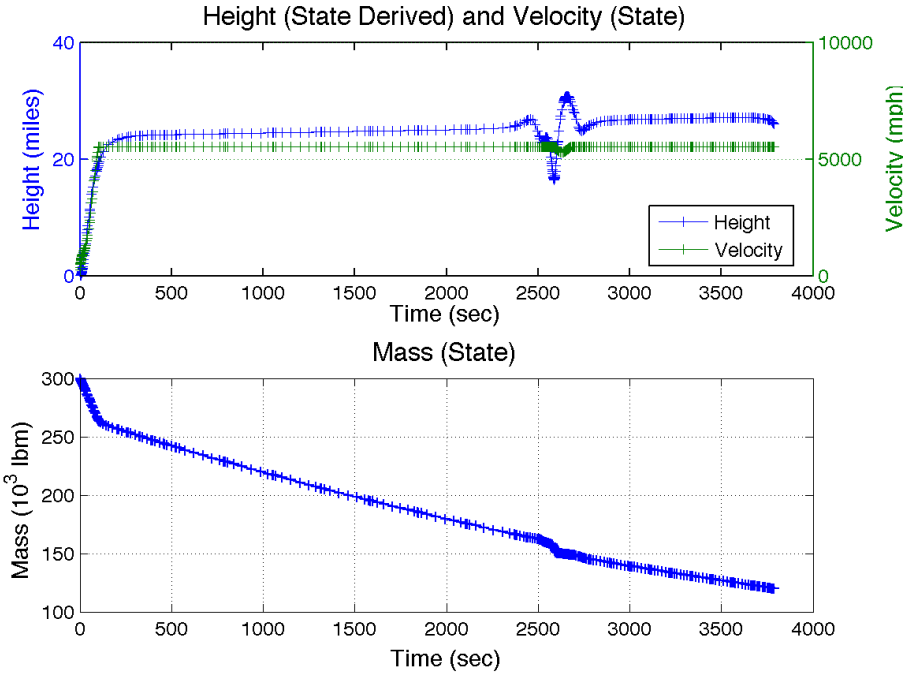


Figure 82. Optimal Trajectory: Minimum Time with Control Penalty, No-Fly Zone and Sensor Constraints (Climb, Cruise, Sense, and Cruise), States

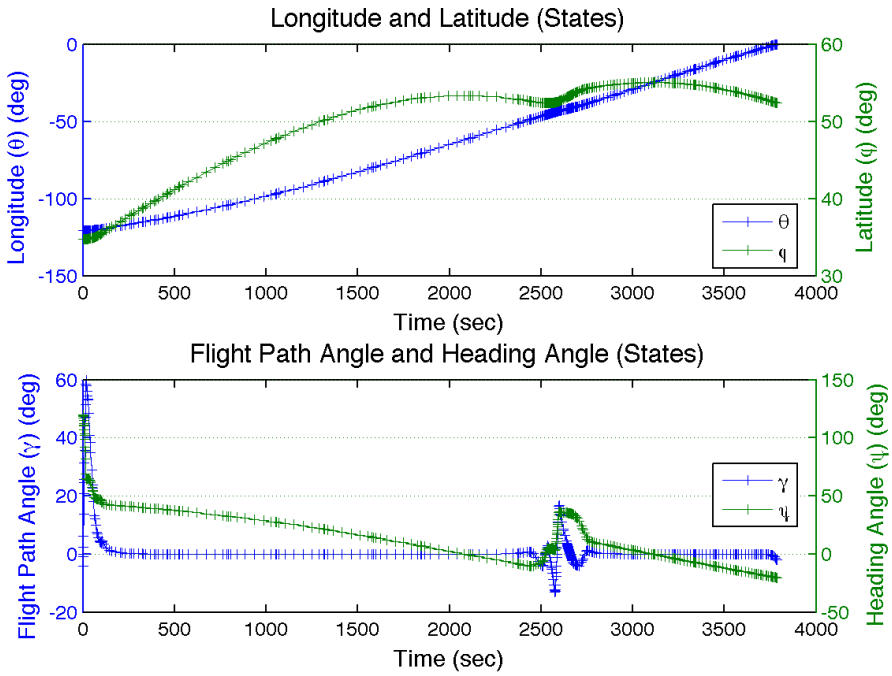


Figure 83. Optimal Trajectory: Minimum Time with Control Penalty, No-Fly Zone and Sensor Constraints (Climb, Cruise, Sense, and Cruise), States (angular)

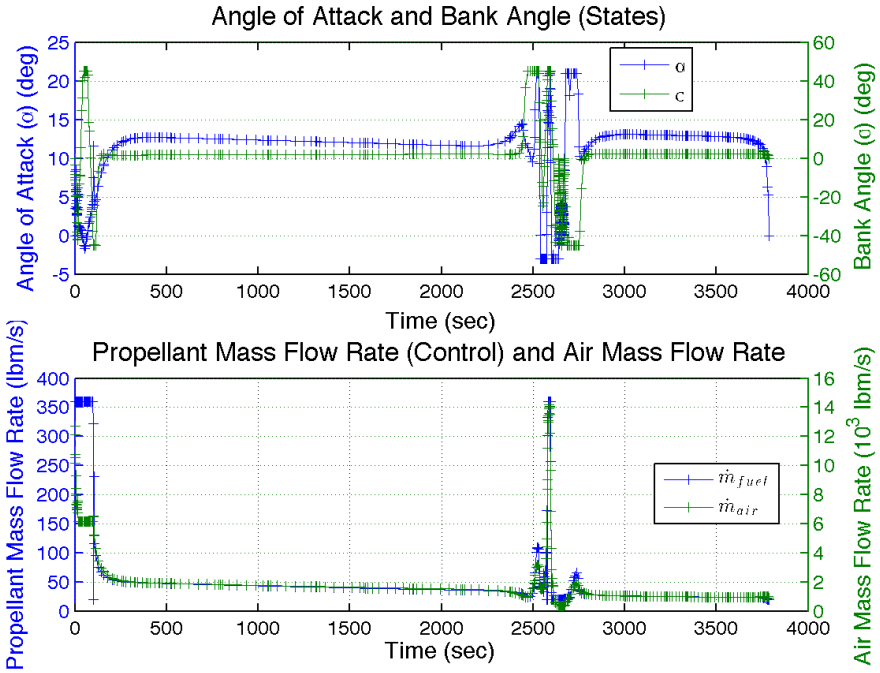


Figure 84. Optimal Trajectory: Minimum Time with Control Penalty, No-Fly Zone and Sensor Constraints (Climb, Cruise, Sense, and Cruise), Controls

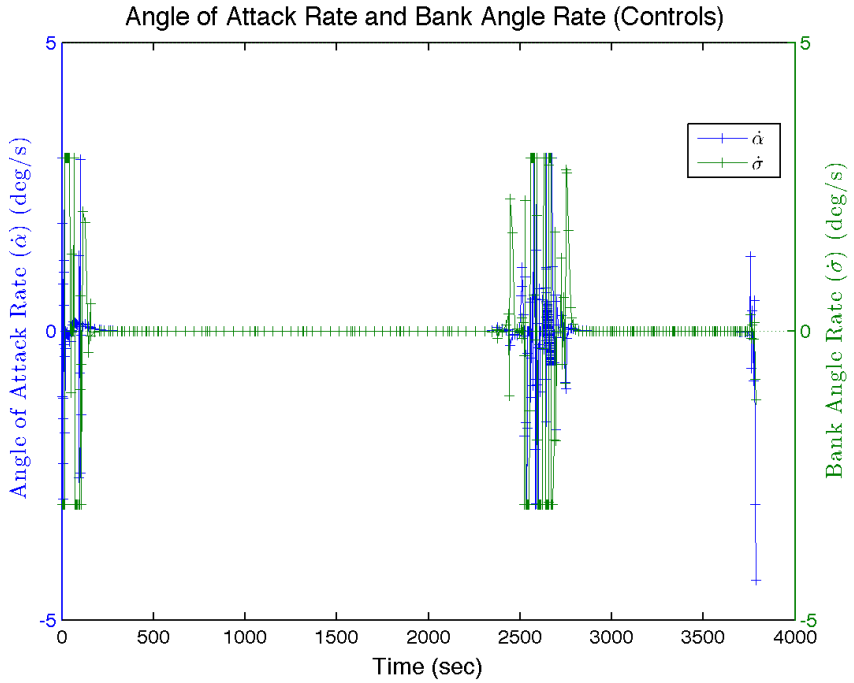


Figure 85. Optimal Trajectory: Minimum Time with Control Penalty, No-Fly Zone and Sensor Constraints (Climb, Cruise, Sense, and Cruise), Controls (angular rate)

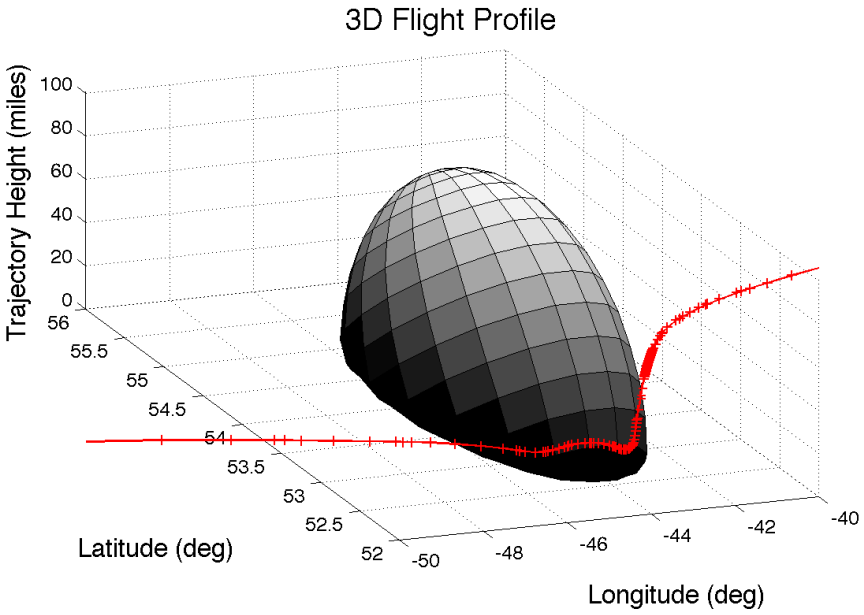


Figure 86. Optimal Trajectory: Minimum Time with Control Penalty, No-Fly Zone and Sensor Constraints (Climb, Cruise, Sense, and Cruise), 3-D Flight Profile

the sensing phase) are shown in Figure 87. The resulting behavior of both look angles are a result of vehicle maneuvers occurring during the sensing phase.

Figure 88 shows the relationship between bank angle and sensor elevation angle during the sensing phase. As seen in Figure 87, the high positive elevation angles are not intuitive. Since the vehicle is maneuvering to avoid the no-fly zone (Figure 77) while still meeting the LOS (elevation) angle from the target to the vehicle (Figure 80), it is climbing and turning during the sensing phase. These maneuvers result in high positive sensor elevation angles.

While Equations 43 and 44 can be used to calculate azimuth and elevation angles for an arbitrary trajectory, they cannot be used directly as optimal control path constraints due to the possibility of solutions to $\tan^{-1}$ ($atan2$) jumping between the extremes of its range, due to the periodic nature of trigonometric functions. As seen in Figure 89, there can be a "wrap around" in the computed sensor azimuth angle,

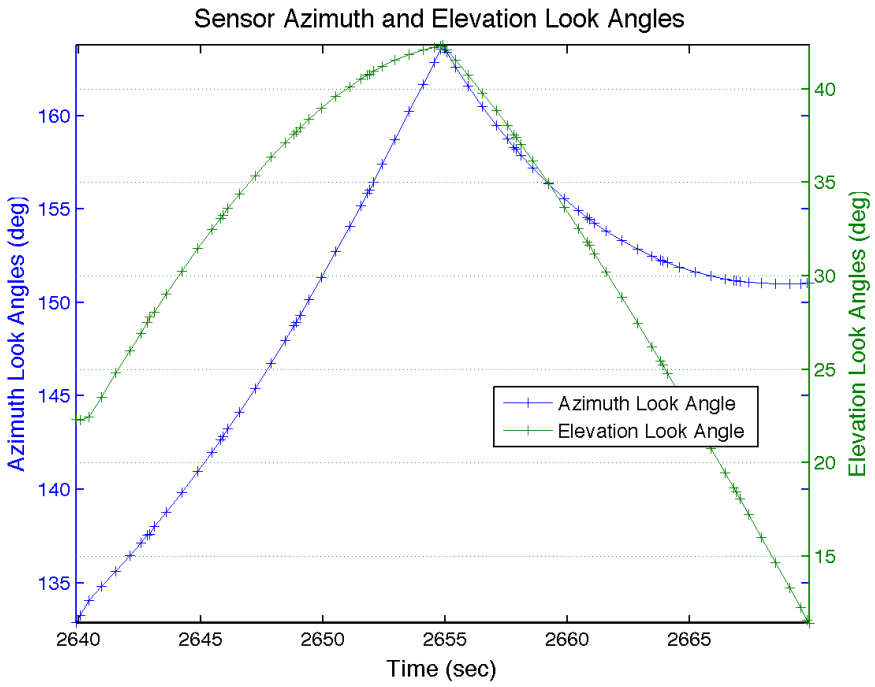


Figure 87. Optimal Trajectory: Minimum Time with Control Penalty, No-Fly Zone and Sensor Constraints (Climb, Cruise, Sense, and Cruise), Sensor Look Angles During Sensing Phase

which would preclude use of this angle as a path constraint in the optimal control formulation used in this research. While this behavior precludes use of these angles in an optimal control formulation, Figure 89 shows this behavior for these angles computed after an optimal solution was determined.

5.2.9 Minimum Time with Temperature Constraints (Climb and Cruise) Scenario.

This scenario is similar to Section 5.2.2 but introduces a temperature-based path constraint using MINIVER temperature calculations. Only the nose temperature constraint is used here since the maximum nose temperature seen in the same scenario without temperature constraints (3955 °F) is at the nose material's (tantalum) maximum allowable temperature (4000 °F) [88] while the maximum body temperature seen in the same scenario without temperature constraints (1999 °F) is below

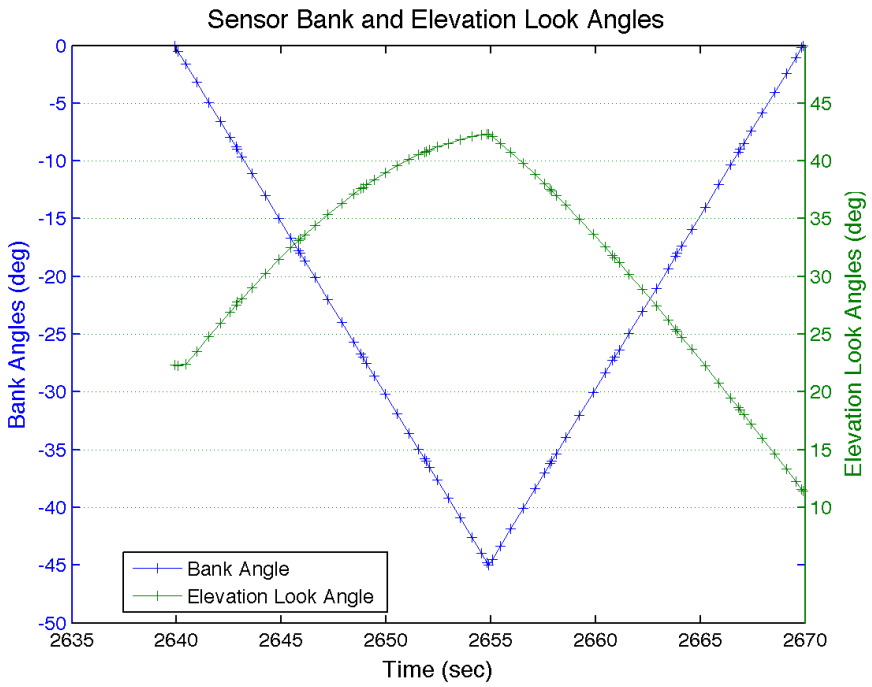


Figure 88. Optimal Trajectory: Minimum Time with Control Penalty, No-Fly Zone and Sensor Constraints (Climb, Cruise, Sense, and Cruise), Bank and Elevation Angles During Sensing Phase

the body material's (molybdenum) maximum allowable temperature (2200 °F) [88]. For this scenario, the nose temperature constraint is set at an arbitrary 3600 °F, 400 degrees (10%) lower than the temperatures experienced in the same minimum time scenario (Section 5.2.2) without temperature constraints.

In order to find a numerical solution as rapidly as possible, SNOPT was used in this scenario. For this research, IPOPT is generally more robust (more likely to compute an optimal solution for difficult formulations) but slower than SNOPT since SNOPT is more efficient solving problems with many linear constraints while IPOPT is more efficient solving problems with many nonlinear constraints. Thus for this scenario some GPOPS settings from Table 3 are different: 'SNOPT' solver, 'first' derivative level, 'RPMDifferentiation' setup method, '10^{-4}' SNOPT tolerance, and '2000' maximum SNOPT iterations.

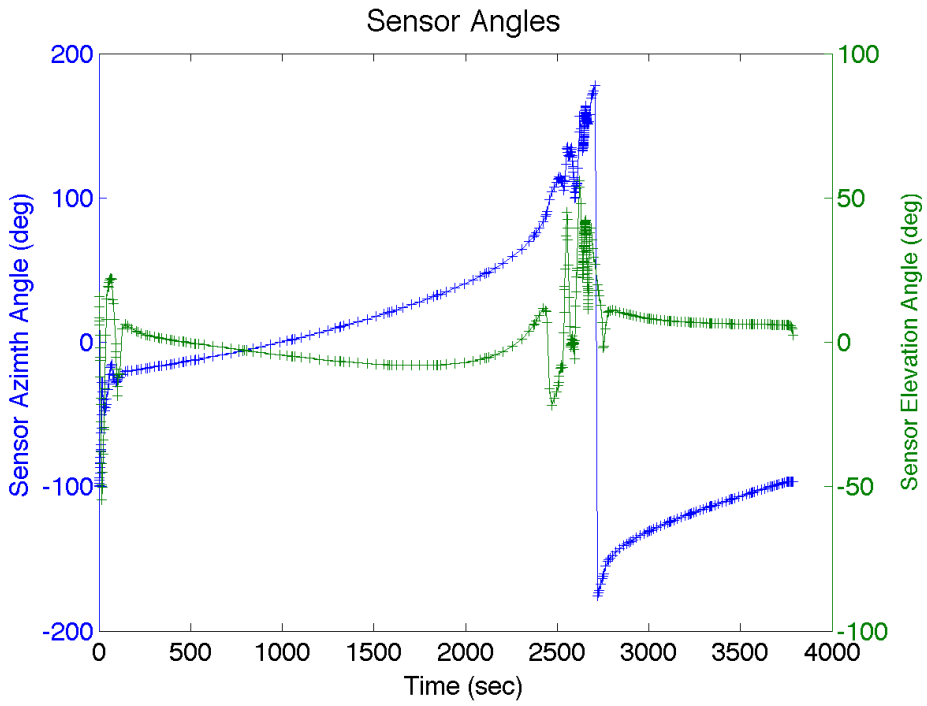


Figure 89. Optimal Trajectory: Minimum Time with Control Penalty, No-Fly Zone and Sensor Constraints (Climb, Cruise, Sense, and Cruise), Sensor Look Angles During Entire Trajectory

As previously discussed in Section 3.5, MINIVER is executing its code only using single-precision values. Since this is less than the MATLAB-default double precision, it can introduce limitations in the solution precision, as well as instabilities in the computations since the noise in using single-precision data with double-precision data may cause the NLP solver to diverge. To reduce these possibilities, as mentioned in the previous paragraph, the NLP solver tolerance was reduced to 10^{-4}. Using that value and the default mesh tolerance value, an objective solution was found meeting both the NLP and mesh tolerance.

Figure 90 shows the effect of introducing the temperature constraint. At a point along the trajectory, the maximum Mach number (velocity) is decreased to reduce heating, as compared to the constant cruise velocity seen in Figure 34. As seen in Figure 93, that point is when the temperature reaches the path constraint.

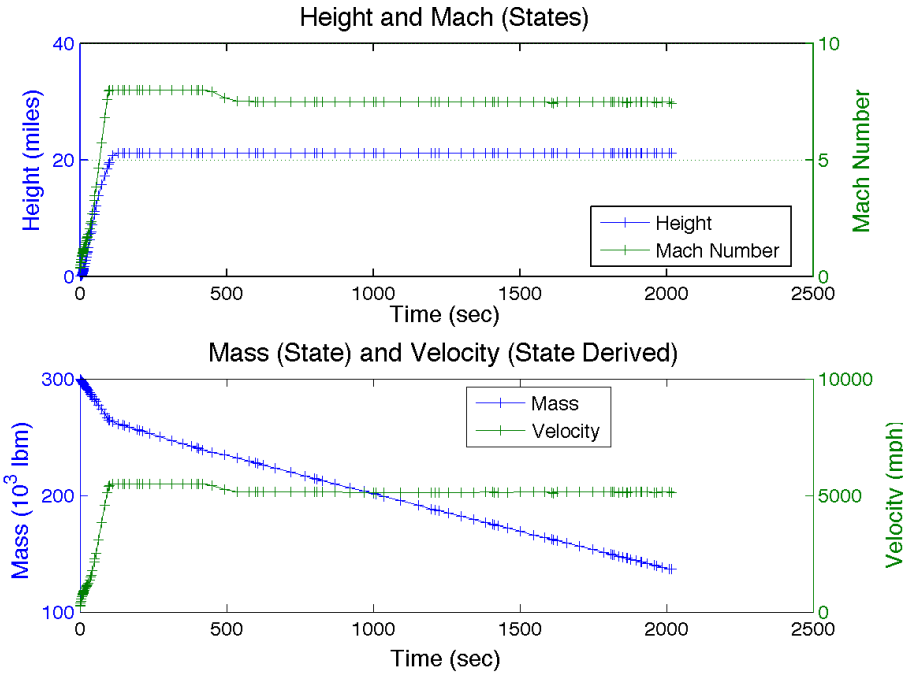


Figure 90. Optimal Trajectory: Minimum Time with Temperature Constraints (Climb and Cruise), States

In Figure 91, it is obvious where the reduction in velocity occurs, since the nose heating rate reduces as well. A corresponding change in Figure 92 shows that the heating load curve slope reduces slightly at the same point where the heating rate decreases.

Figure 93 shows the impact of the temperature constraint. When the nose temperature curve reaches the path constraint, the vehicle slows down and the nose temperature remains constant until the end of the scenario.

5.2.10 Minimum Time with Control Penalty and Temperature Constraints (Climb, Cruise, Refuel, Cruise, and Land) Scenario.

This scenario is similar to Section 5.2.6 but adds a temperature-based path constraint using MINIVER temperature calculations. As in the other scenario with a temperature-based path constraint (Section 5.2.9), only the nose temperature path

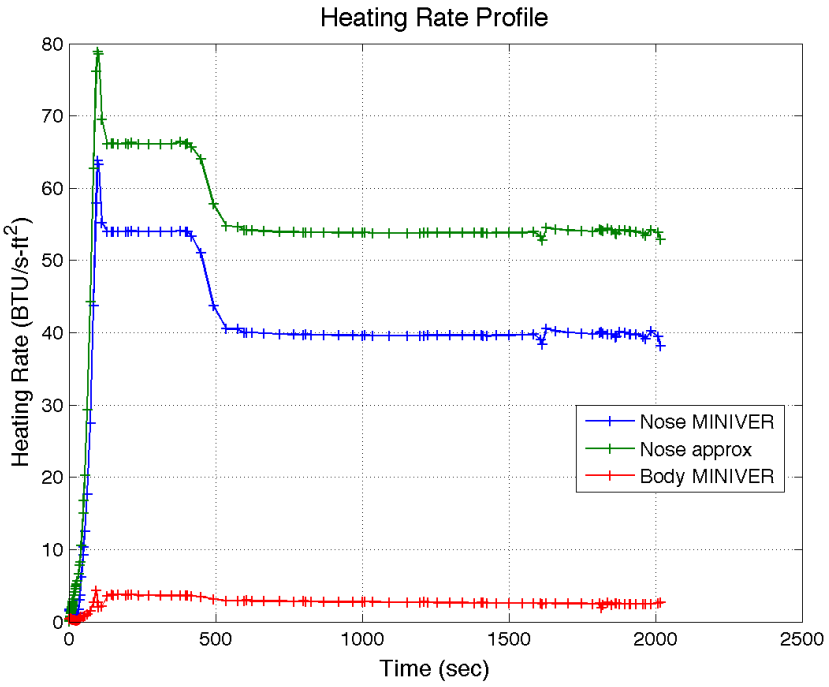


Figure 91. Optimal Trajectory: Minimum Time with Temperature Constraints (Climb and Cruise), Heating Rates

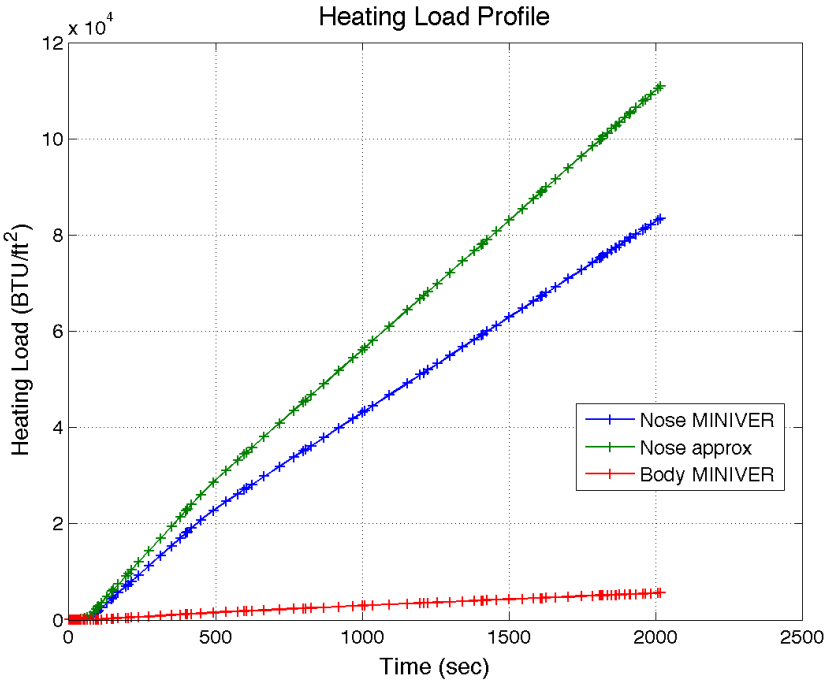


Figure 92. Optimal Trajectory: Minimum Time with Temperature Constraints (Climb and Cruise), Heating Loads

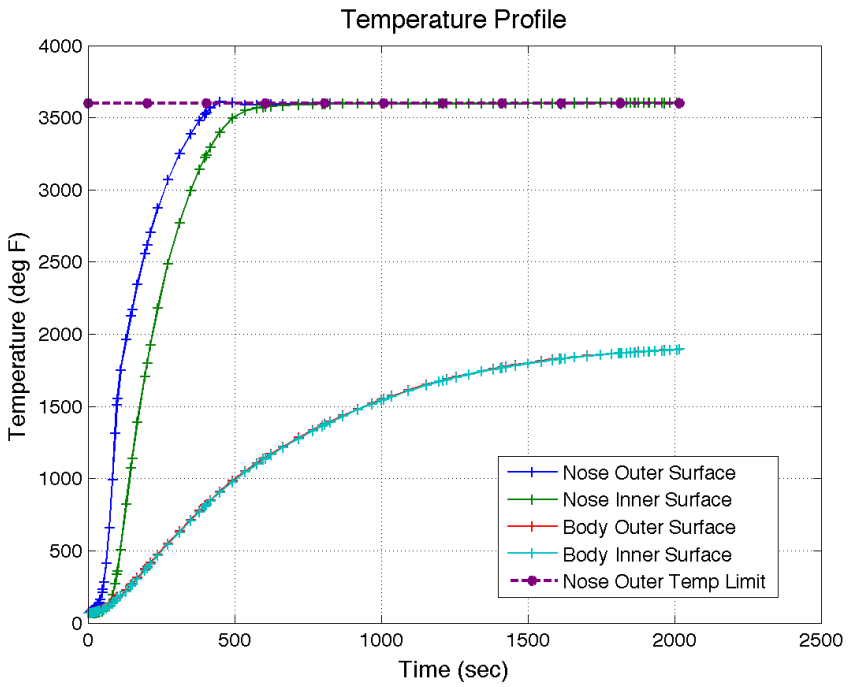


Figure 93. Optimal Trajectory: Minimum Time with Temperature Constraints (Climb and Cruise), Temperatures

constraint used, and it is set at the same 3600 °F as the previous scenario. For this scenario, it is a reduction of about 600 degrees from the maximum temperatures experienced in the same minimum time scenario (Section 5.2.6) without temperature constraints. Since the maximum body temperatures are not stressing, they again are not included as path constraints.

Unlike the other scenario with a temperature-based path constraint, this scenario uses IPOPT due to the use of multiple phases—SNOPT aborts execution citing numerical difficulties. While IPOPT is more robust, it is significantly slower for this scenario since IPOPT calls the objective and constraints functions much more often than SNOPT. Since MINIVER is executed the constraint function, it is called much more often than in SNOPT and results in extended run times. As previously discussed in Sections 3.5 and 5.2.9, MINIVER is executing its code only using single-precision values. To reduce the possibility of numerical instabilities, as mentioned in

the previous paragraph, the NLP solver tolerance was reduced to 10^{-4} and the mesh error tolerance was reduced to 10^{-2}. Even using these reduced values, an optimal solution was reached but the mesh tolerance was not met. To meet the tolerance, the limit for maximum number of mesh iterations would have to be increased.

Figure 94 shows that the addition of temperature constraints does not make a significant change to the ground track seen in the optimal trajectory without temperature constraints, Figure 63.



Figure 94. Optimal Trajectory: Minimum Time with Control Penalty and Temperature Constraints (Climb, Cruise, Refuel, Cruise, and Land), 3-D Flight Trajectory

Figure 95 shows a 2-D flight profile that has a significant scalloping (oscillation) in the height profile. This is due to the less stringent NLP solver and mesh error tolerances. While these were reduced to get a faster runtime (for this scenario, around 24 hours due to the execution of MINIVER code to calculate temperature path constraints), the reduced error tolerances resulted in a solution that is not refined as other solutions presented in this research.

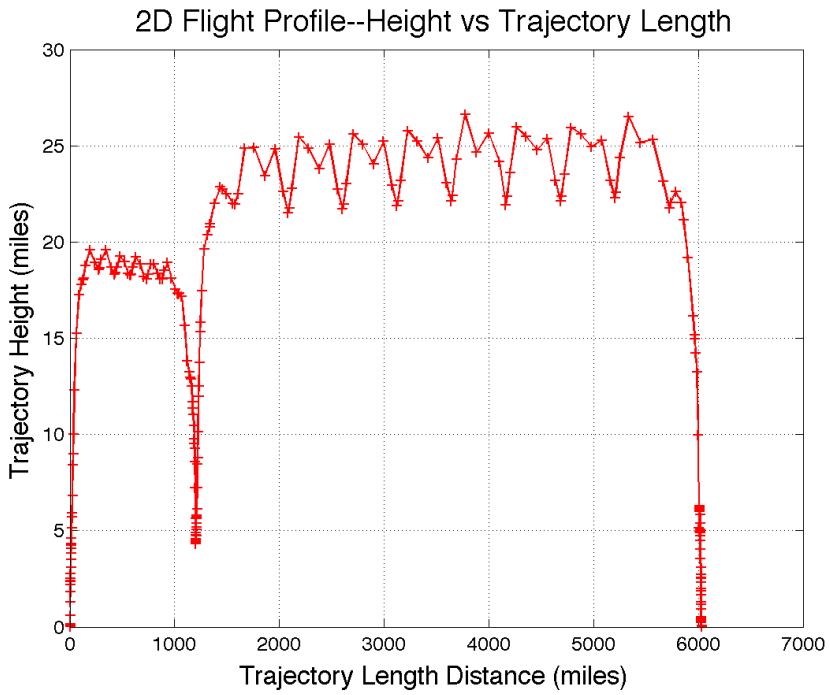


Figure 95. Optimal Trajectory: Minimum Time with Control Penalty and Temperature Constraints (Climb, Cruise, Refuel, Cruise, and Land), 2-D Flight Profile

Figure 96 shows the effect of introducing the temperature constraint. Compared to Figure 65, the maximum Mach number (velocity) is significantly decreased in both phases to reduce heating. In the first phase, the trajectory height is similar while in the second phase, it is significantly decreased. This behavior is different than the simpler temperature-based scenario (Section 5.2.9) and is due to the reduced error tolerances in this scenario.

Figure 97 shows the heating rates for the scenario. Typical for this vehicle, the body (windward fuselage, half way down vehicle) heating is much lower than at the nose. The maximum values for nose heating rate occur when the vehicle reaches cruise velocity in the first phase, since the cruise altitude in the first phase is lower than the second phase, resulting in higher atmospheric density. For this scenario, in the first phase, the difference between the nose heating rate from MINIVER and Chapman's equation is much higher than the typical 25 percent for level flight, since

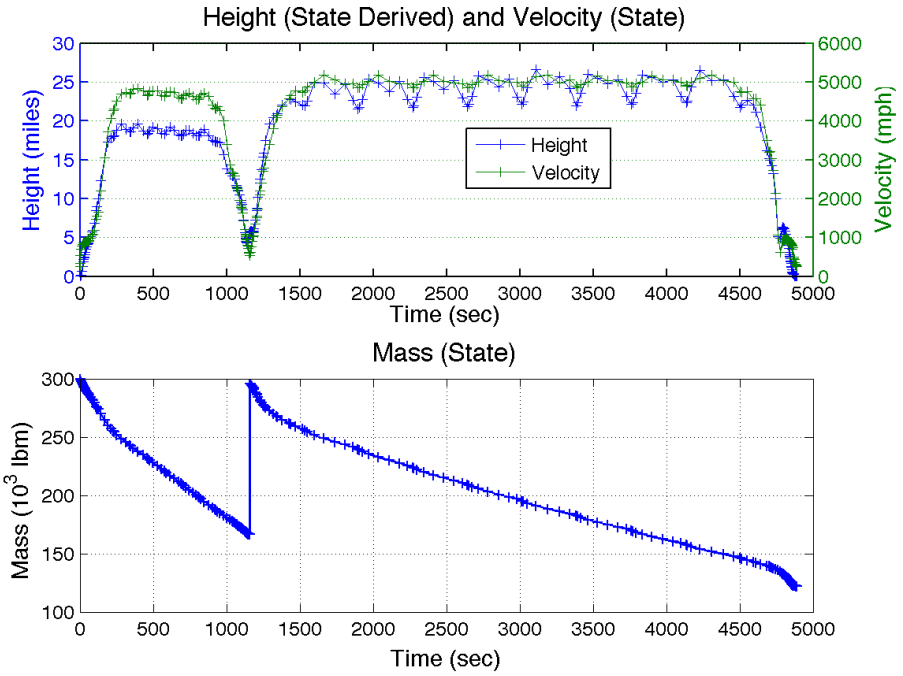


Figure 96. Optimal Trajectory: Minimum Time with Control Penalty and Temperature Constraints (Climb, Cruise, Refuel, Cruise, and Land), States

as previously discussed, Chapman's equation is more appropriate for higher altitude trajectories.

Figure 98 shows the heating loads for the flight profile. Since heating loads are just the integrals of heating rates (Equation 2), these curves are relatively linear for each phase, with the exceptions of the descent portions of flight, for both refueling and landing.

Figure 99 shows the temperature profile at the nose and body, calculated by MINIVER. For each fuselage position, two values are given, the outer and inner surfaces of the vehicle skin. For this scenario, the nose temperature was constrained to 3600 deg Fahrenheit. While the temperature after the refueling event is at this limit, the temperatures before the refueling event is slightly below this constraint. As mentioned previously in this section, due to lower NLP and mesh error tolerances, the temperature behavior in this scenario is different than the previous scenario, which

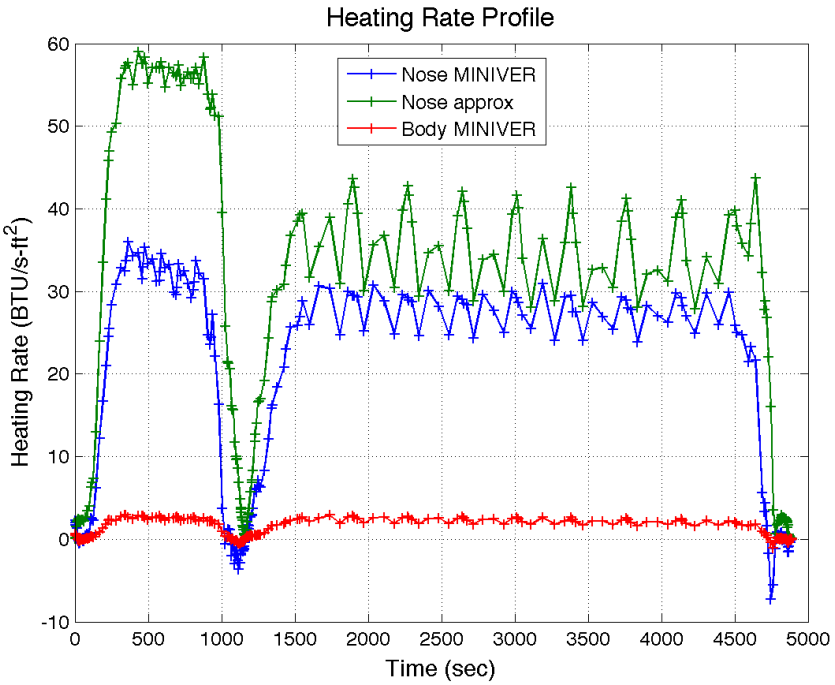


Figure 97. Optimal Trajectory: Minimum Time with Control Penalty and Temperature Constraints (Climb, Cruise, Refuel, Cruise, and Land), Heating Rates

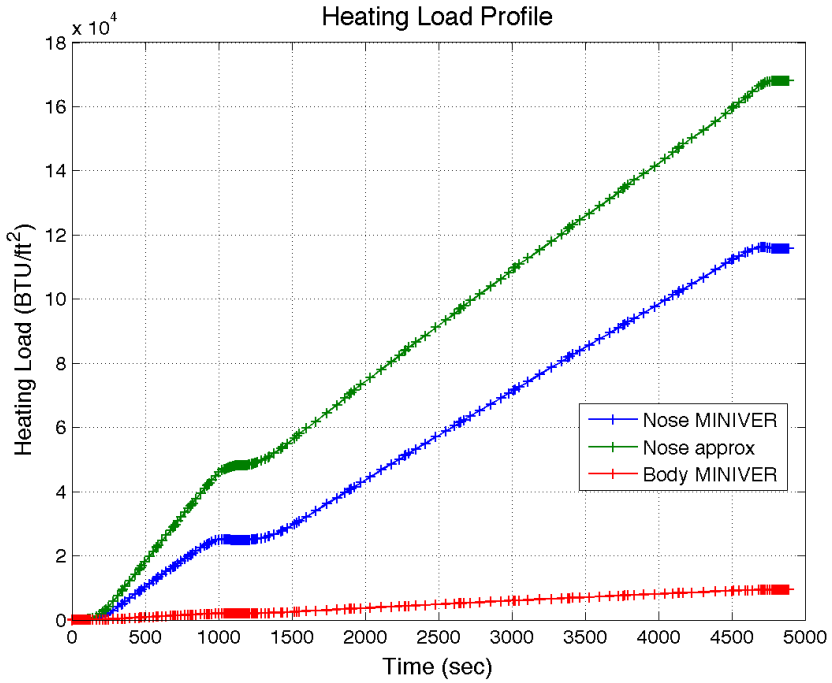


Figure 98. Optimal Trajectory: Minimum Time with Control Penalty and Temperature Constraints (Climb, Cruise, Refuel, Cruise, and Land), Heating Loads

had temperatures at the constraint value with higher NLP and mesh error tolerances.

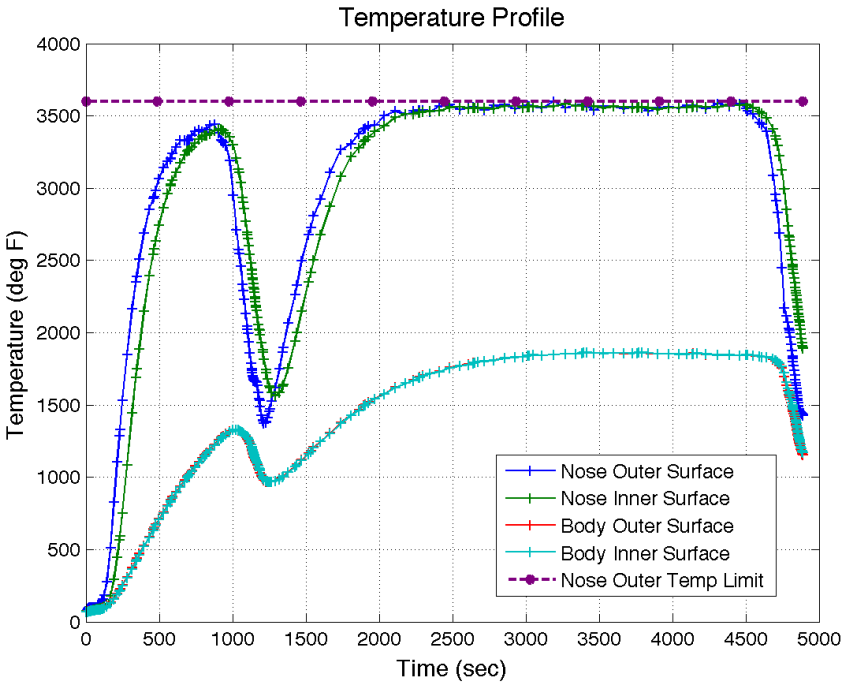


Figure 99. Optimal Trajectory: Minimum Time with Control Penalty and Temperature Constraints (Climb, Cruise, Refuel, Cruise, and Land), Temperatures

As seen in Figure 41, Maximum Relative Error default (10^{-2}) is not met but the general trend is leading towards meeting this value, albeit slowly. As mentioned previously, the number of mesh iterations can be increased at the risk of the solution diverging. Given the execution time for this scenario was about 24 hours, this scenario was not rerun to possibly improve the solution.

5.3 Rapid Mission Planning Utility

As previously discussed in Section 1.3.3, this research defines rapid mission planning as mission planning that could be completed in 30 minutes. As shown in Figure 101, 9 of the 10 scenarios implemented in this research meet this timeline. In general, the simpler scenarios easily fall within this timeliness parameter. The parameter is almost exceeded or exceeded when more complicated scenarios are implemented.

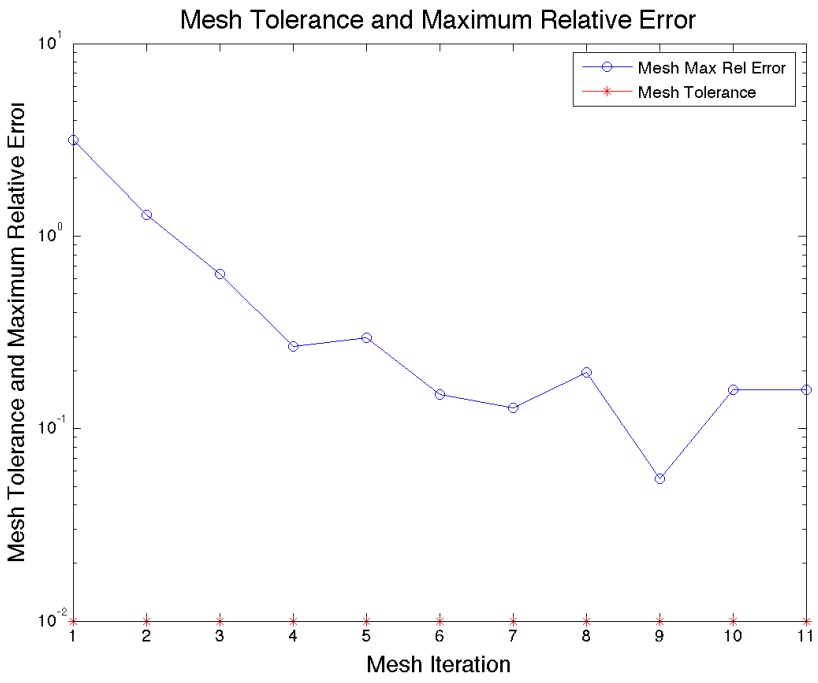


Figure 100. Optimal Trajectory: Minimum Time with Control Penalty and Temperature Constraints (Climb, Cruise, Refuel, Cruise, and Land), Maximum Relative Error

Scenario attributes that contribute to longer execution times include scenarios with stressing path constraints, scenarios with multiple phases, or scenarios with temperature path constraints.

In scenarios from Sections 5.2.6 and 5.2.8, the run times are driven by the multiple phases in the scenarios. Each phase drives a separate optimal control problem that will be solved by the NLP solver. For the scenario from Section 5.2.7, the run times are driven by both the multiple phases as well as the stressing path constraints, having both a no-fly zone and g-limits.

Both scenarios that incorporate temperature path constraints, Sections 5.2.9 and 5.2.10, have longer run times due to using MINIVER to calculate temperature path constraints. As discussed in Section 3.5, MINIVER is being run in MATLAB as two executable files (LANMIN and EXITS) that combined have a typical run time of 0.5–1 second. With the NLP solver being an iterative solver, LANMIN/EXITS could

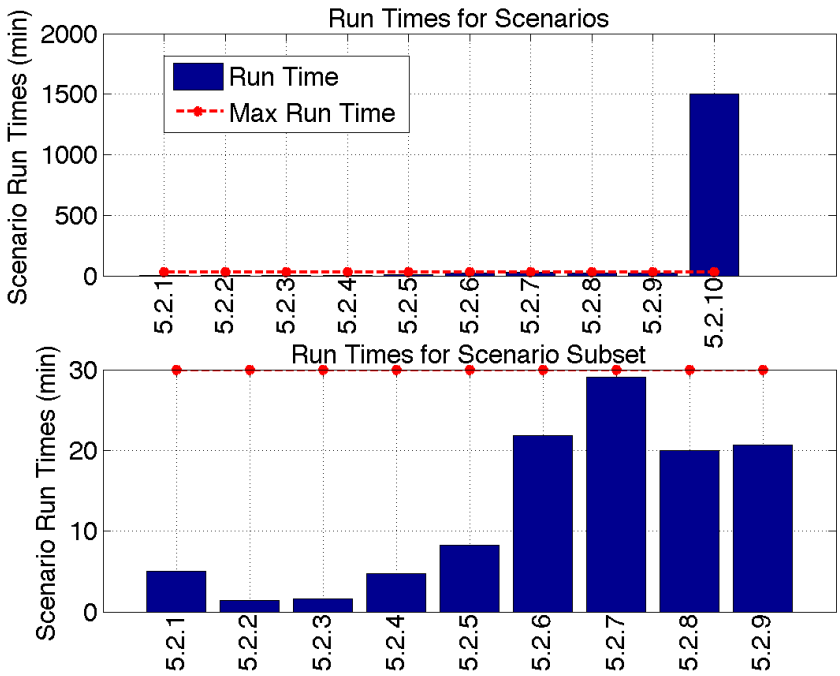


Figure 101. Scenario Run Times

Table 6. Scenario Timing Attributes

Scenario	Scenario Run Time (sec)	Phases	Temperature Path Constraints
5.2.1	303	1	0
5.2.2	84	1	0
5.2.3	99	1	0
5.2.4	286	1	0
5.2.5	493	1	0
5.2.6	1312	2	0
5.2.7	1742	2	0
5.2.8	1197	3	0
5.2.9	771	1	1
5.2.10	90125	2	1

be called thousands of times in a scenario run, becoming the significant driver in total scenario run time.

The scenario from Section 5.2.10 has by far the longest run time, due to having both a stressing temperature path constraint (a reduction of 600 °F from the same scenario without a temperature constraint) having a multiple phase formulation, and using IPOPT (vice SNOPT) with MINIVER in the constraint function formulation.

5.4 Results and Analysis Summary

In this chapter, the optimal control formulation and vehicle model were used to run several different type of scenarios. The results of these scenario runs were presented and discussed. As part of the discussion, the incorporation of aerothermal path constraints and sensor parameter path constraints and objective functions were described, to include the integration of MINIVER into the problem methodology.

This chapter demonstrated that optimal trajectories can determined using methodologies developed in this research for several objective functionals and several path constraints. The results verified that several path constraints like no-fly zones, g-limits, and sensor look angles can be used to implement mission parameters. Additionally, temperature path constraints can be implemented but the precision of the resulting optimal solution is limited due to the implementation of the MINIVER aerothermal code.

This chapter also demonstrated the usability of this research for rapid mission planning. While most of the scenarios implemented in this research can be labeled "rapid" for use as a mission planner, there are several caveats to this statement. A scenario with multiple complicated mission parameters that would drive a multiple phase solution could result in a lengthy run time. Also, a scenario with temperature path constraints could also result in length run times, depending on the severity of

the path constraint. Overall, this research has demonstrated a methodology that would be useful for rapid mission planning. In some of the cases that could result in lengthy mission planning cycles, insights from parametric analyses could provide simplifications that could result in faster mission planning capabilities. The following chapter discusses the parametric analysis and its applicability to this research.

VI. Parametric Analysis

THIS chapter documents the parametric analysis applying the methodology and models discussed in previous chapters for a specific scenario and varying the path length and maximum Mach number. As seen in Chapter VI, optimal trajectories can determined using methodologies developed in this research for several objective functionals and several path constraints. While the ability to generate optimal solutions was shown, some of the scenarios demonstrated that specific formulations could result in long run times, thus not supporting rapid mission planning.

As stated in Chapter I, one of the research objectives is to perform a parametric analysis to improve trajectory generation capabilities. In this chapter, this parametric analysis is performed for a specific scenario to examine trends and performance attributes that could provide insight to simplify problem formulation to accelerate trajectory generation run times.

6.1 Parametric Analysis Overview

The parametric analysis was run with the same hardware and software configuration described in Section 5.1 and Table 3. The state and control limits used in this analysis are the same as in Table 4

The goal of the analysis is to numerically compute results for each of the scenario specific settings, generating solutions that resemble typical flight profiles and constraints. By comparing the results from different path lengths and maximum Mach numbers, trends can be observed and analyzed.

The scenario used in this analysis is a minimum time climb and cruise scenario similar to the scenario analyzed in Section 5.2.2. The Mach numbers were varied from Mach 5 to 9 in 0.5 increments and the Great Circle Length between start and end

points were varied from approximately 500 to (statute) 3500 miles in approximately 500 mile increments.

6.2 Parametric Analysis Results

The data generated in this parametric analysis is presented as several sets of figures in Appendix D. As described in the previous section, the minimum time scenario was run varying both Mach number and Great Circle Distance, resulting in 63 optimal solutions. Each section in Appendix D aggregates these solutions for a specific scenario parameter: trajectory height, stoichiometric fuel-air ratio, dynamic pressure, g's, nose heating rate, nose heating load, nose temperature, body heating rate, body heating load, and body temperature. For each of these parameters, a 2D figure is provided varying the Great Circle Distance and then varying Mach number. The data from each of these 2D figures is then combined into a single 3D figure shown at the front of each section in Appendix D. Finally, for each of these parameters, the maximum value is found for each Great Circle Distance — Mach number combination and that information is portrayed in the 3D charts presented in this section.

6.2.1 Total Flight Time Profile.

Figure 102 shows that the relationship between total flight time and Great Circle Distance is roughly linear, which is expected since most of the flight time is spent in a near-constant velocity cruise. The relationship between total flight time and maximum Mach number is an inverse relationship since $time = distance/velocity$.

6.2.2 Maximum Height Profile.

Figure 103 shows that the relationship between maximum height and Great Circle Distance is roughly linear except at short flights and low Mach numbers. The rela-

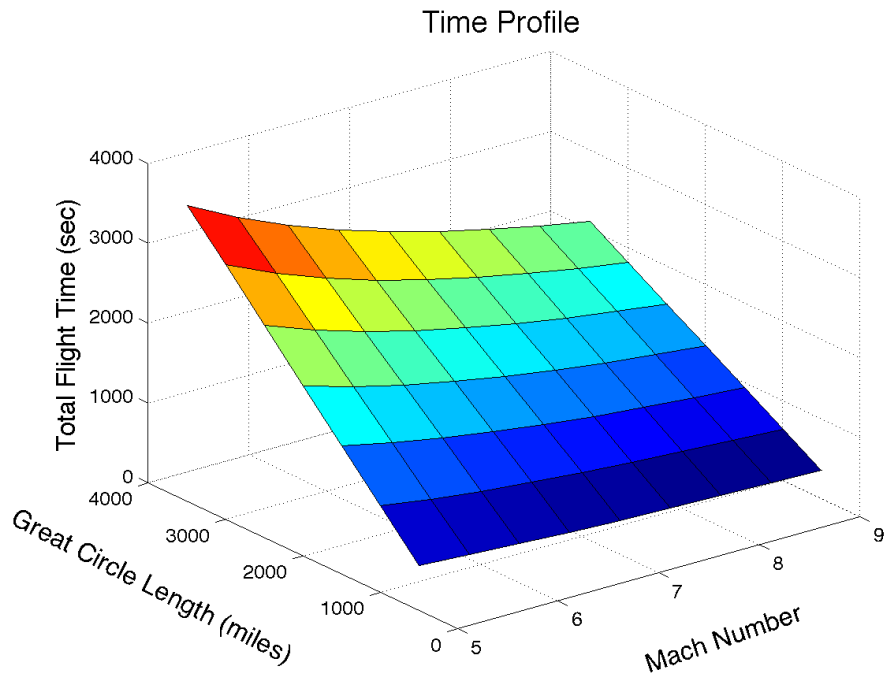


Figure 102. Total Flight Time vs. Mach Number and Great Circle Length

tionship between maximum height and maximum Mach number is also roughly linear except at short flights and low Mach numbers. At low Mach numbers, the vehicle is operating as a ramjet since the ramjet-scramjet transition occurs at Mach 6. For this vehicle, as shown in Figure 17, the engine I_{sp} is much higher for the ramjet mode than for the scramjet mode, thus the larger gradients at maximum Mach numbers under 6. In addition, as the flight gets longer, the time the vehicle takes to achieve a higher cruise altitude is less than the time savings in the rest of the scenario.

6.2.3 Maximum Velocity Profile.

Figure 104 shows that the relationship between maximum velocity and Great Circle Distance is roughly constant. The relationship between maximum velocity and maximum Mach number is linear since the maximum velocity is related to maximum Mach number and maximum height (drives speed of sound at that altitude). Since the

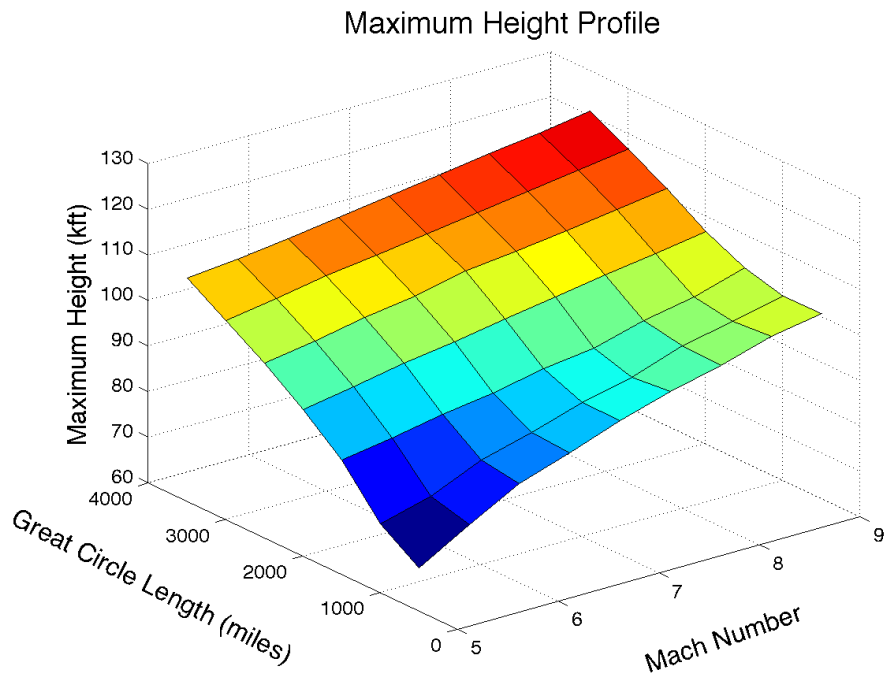


Figure 103. Maximum Height vs. Mach Number and Great Circle Length

maximum Mach number is a specified value, it is expected that the relationship with maximum velocity should be linear, since the vehicle is operating in an atmospheric regime that has a linear relationship between altitude and local speed of sound, as shown in Figure 10.

6.2.4 Maximum Dynamic Pressure Profile.

Figure 105[1] shows that the relationship between maximum dynamic pressure and Great Circle Distance is roughly constant except at short flights and low Mach numbers. The relationship between maximum dynamic pressure and maximum Mach number is roughly constant except at short flights or higher maximum Mach numbers. For shorter flights (shorter Great Circle Distance), the vehicle accelerates faster

[1]The Mach number scale is inverted here since this figure was rotated to show the behavior at longer great circle lengths

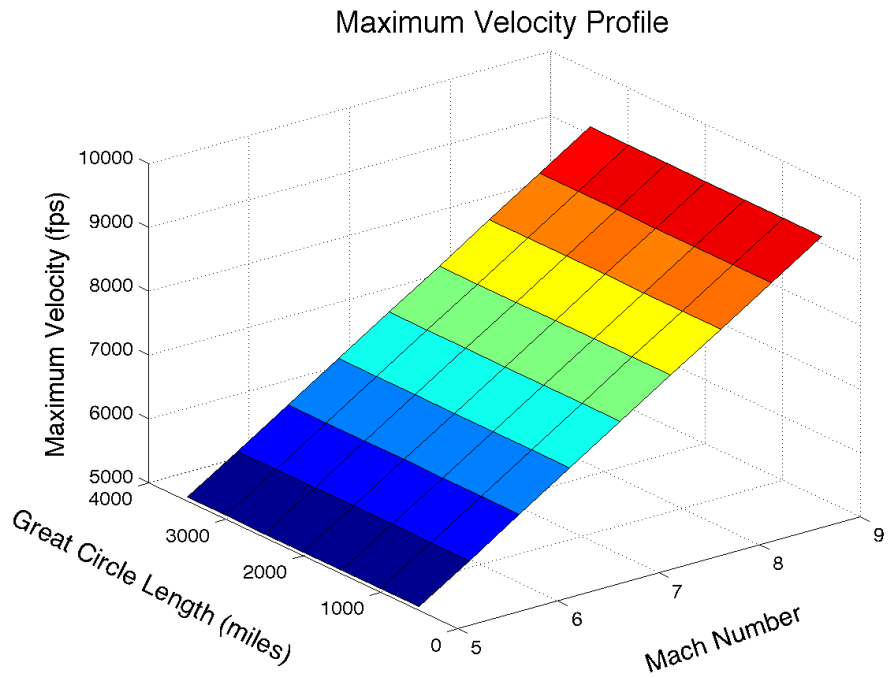


Figure 104. Maximum Velocity vs. Mach Number and Great Circle Length

because it burns more fuel during the ascent and less fuel during a shorter cruise period, since it has a short cruise distance. Since the vehicle has a higher velocity lower in the atmosphere, it experiences higher maximum dynamic pressures than if it has a longer cruise period.

6.2.5 Maximum g's Profile.

Similar to Figure 105, Figure 106 shows that the relationship between maximum g's and Great Circle Distance is roughly constant except at short flights. The relationship between maximum g's and maximum Mach number is also roughly constant except at short flights or higher maximum Mach numbers. For shorter flights or higher maximum numbers, the vehicle can climb more aggressively.

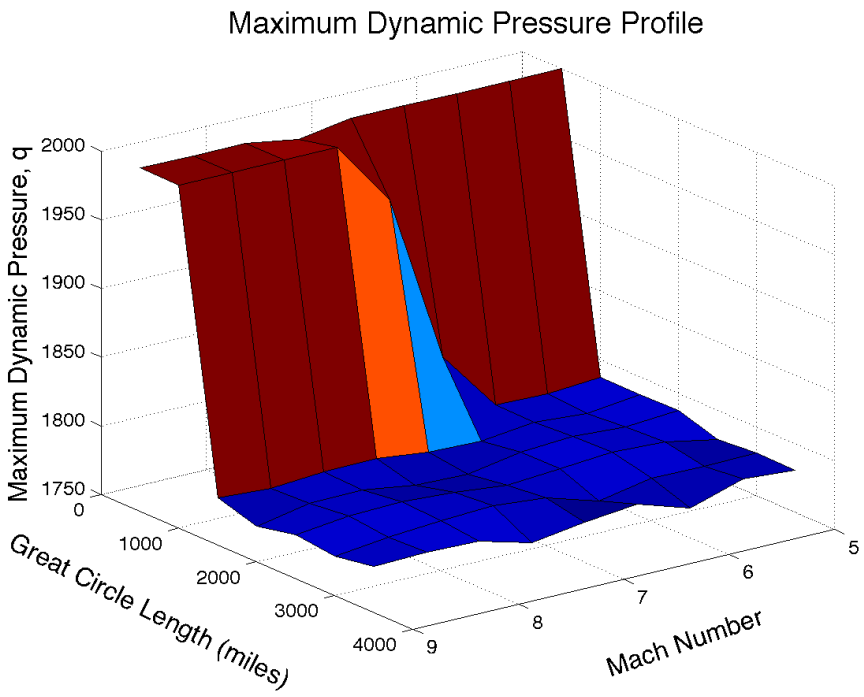


Figure 105. Maximum Dynamic Pressure vs. Mach Number and Great Circle Length

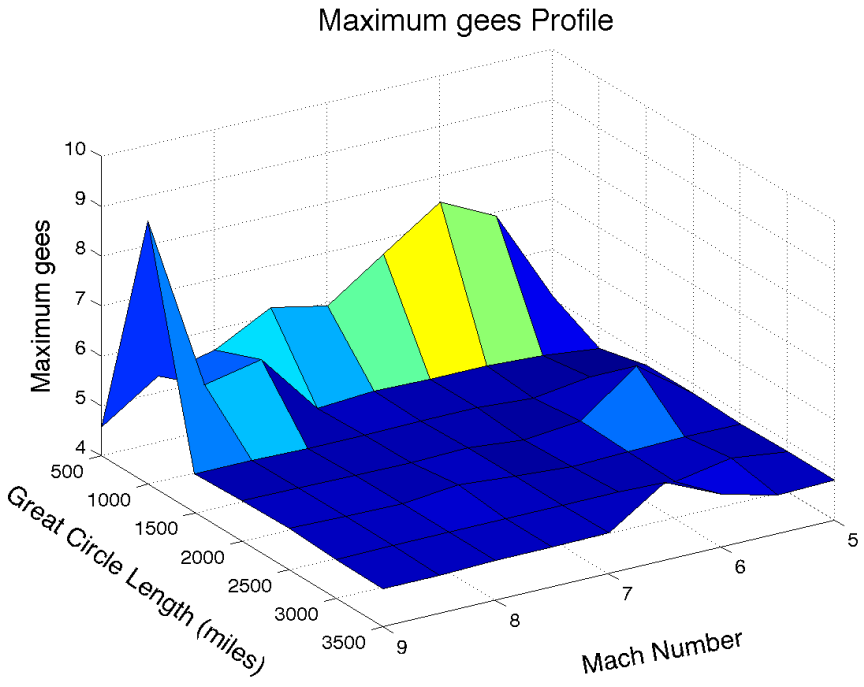


Figure 106. Maximum g's vs. Mach Number and Great Circle Length

6.2.6 Maximum Nose Heating Rate Profile.

Figure 107 shows that the relationship between maximum nose heating rate and Great Circle Distance is roughly constant. The relationship between maximum nose heating rate and maximum Mach number is approximated by a higher-order polynomial. Since nose heating can be approximated by Chapman's Equation (Equation 1) which includes V^3 as one of the terms, the higher-order polynomial relationship with maximum Mach number is expected.

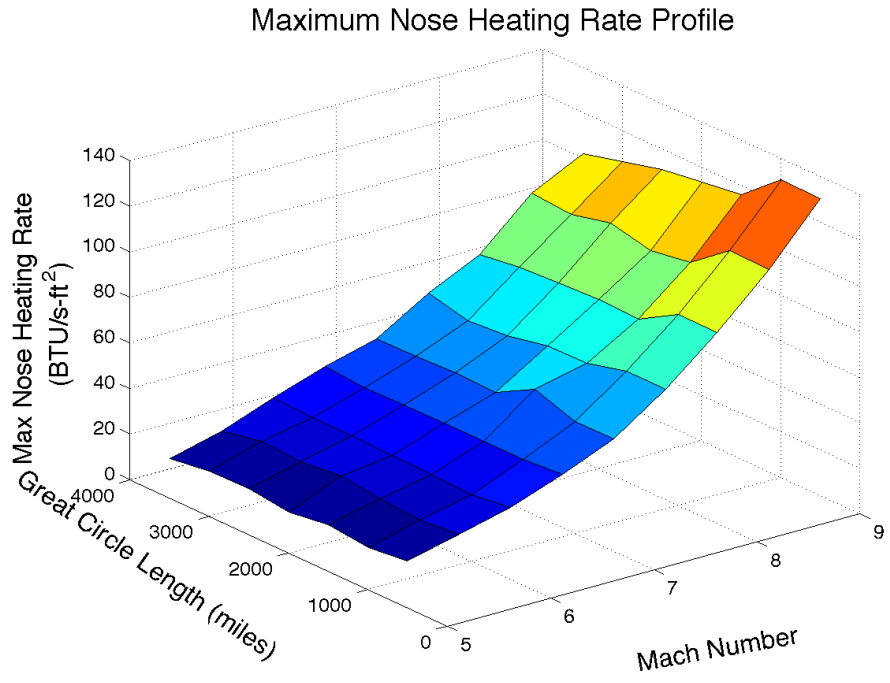


Figure 107. Maximum Nose Heating Rate vs. Mach Number and Great Circle Length

6.2.7 Maximum Nose Heating Load Profile.

The heating load is an integral of the heating rate (Equation 2). Figure 108 shows that the relationship between maximum nose heating load and Great Circle Distances is linear at low maximum Mach numbers and a higher-order polynomial

at high maximum Mach numbers. Similarly, the figure shows that the relationship between maximum nose heating load and maximum Mach number is linear at low Great Circle Distances and a higher-order polynomial at high Great Circle Distances. Since heating loads are not constrained, the vehicle can be subject to higher heating loads at longer flight times and higher maximum Mach numbers.

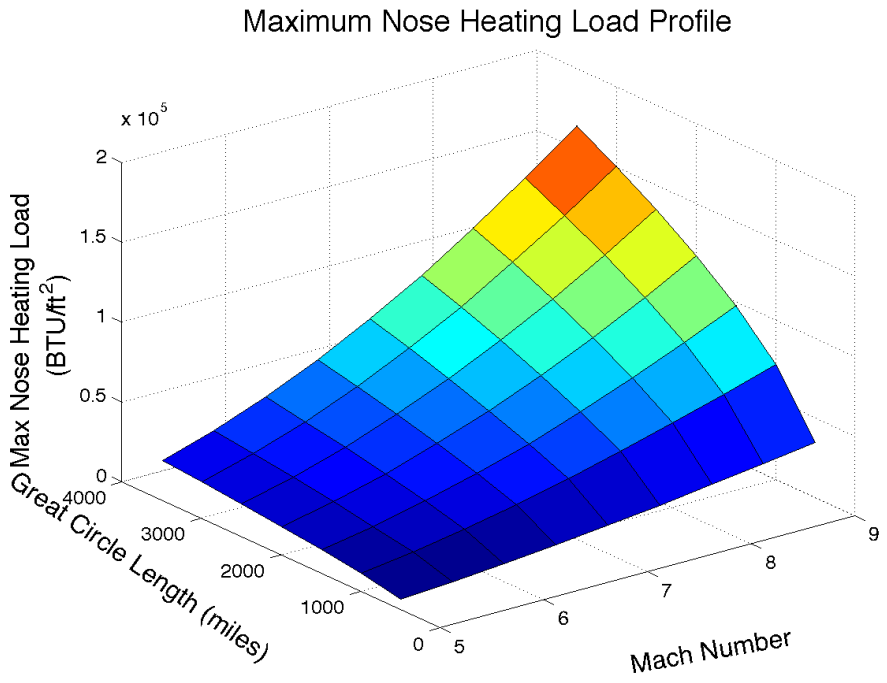


Figure 108. Maximum Nose Heating Load vs. Mach Number and Great Circle Length

6.2.8 Maximum Nose Temperature Profile.

Figure 109 shows that the relationship between maximum nose temperature and Great Circle Distance is roughly constant. The relationship between maximum maximum nose temperature and maximum Mach number is nearly linear. Since this a climb and cruise scenario, and much of the flight is spent in high-speed and altitude cruise, the vehicle quickly gets to the maximum temperature (due to the vehicle

achieving a thermal balance early in cruise). At higher Mach numbers, the flowfield generates additional thermal energy resulting in a higher maximum nose temperature.

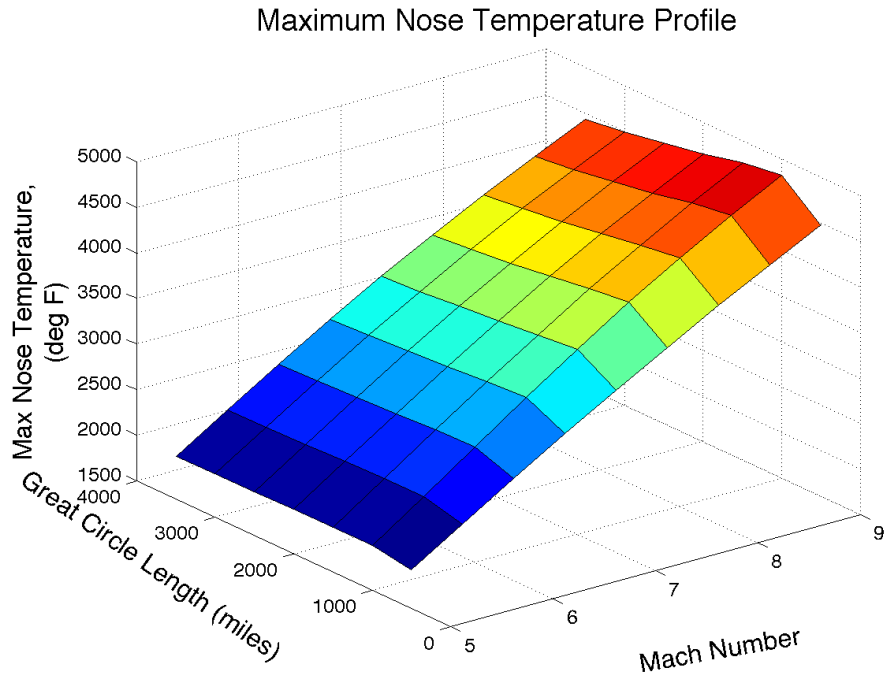


Figure 109. Maximum Nose Temperature vs. Mach Number and Great Circle Length

6.2.9 Maximum Nose Temperature Duration Profile.

Figure 110 shows the percentage of flight time spent at close to the maximum nose temperature. The relationship between maximum nose temperature duration and Great Circle Distance is almost linear. The relationship between maximum nose temperature duration and maximum Mach number is constant. Since this a cruise and climb scenario, and much of the flight is spent in high-speed and altitude cruise, the vehicle quickly gets to the maximum temperature (due to the vehicle achieving a thermal balance early in cruise).

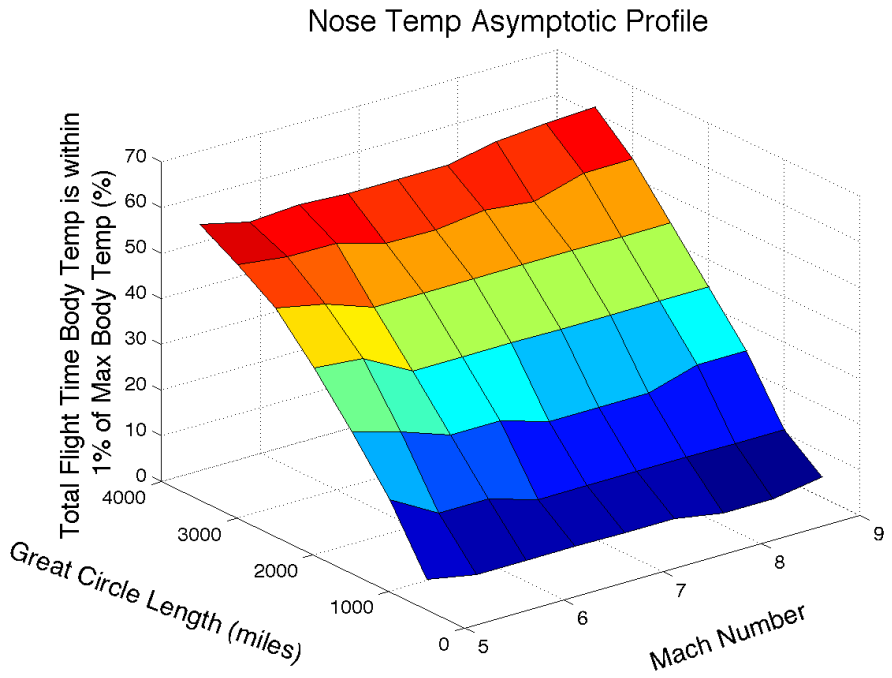


Figure 110. Maximum Nose Temperature Duration vs. Mach Number and Great Circle Length

6.2.10 Maximum Body Heating Rate Profile.

Figure 111 shows that the relationship between maximum body heating rate and Great Circle Distance is roughly constant except at high Mach numbers. The relationship between maximum maximum body heating rate and maximum Mach number is roughly constant except at high Mach numbers except at lower Great Circle Distances. For shorter flights or higher maximum numbers, the vehicle can climb more aggressively, resulting in higher heating rates.

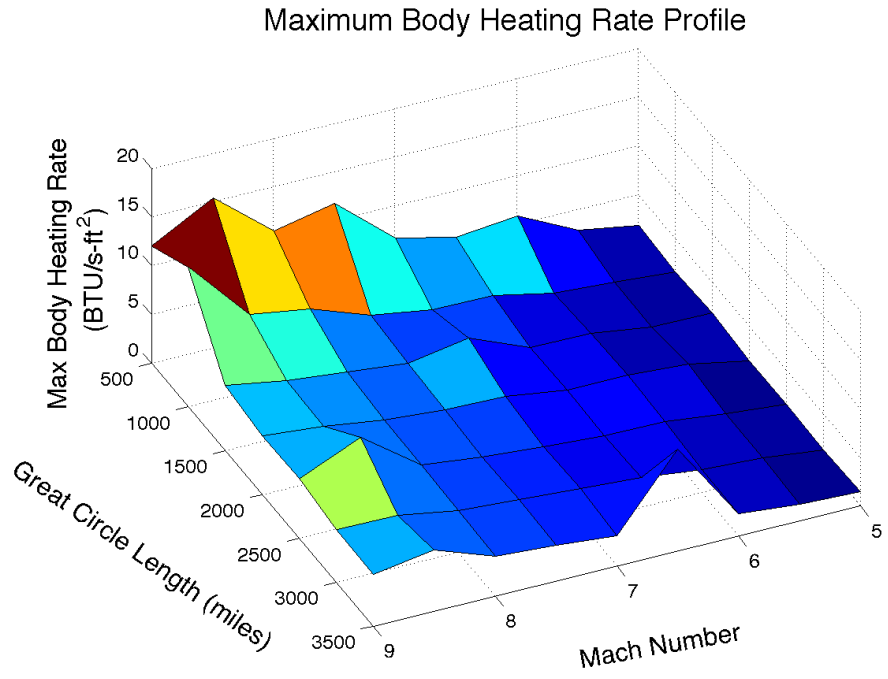


Figure 111. Maximum Body Heating Rate vs. Mach Number and Great Circle Length

6.2.11 Maximum Body Heating Load Profile.

As previously stated, the heating load is an integral of the heating rate (Equation 2). Figure 112 shows that the relationship between maximum body heating load and Great Circle Distances is linear at low maximum Mach numbers and a higher-

order polynomial at high maximum Mach numbers. Similarly, the figure shows that the relationship between maximum body heating load and maximum Mach number is linear at low Great Circle Distances and a higher-order polynomial at high Great Circle Distances. Since heating loads are not constrained, the vehicle can be subject to higher heating loads at longer flight times and higher maximum Mach numbers.

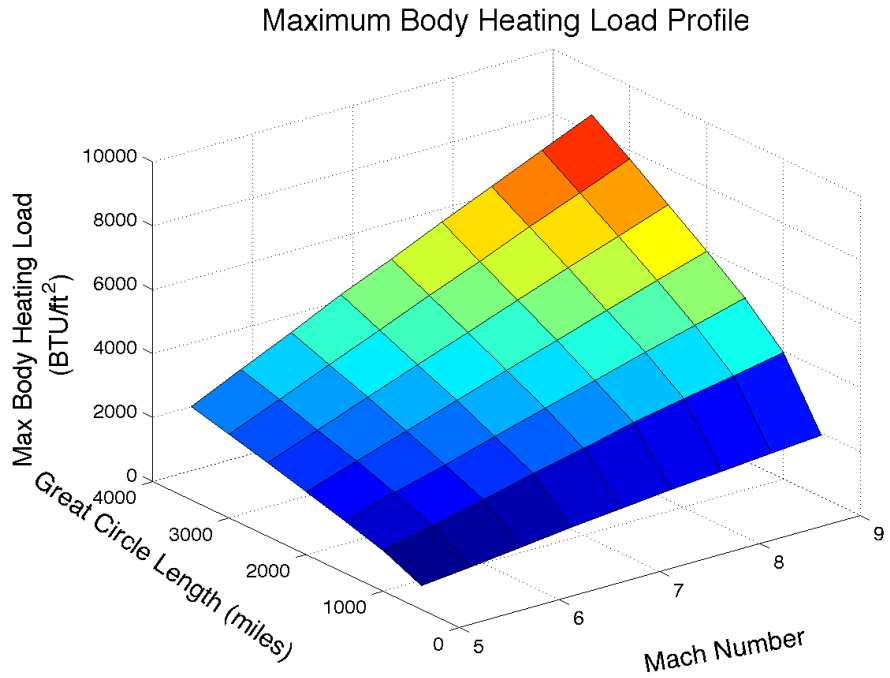


Figure 112. Maximum Body Heating Load vs. Mach Number and Great Circle Length

6.2.12 Maximum Body Temperature Profile.

Similar to Figure 109, Figure 113 shows that the relationship between maximum body temperature and Great Circle Distance is roughly constant. The relationship between maximum body temperature and maximum Mach number is nearly linear except at short great circle distances. Since this a cruise and climb scenario, and much of the flight is spent in high-speed and altitude cruise, the vehicle quickly gets to the maximum temperature (due to the vehicle achieving a thermal balance early in

cruise). At higher Mach numbers, the flowfield generates additional thermal energy resulting in a higher maximum body temperature.

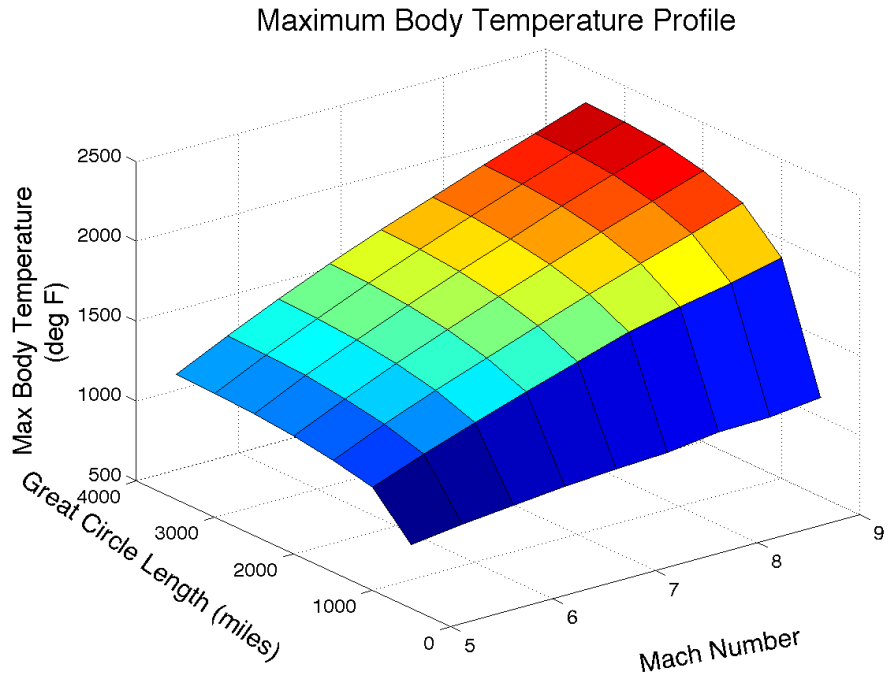


Figure 113. Maximum Body Temperature vs. Mach Number and Great Circle Length

6.2.13 Maximum Body Temperature Duration Profile.

Similar to Figure 110, Figure 114 shows the percentage of flight time spent at close to the maximum body temperature. The relationship between maximum body temperature duration and Great Circle Distance is a lower order polynomial. The relationship between maximum maximum body temperature duration and maximum Mach number is roughly constant. Since this a cruise and climb scenario, and much of the flight is spent in high-speed and altitude cruise, the vehicle quickly gets to the maximum temperature (due to the vehicle achieving a thermal balance early in cruise).

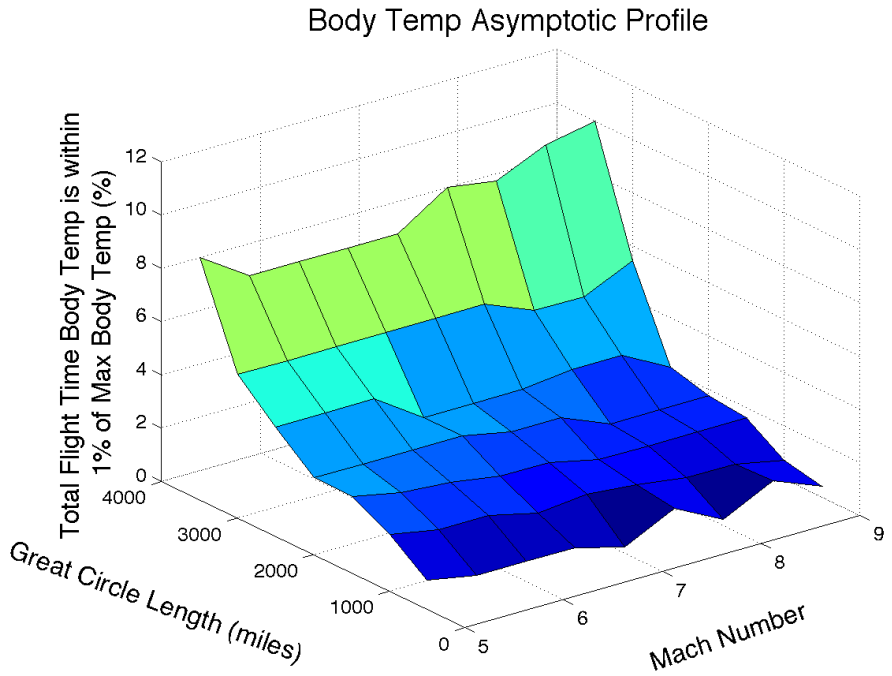


Figure 114. Maximum Body Temperature Duration vs. Mach Number and Great Circle Length

6.3 Parametric Analysis Insights

As shown in Section 6.2.8, the relationship between maximum nose temperature and maximum Mach number is nearly linear. While this analysis is only applicable to this scenario, this insight can be used to change the formulation of the scenario to improve trajectory generation times. Specifically, by removing the temperature-based path constraint and lowering the maximum Mach number (and as a result, maximum velocity) allowed, the maximum nose temperature can be indirectly used as a path constraint.

Figure 115 shows the relationship between flight times and maximum nose temperatures for various maximum Mach numbers. This data can be used to associate a maximum nose temperature with a maximum Mach number, and subsequently a total flight time. For example, in this figure, assume that the desired maximum nose temperature is 3600 °F. This condition is met if the maximum Mach number is set to approximately 7.47, which then results in a total flight time of approximately 2044 seconds. Thus, instead of including a temperature-based trajectory constraint, the maximum Mach number can be reduced to implement the same temperature constraint. In this case, using this simplification results in an almost order of magnitude reduction in scenario run times, as shown in the run times for similar scenarios in Sections 5.2.2 (no temperature-based path constraint) and 5.2.9 (temperature-based path constraint) in Table 6, 771 vice 84 seconds, respectively. While this specific result is applicable to this scenario only, a similar effort can be performed a priori and the results used for subsequent analysis.

Figure 116 shows the difference in flight times for the same scenario, whether it has a temperature-based path constraint, or it has the equivalent reduced maximum Mach number derived by the procedure in the previous paragraph. As shown the figure, the difference between the two approaches is smaller when the temperature

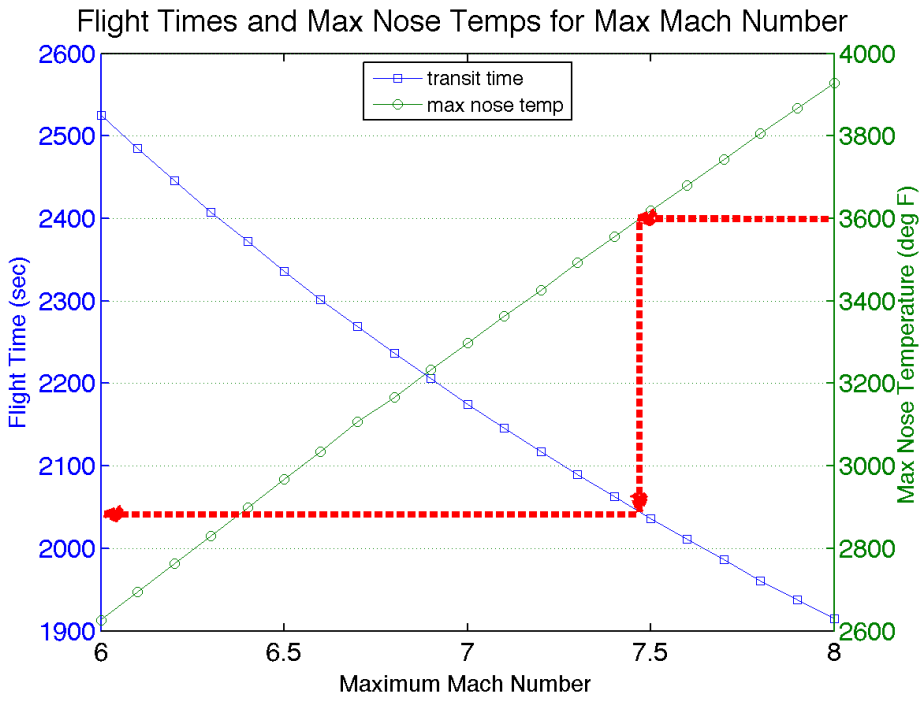


Figure 115. Flight Times and Maximum Nose Temperatures vs. Mach Number

constraint is near the maximum nose temperature without a temperature-based path constraint, but increases as the constraint is set to lower values. The difference in the two approaches increases slightly since as the temperature constraint is set lower, the trajectory's cruise velocity is reduced to a lower value (to generate less heat rate/load) earlier in the trajectory, resulting in lower flight times. Figure 117 shows the difference in flight times over desired maximum nose temperatures.

6.4 Parametric Analysis Summary

This chapter presented a parametric analysis for a simple minimum time climb and cruise scenario. While most of the observed relationships were expected or obvious, only two relationships were very insightful; the relationships between maximum body and nose temperature and maximum Mach number is linear. This insight can be

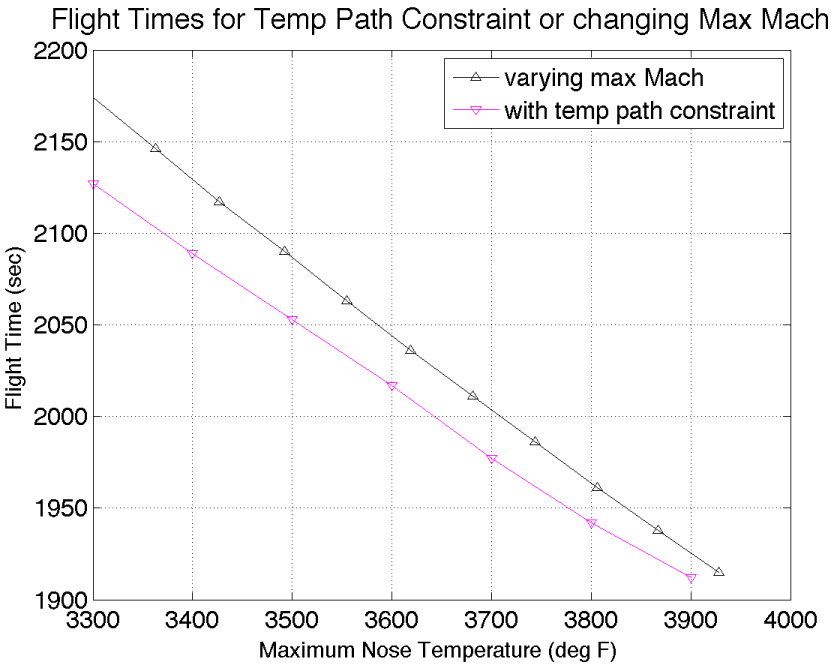


Figure 116. Flight Times for Either Temperature Path Constraint or Changing Maximum Mach Number

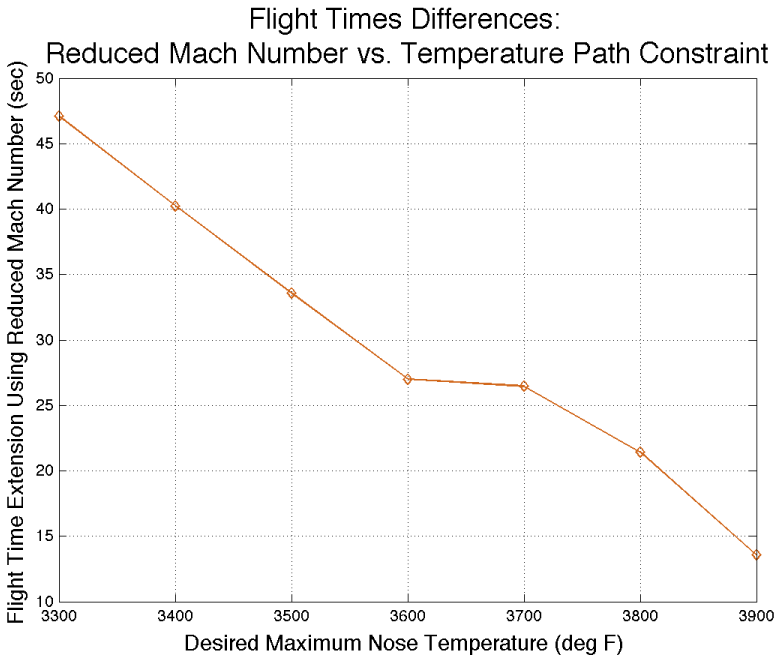


Figure 117. Flight Times Differences Between Temperature Path Constraint and Changing Maximum Mach Number

used to simplify the two temperature-based path constraint scenarios in Sections 5.2.9 and 5.2.10. With this insight, the temperature path constraint can be replaced by a lower maximum Mach number (maximum velocity). While this parametric analysis was completed for this minimum time climb and cruise scenario, the insight can be extended to many scenarios since a typical scenario starts with a take-off from a runway, climb to cruise altitude then cruise at a near-constant altitude and velocity. More complicated scenarios are typically multi-phase with the phased parts similar to this climb and cruise scenario. Therefore, with this simplification, trajectory generation becomes simpler and quicker, supporting the rapid mission planning objective.

VII. Conclusions and Recommendations

7.1 Conclusions

THIS research successfully developed a methodology for temperature-constrained optimal trajectories for a scramjet-based hypersonic reconnaissance vehicle that can support rapid mission planning. The methodology and results of this research built upon existing research efforts by increasing model fidelity and incorporating a robust aerothermal analysis tool to compute vehicle Thermal Protection System (TPS) temperatures. As a result, the work presented here successfully addresses the research objectives discussed in Chapter I:

- Integration of a higher-fidelity hypersonics dynamics model into an optimal control formulation
  - This research successfully implemented a higher-fidelity model to model an air-breathing, hypersonic vehicle. Increased fidelity features include: rotating, spherical earth and an accurate, non-linear programming (NLP) solver-compatible standard atmosphere. The Generic Hypersonic Aerodynamic Model Example (GHAME) hypersonic vehicle model was implemented using a spline-based solver to maintain modeling accuracy while being compatible with NLP solvers. The generated solutions were highly sensitive to the interpolation methods used incorporating vehicle model and atmospheric lookup tables. The vehicle model was incorporated into an optimal control formulation that includes constraints on both the vehicle as well as mission parameters. Optimal trajectories were developed using several different performance costs in the optimal control formulation: minimum time, minimum time with control penalties, and maximum range. The resulting analysis demonstrated that optimal trajectories that meet specified mission parameters and constraints can be quickly determined and used for larger-scale operational and campaign planning and execution.
- Integration of a moderate-fidelity aerothermal model into an optimal control formulation
  - This research used MINIVER aerothermal design tool to calculate the temperature-based path constraints within MATLAB. While this tool has a higher-fidelity than the typical aerothermal heat rate approximations, it

still is only considered medium fidelity. The MINIVER FORTRAN 66 had to be modified to avoid the truncation error due to the use of ASCII data files. With the changes to support this research, the MINIVER code is still using input/output data files that are single-precision floating-point. Since MATLAB runs at double-precision, this MINIVER limitation resulted in limits on the user-defined error tolerances for both the NLP solver and the GPOPS adaptive mesh capability, driving the solution precision. Even with these limitations, research methodology was successfully able to implement and demonstrate temperature-based path constraints in several scenarios.

- Integration of sensor constraints into an optimal control formulation
  - This research demonstrated the use of sensor geometric limitations as trajectory path constraints. These sensor-based path constraints were successfully used to ensure that the optimal trajectory would support specified target elevation constraints and sensor slant angles. While sensor azimuth and elevation look angles were not implemented as path constraints, they were shown to be applicable once the limitation of look angle calculation piecewise continuity will be resolved by future research.
- Implement a parametric analysis to examine trends and performance attributes that could simplify or accelerate trajectory generation
  - This research conducted a parametric analysis on a specific scenario to identify vehicle or optimal trajectory attributes that could simplify and/or accelerate generation of optimal trajectories. While most of the observed relationships were expected, only one insightful relationship was used to modify the optimal control formulation. The relationship between maximum body and nose temperature and maximum Mach number was found to be nearly linear. With this insight, the temperature path constraint can be removed with a reduction of the user-specified maximum Mach number to reduce the maximum allowable velocity. While this parametric analysis was completed for a specific scenario, the insight can be extended to many scenarios since a typical scenario profile includes a similar profile to the analyzed scenario. With this simplification, trajectory generation becomes simpler and quicker, supporting the rapid mission planning objective, with only a small reduction in the optimality of the solution.

While this work successfully met the previous research objectives, there are inherent limitations in the employed methodology. While a three degrees of freedom (3DOF) state and dynamics model was adequate for this research, a full six degrees of freedom (6DOF) model would increase the analysis fidelity. Another approach to

increase analysis fidelity is the to use a vehicle model that is more accurate and better models the transitions between engine cycles. While successfully integrated, a faster and higher fidelity aerothermal model would greatly improve the usefulness of the analysis, especially if the current single-precision floating point precision limitation could be resolved. To better model the effects of sensor limitations, a generic sensor could be used to model sensor limitations instead of modeling just sensor geometric constraints. Finally, to improve the computations speed of this methodology, the tools and developed code could be moved from MATLAB to a programming language such as C.

As a result of this research, optimal trajectories for hypersonic vehicles can be generated with a wide variety of objective functionals and path constraints, especially vehicle temperatures. Ultimately, is it reasonable to craft a rapid mission planner and systems engineering tool with the methodology developed in this research? As described earlier in this section, the answer is yes except for the most stressing scenarios, to include scenarios with temperature-based path constraints, due to the aerothermal model used, as well as the scenarios with several phases and complicated path constraints. With the implementation of the findings from the parametric analysis, this methodology is even better suited for rapid mission planning. To address the most complicated scenarios, upgrading the hardware from a consumer-grade desktop to a workstation would reduce scenario run times even further.

7.2 Contributions

This research resulted in several contributions. First, it developed a framework to quickly develop optimal trajectories for a hypersonic air-breathing vehicle based on existing system and experimental data, using an optimal control formulation and incorporating temperature path constraints. Several increased-fidelity modeling fea-

tures were included: a rotating, spherical earth; g-limits, higher fidelity atmospheric modeling; and higher fidelity vehicle modeling. Several different types of path constraints were also incorporated, including specified takeoff/landing locations, airborne refueling constraints, specified no-fly zones, and specified targets for sensor data collections. The resulting analysis demonstrated that optimal trajectories that meet specified mission parameters and constraints can be quickly generated for mission planning and systems engineering efforts.

7.3 Recommendations for Future Research

There are several recommendations for future work.

- Improved modeling of aerodynamic and propulsion coefficients
  - While the Generic Hypersonic Aerodynamic Model Example (GHAME) model was available and open source, it had significant issues that impacted the research results. Better and more vehicle-specific models for aerodynamic and propulsion terms would eliminate some of the modeling limitations observed in the generated results. While the model may not be releasable to the public, one model could be developed for government use and promoted across government researchers.
- Improved modeling of transition between engines modes/cycles
  - While GHAME modeled Turbine Based Combined Cycle (TBCC) transitions between engine cycles (turbine-to-ramjet-to-scramjet), the data set was sparse in these regions, which resulted in transients observed in the generated results. Using higher fidelity engine models would be necessary for more detailed research and analysis. Like in the improved modeling of aerodynamic and propulsion coefficients recommendation, better and more vehicle-specific transition models would eliminate some of the modeling limitations observed in the generated results.
- Improved techniques of implementing lookup tables in a methodology that requires $\mathbb{C}^2$
  - There are many techniques in the community to take a dataset and make its first and second derivatives continuous (required for NLP solvers in this research). Many researchers just use polynomial curve fits which are a low to medium fidelity approach. Using various types of spline fits can result

in more accurate modeling but also can introduce higher-order effects. A follow-on research effort that explores the strengths and weakness of data-fitting techniques as well as their applicability in using would tabulated data sources with NLP solvers would be beneficial for future efforts.

- Improved tools to quickly calculate vehicle aerothermal parameters
  - While using MINIVER (Miniature Version) for temperature calculations was eventually successful, there are inherent limitations in the MINIVER code. In addition using MINIVER as an executable within MATLAB is computationally very inefficient. A follow-on research effort that develops or adapts an aerothermal analysis tool that supports rapid trajectory calculations and compatible and can be run in multiple environments including MATLAB would be very beneficial.
- Improved methods to determine sensor look angle constraints
  - To use sensor look angles as path constraints in a formulation using gradient-based solvers (used in this research), the computation of sensor look angles need to generate results that are compatible with iterative NLP solvers. For example, while the can generate results that are continuous and differentiable on a limited range, they cannot be piecewise continuous when extended beyond this range. For example, as seen in this research, computed look angles can jump between -π and π for an arbitrary candidate trajectory since they are computed using $\tan^{-1}$ with a range $(-\pi, \pi)$. A follow-on research effort that investigates different approaches to represent angles such as quaternions, would allow the expanded use of angular computations in both objective functionals and path constraints.
- Improved methods to determine hypersonic flowfield impacts on operating electro-optical (EO) sensors
  - Since this research is focused on trajectory optimization and not hypersonic aerodynamics or EO sensor development, this research incorporates only the effects of trajectory geometries on sensors, Thus, this research did not include thermal and flowfield effects on the sensor which should be candidates for future research topics.

Appendix A. Derivation of Chapman Equation Constants

As discussed in Chapter II, Chapman's equation [17] is frequently used as a simplified method to calculate stagnation point heating rates for reentry and high-velocity air-breathing vehicles. While most of the derivation and implementation of Chapman's equation is well documented in the researched cited previously (e.g., Nizami [58]), the derivation of the Chapman equation constants for specific implementation is not well defined in literature.

From Chapman's well-known 1958 paper, he approximates the stagnation point heating rate as:

$$\dot{Q}_{conv} \propto \sqrt{\rho} V^3 \tag{46}$$

In this equation, both of the power terms (1/2 and 3) as well as the implicit constant of proportionality can be based on enthalpy theory, flowfield conditions and shock tube experiments for a vehicle near orbital velocity. As discussed by Chapman, both power terms have common values for a wide range of vehicles but the constant of proportionality is not well defined for vehicles having velocities much less than orbital velocity.

To determine a general equation for the constant of proportionality in Chapman's equation, Hankey used several flowfield approximations for hypersonic flows to derive the following equations for air, assuming a constant specific heat ratio[1], γ_r and defining θ_i as the polar angle that specifies the stagnation point on the leading edge, and R as the nose radius [38].

$$\dot{Q}_{conv} = \frac{K\sqrt{\rho} V^3 F}{\sqrt{R}} \frac{\text{BTU}}{\text{ft}^2\text{sec}} \tag{47}$$

[1]Specific heat is a function of temperature, but will be assumed constant for this derivation. If the specific heat ratio is not constant, then an iterative solution technique would have to be used since skin temperature is a function of heat flux which is a function of temperature.

where, with A as a laminar skin friction coefficient constant ($A = 0.332$), J conversion factor ($J = 778 ftlb/BTU$), and C_μ viscosity coefficient ($C_\mu = \left(\mu_0/\sqrt{C_p}\right)^{.5} = \left(2.27e-8/\sqrt{6006}\right)^{.5}$)

$$\begin{aligned} K &= \frac{A[2\gamma_r/(\gamma_r-1)]^{1/2}(C_\mu)}{J2^{1.25}} \\ &= \frac{.332[2\gamma_r/(\gamma_r-1)]^{1/2}\left(2.27e-8/\sqrt{6006}\right)}{778*2^{1.25}} \end{aligned} \tag{48}$$

$$F = \frac{\left(\cos^{((1+\gamma_r)/2\gamma_r)}\theta_i\right)\left(1-\cos^{(2(\gamma_r-1)/\gamma_r)}\theta_i\right)^{1/4}}{\theta_i^2} \tag{49}$$

Since, $\lim_{\theta_i \to 0} \cos^{((1+\gamma_r)/2\gamma_r)}\theta_i = 1$ and applying the Taylor expansion theorem for $\cos\theta$, F simplifies to:

$$\lim_{\theta_i \to 0} F = \lim_{\theta_i \to 0} \sqrt{\frac{\sqrt{1-\cos^{(2(\gamma_r-1)/\gamma_r)}\theta_i}}{\theta_i}} \approx \lim_{\theta_i \to 0} \sqrt{\frac{\sqrt{1-\left(1-\frac{\theta^2}{2}\right)^{2(\gamma_r-1)/\gamma_r}}}{\theta_i}} \tag{50}$$

Applying a binomial expansion and simplifying,

$$\lim_{\theta_i \to 0} F \approx \sqrt[4]{\frac{\gamma_r - 1}{\gamma_r}} \tag{51}$$

Assuming a constant specific heat ratio for air, $\gamma_r = 1.4$, results in $F = .7330$ and $K = 8.1208e-9 \frac{BTU \sec^{1.5}\sqrt{^\circ R}}{ft^3\sqrt{lb}}$. With both F and K defined, Chapman's equation can be used for any applicable scenario.

Appendix B. Linear-Quadratic Optimal Control Problem Example

As part of the Air Force Institute of Technology (AFIT) AERO899 Independent Study course [75], Suplisson, Smith and Masternak studied example problems that are related to using pseudospectral methods to solve optimal control problems. One of these problems was a linear-quadratic optimal control problem (Linear Quadratic Regulator (LQR)-type problem) used by Pietz [62] in his research. Pietz used the problem formulation and solution as an example problem throughout his thesis. This example is a good comparison showing the accuracy of both a 'generic' pseudospectral solver and also General Pseudospectral Optimal Control Software (GPOPS) when compared to an exact analytical solution.

Consider the following linear-quadratic optimal control problem

$$\text{Minimize: } \int_0^1 y(t)^2 + \tfrac{1}{2}u(t)^2 dt \tag{52}$$

Subject to:

$$\dot{y}(t) = \tfrac{1}{2}y(t) + u(t),\ t \in [0,1] \tag{53}$$

$$y(0) = 1$$

The solution as derived by Smith [75] uses Calculus of Variations to evaluate necessary conditions and results in an exact analytical solution to this problem for the state

$$y^*(t) = \frac{2e^{3t} + e^3}{e^{3t/2}\left(2 + e^3\right)}, t \in [0,1] \tag{54}$$

the control

$$u^*(t) = \frac{2\left(e^{3t} - e^3\right)}{e^{3t/2}\left(2 + e^3\right)}, t \in [0,1] \tag{55}$$

and the co-state

$$cs^*(t) = -\frac{2\left(e^{3t} - e^3\right)}{e^{3t/2}\left(2 + e^3\right)}, t \in [0, 1] \tag{56}$$

As part of AERO899, Suplisson, Smith and Masternak used MATLAB's fmincon[1] optimization function to obtain a numerical solution to this optimal control problem using pseudospectral methods. This 'generic' pseudospectral solver uses Legendre spectral algorithms from Li-Lian [71] to compute the Legendre-Gauss-Radau (LGR) quadrature nodes and weights used in numerical integration to evaluate the problem's objective function. These same algorithms are also used to compute the first-order LGR differentiation matrix used in numerical differentiation to evaluate the problem's constraint function. Fmincon uses these constraint and objective functions, along with the given boundary conditions, to compute a control to minimize the objective function.

GPOPS was also used to obtain a numerical solution to this optimal control problem using pseudospectral methods. While the 'generic' solver above had to reference LGR-related algorithms, GPOPS has similar algorithms included within GPOPS. As with the 'generic', objective and constraint functions, as well as boundary conditions, were included in the MATLAB code.

The numerical solution from each of these techniques was plotted against the exact analytical solution over the same time interval. As shown in Figure 118, both numerical approximations show good correlation to the exact analytical solution. Note that the collocation points differ for each of the numerical solutions[2], but they used the same number of collocation points to compute the solution shown.

Table 7 shows the relative error as defined by Golub [34]. Relative error, η, is a measure of the difference between an exact solution ($\mathbf{y_{exact}}$) and an approximate

[1] finds the minimum constrained non-linear multi-variable function

[2] the 'generic' pseudospectral solver uses only one interval with LGR node distribution while the GPOPS solution uses several intervals with LGR node distribution

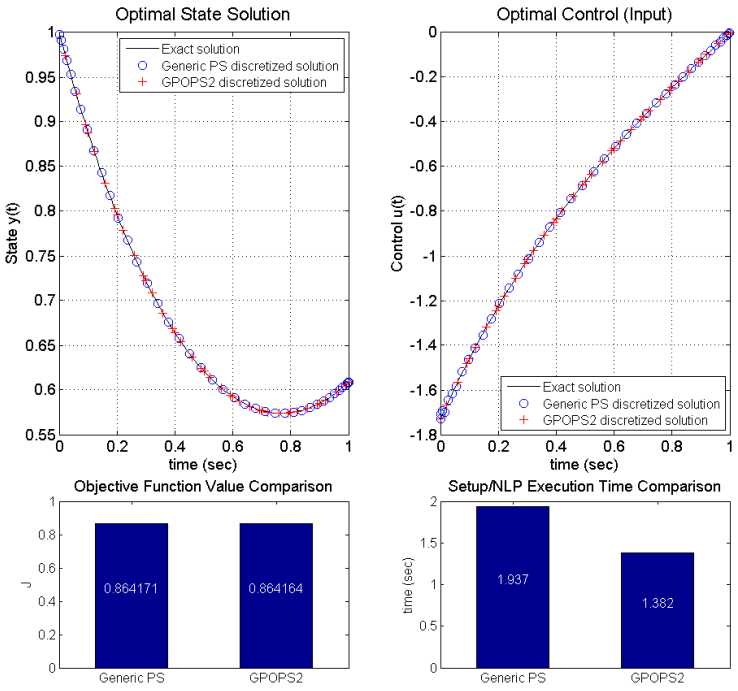


Figure 118. Solution to Linear-Quadratic Optimal Control Problem, comparing exact to discretized approximate solutions

solution ($\mathbf{y_{approx}}$), normalized with the exact solution.

$$\eta = \frac{\|\mathbf{y_{exact}} - \mathbf{y_{approx}}\|_2}{\|\mathbf{y_{exact}}\|_2} \tag{57}$$

	Generic Pseudospectral Error	GPOPS Error
State	9.77E-05	1.47E-09
Control	6.60E-02	3.32E-07

Table 7. Numerical Approximation Error for Linear-Quadratic Optimal Control Problem

For this problem, the GPOPS solution was significantly closer to the exact analytical solution than the Generic pseudospectral (PS) solution. This is primarily a result from GPOPS using intervals to break up the problem interval into smaller mesh intervals and approximate the solution in each interval using a high order polynomial. Even though GPOPS and the Generic PS methods used the same number of collocation points for this problem, GPOPS' use of mesh intervals resulted in a significantly more accurate solution.

Appendix C. Scenario Results Detailed Charts

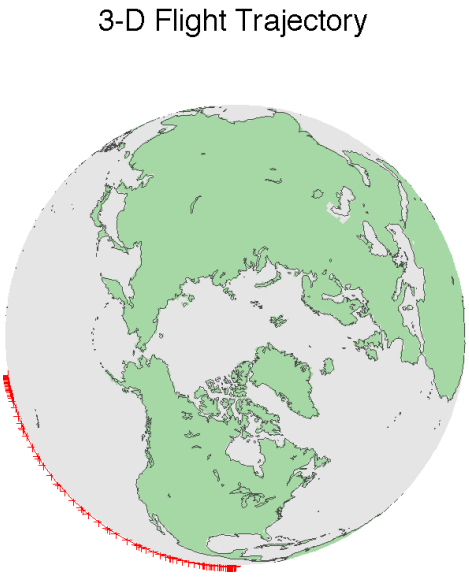
This appendix contains the complete set of results presented and discussed in Chapter V.

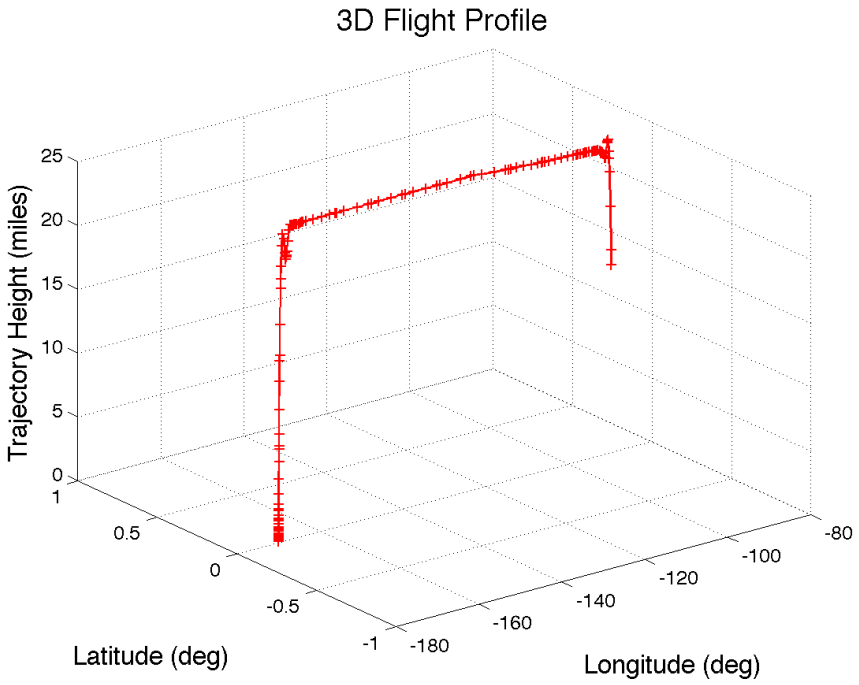
In this appendix, the section name describes the scenario results presented in that section. Each section has the set of charts listed below. Some of them are already shown in Chapter V but are included here to present a complete set of scenario results.

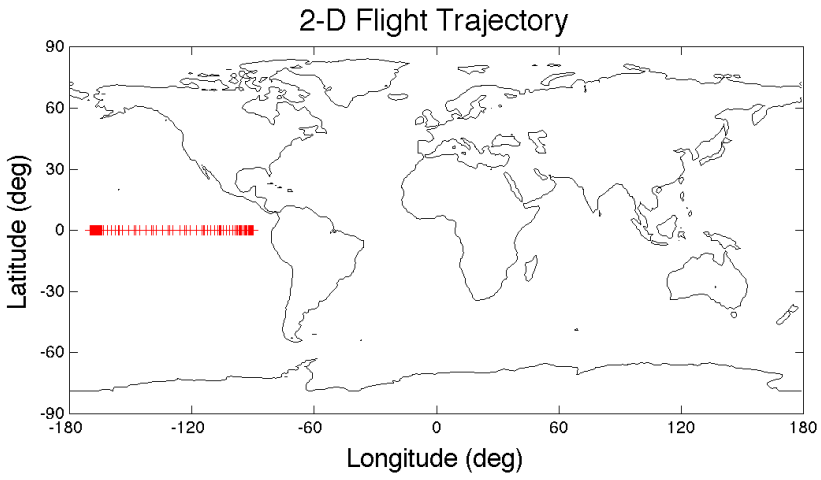
- 3-D Flight Trajectory
- 3-D Flight Profile
- 2-D Flight Trajectory
- 2-D Flight Profile
- No-Fly Zone Distances (for scenarios with this path constraint)
- Sensor target elevation and slant range (for scenarios with these path constraints)
- Height Velocity, and Mass States vs. Time
- Longitude, Latitude, Flight Path Angle and Heading Angle States vs. Time
- Angle of Attack, Bank Angle, and Propellant Mass Flow Rate Controls vs. Time (for scenarios that use angular controls)
- Angle of Attack and Bank Angle States, and Propellant Mass Flow Rate Controls vs. Time (for scenarios that use angular rate controls)
- Angle of Attack and Bank Angle Rate Controls vs. Time (for scenarios that use angular rate controls)
- Path Constraints and G-forces vs. Time
- Lookup Table Values vs. Time
- Stability and Vehicle Frame Forces vs. Time
- Heating Rate Profile vs. Time
- Heating Load Profile vs. Time
- Temperature Profile vs. Time

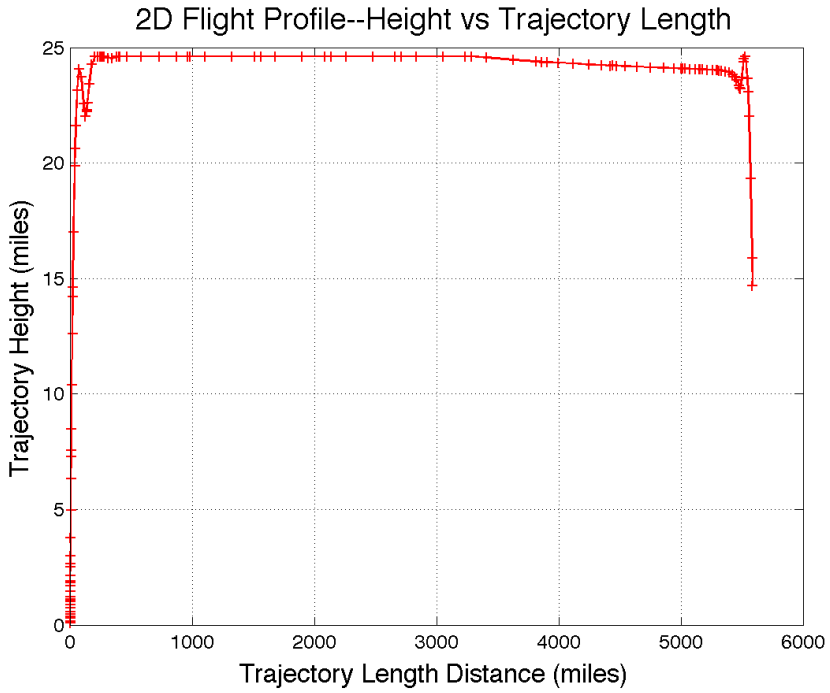
- Mesh Tolerance and Maximum Relative Error vs. Mesh Iteration
- Mesh Interval History
- Mesh Interval and Collocation Point History

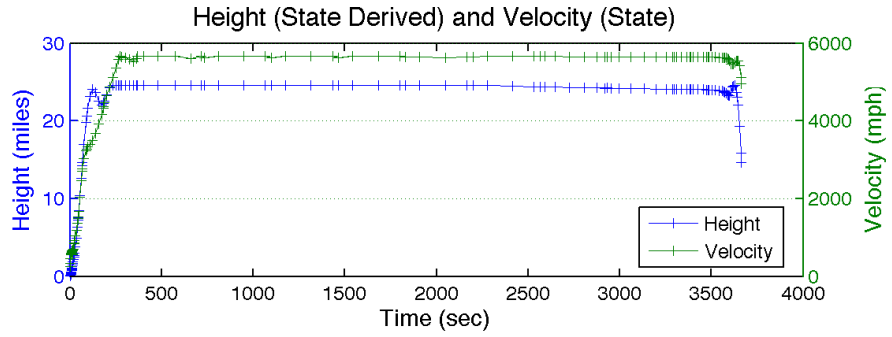
C.1 Maximum Range; Climb and Cruise

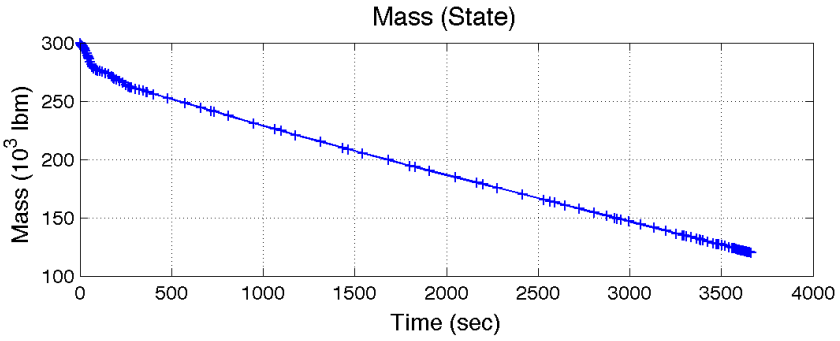


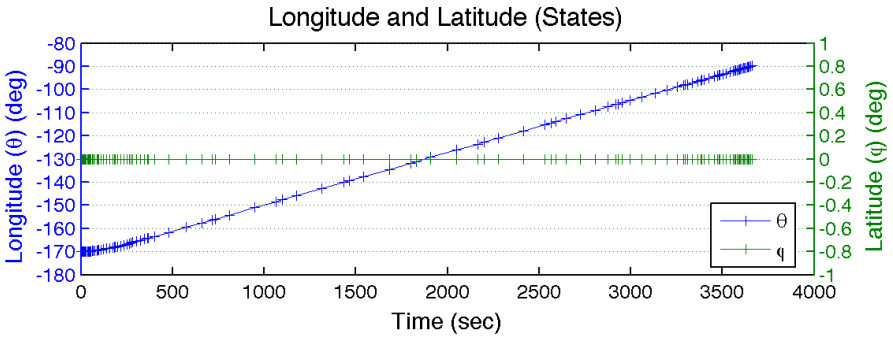

3D Flight Profile
Trajectory Height (miles)
25
20
15
10
5
0
1
0.5
0
-0.5
-1
Latitude (deg)
-180
-160
-140
-120
-100
-80
Longitude (deg)

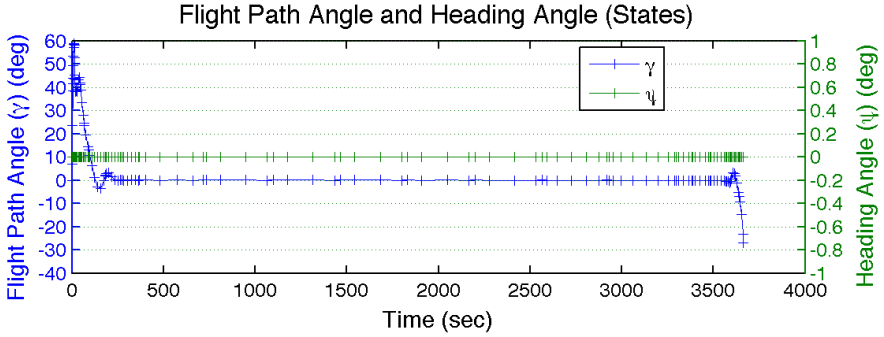

2-D Flight Trajectory
Latitude (deg)
90
60
30
0
-30
-60
-90
-180
-120
-60
0
60
120
180
Longitude (deg)

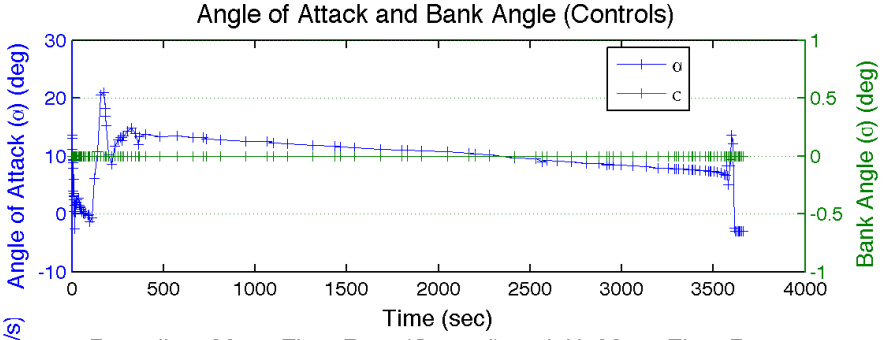

2D Flight Profile--Height vs Trajectory Length
Trajectory Height (miles)
Trajectory Length Distance (miles)
0
5
10
15
20
25
0
1000
2000
3000
4000
5000
6000

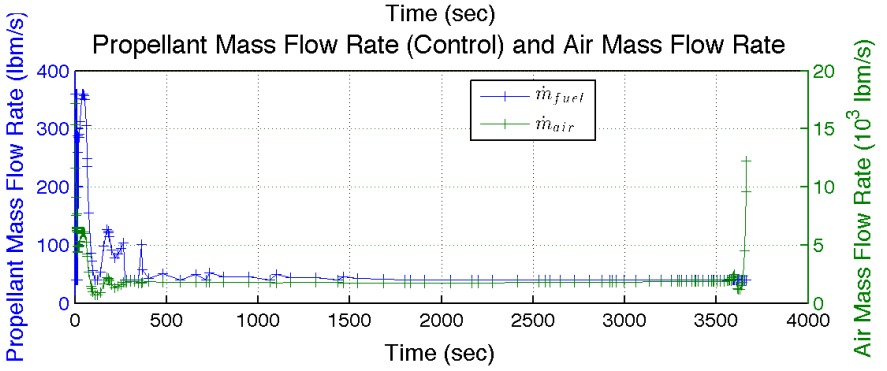

Height (State Derived) and Velocity (State)
Height (miles)
Velocity (mph)
Time (sec)
Height
Velocity
0
10
20
30
0
2000
4000
6000
0
500
1000
1500
2000
2500
3000
3500
4000

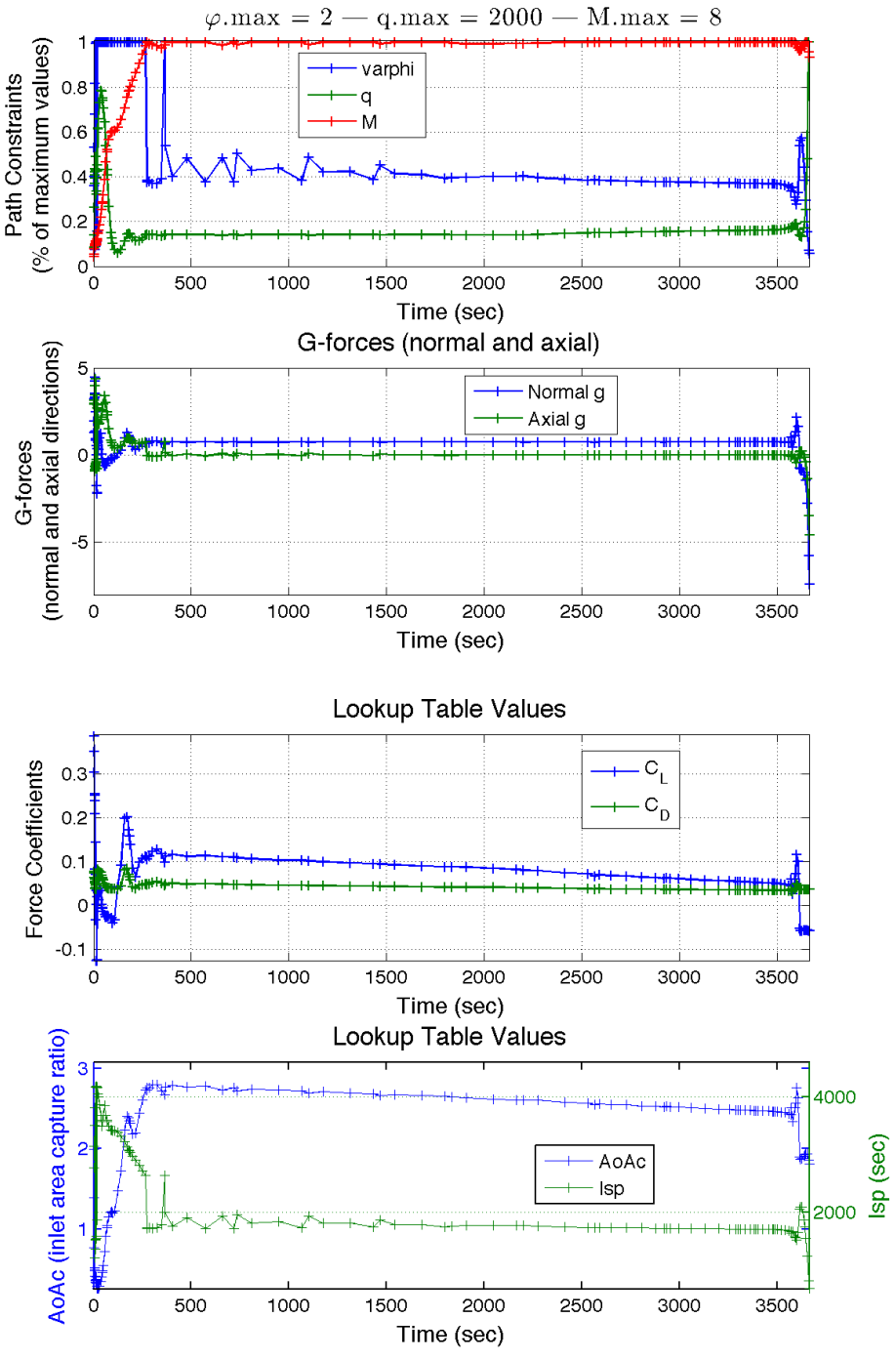

Mass (State)
Mass (10^3 lbm)
Time (sec)
100
150
200
250
300
0
500
1000
1500
2000
2500
3000
3500
4000

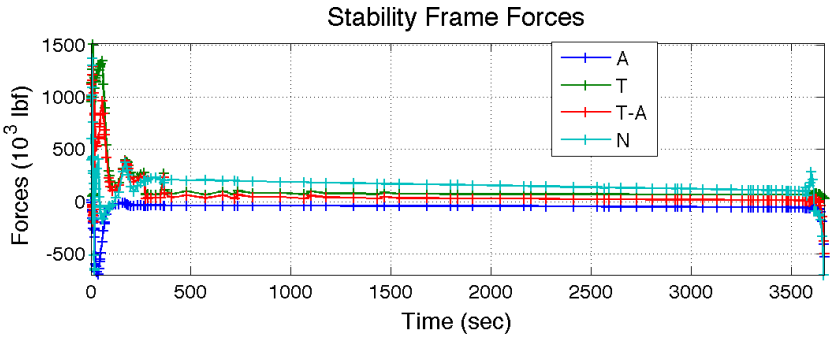

Longitude and Latitude (States)
Longitude (θ) (deg)
Latitude (φ) (deg)
Time (sec)
θ
φ

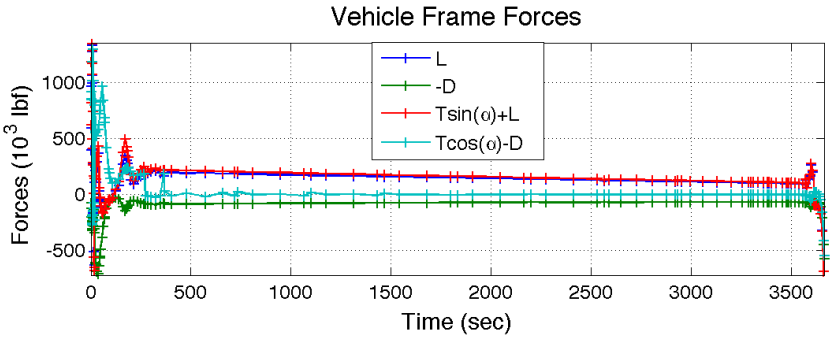

Flight Path Angle and Heading Angle (States)
Flight Path Angle (γ) (deg)
Heading Angle (ψ) (deg)
Time (sec)
γ
ψ

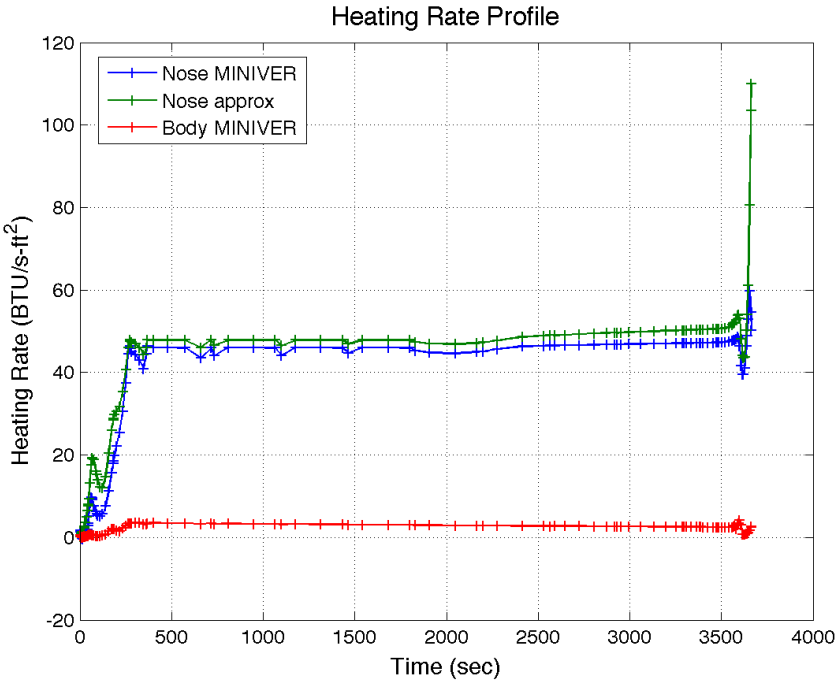

Angle of Attack and Bank Angle (Controls)
Angle of Attack (α) (deg)
Bank Angle (σ) (deg)
Time (sec)
α
σ

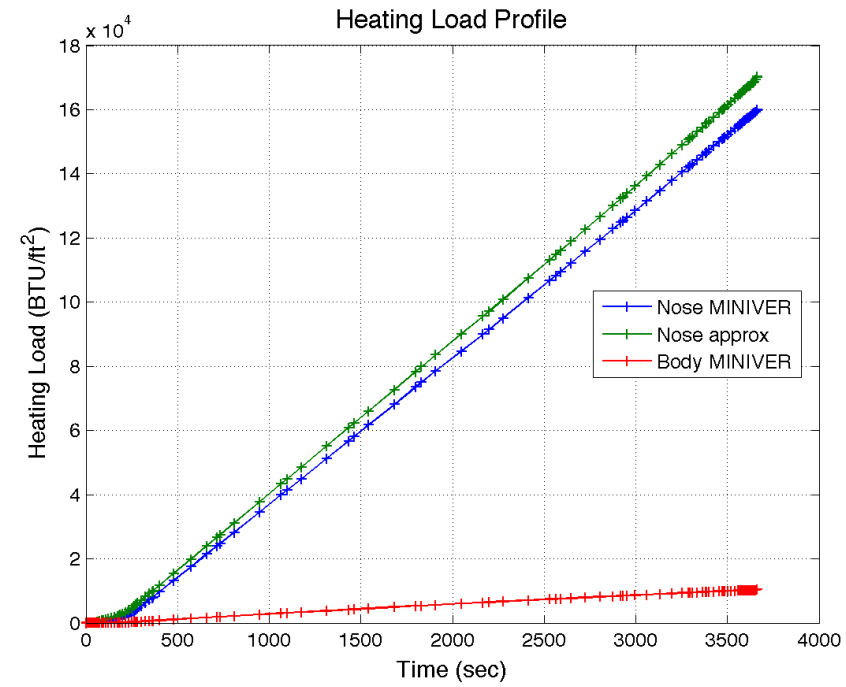

Propellant Mass Flow Rate (Control) and Air Mass Flow Rate
Propellant Mass Flow Rate (lbm/s)
Air Mass Flow Rate (10^3 lbm/s)
Time (sec)
$\dot{m}_{fuel}$
$\dot{m}_{air}$

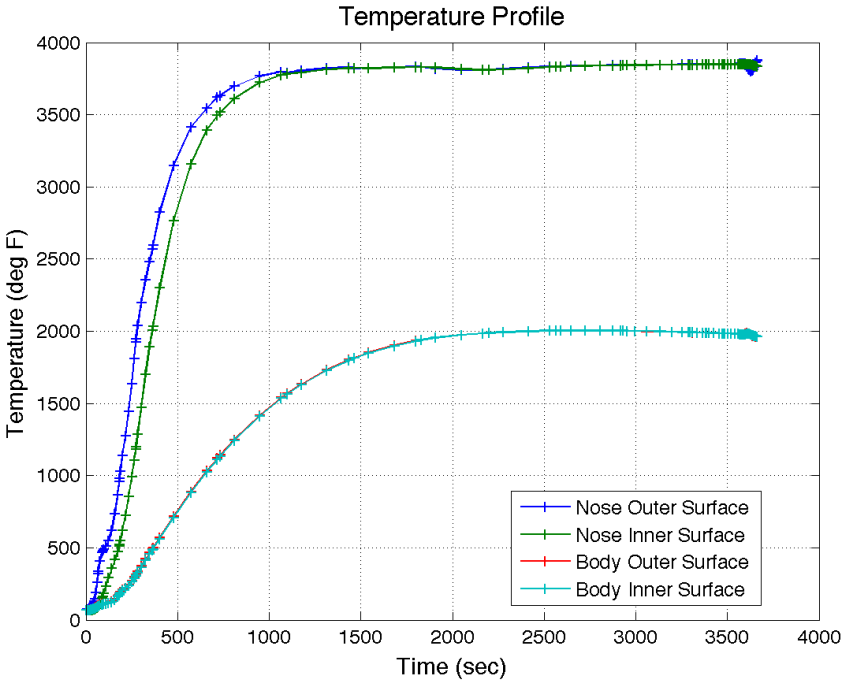

φ.max = 2 — q.max = 2000 — M.max = 8
Path Constraints
(% of maximum values)
varphi
q
M
Time (sec)
G-forces (normal and axial)
G-forces
(normal and axial directions)
Normal g
Axial g
Time (sec)
Lookup Table Values
Force Coefficients
C_L
C_D
Time (sec)
Lookup Table Values
AoAc (inlet area capture ratio)
Isp (sec)
AoAc
Isp
Time (sec)

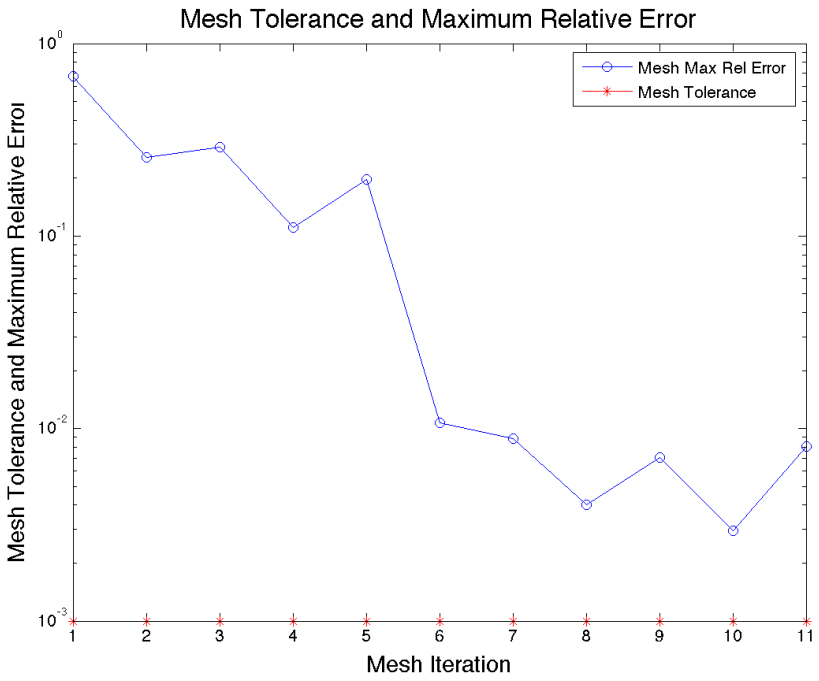

Stability Frame Forces
Forces (10^3 lbf)
Time (sec)
A
T
T-A
N

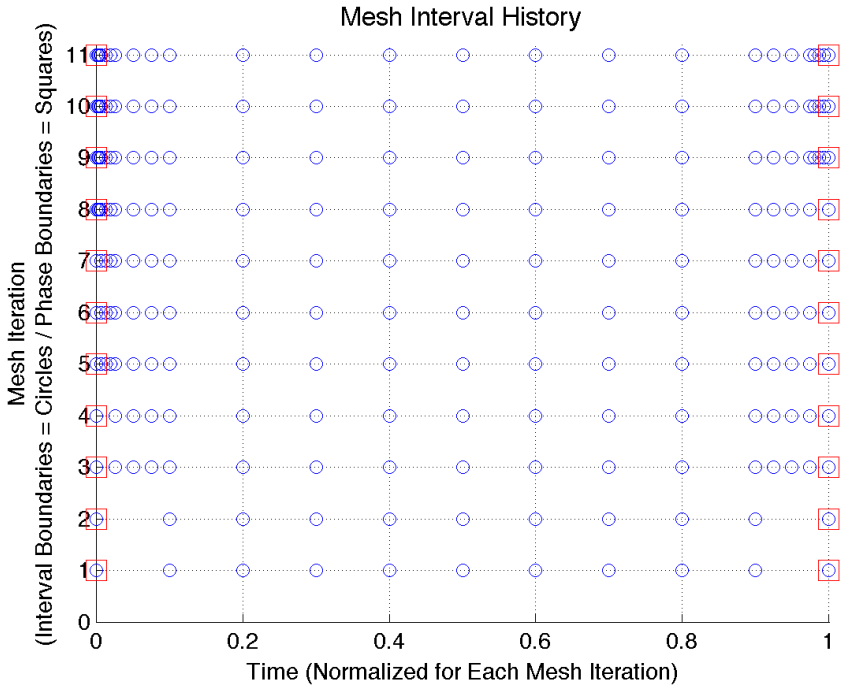

Vehicle Frame Forces
Forces (10^3 lbf)
Time (sec)
L
-D
Tsin(α)+L
Tcos(α)-D

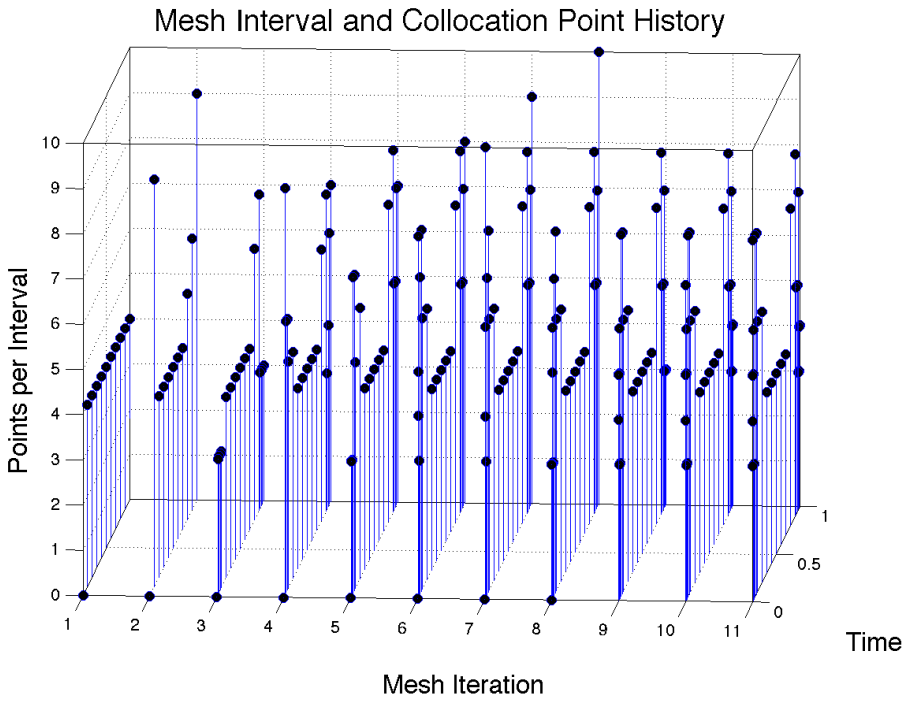

Heating Rate Profile
Heating Rate (BTU/s-ft^2)
Time (sec)
Nose MINIVER
Nose approx
Body MINIVER


Heating Load Profile
x 10^4
Heating Load (BTU/ft^2)
Time (sec)
Nose MINIVER
Nose approx
Body MINIVER


Temperature Profile
Temperature (deg F)
Time (sec)
Nose Outer Surface
Nose Inner Surface
Body Outer Surface
Body Inner Surface

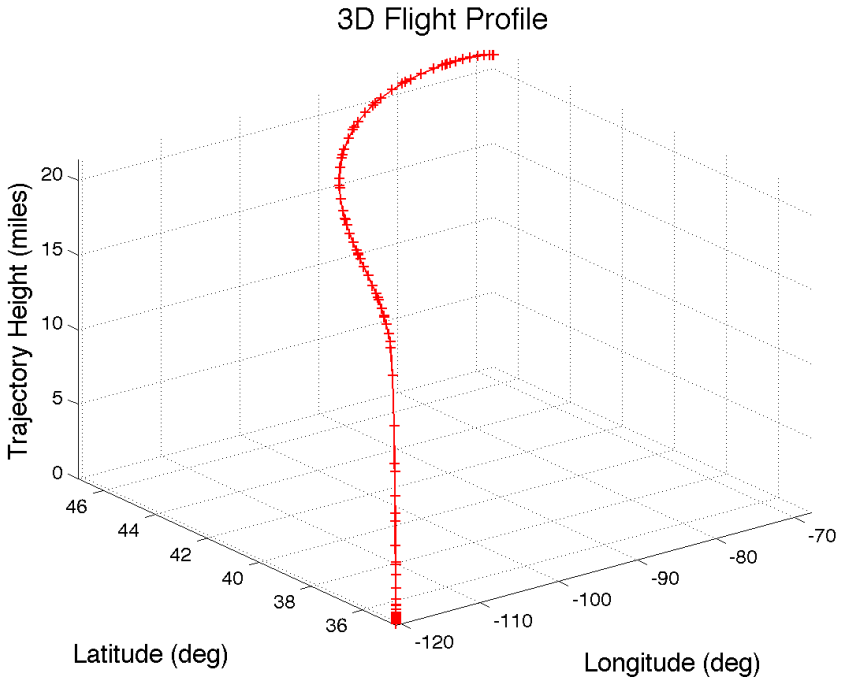

Mesh Tolerance and Maximum Relative Error
Mesh Max Rel Error
Mesh Tolerance
Mesh Tolerance and Maximum Relative Error
Mesh Iteration
10^0
10^-1
10^-2
10^-3
1
2
3
4
5
6
7
8
9
10
11

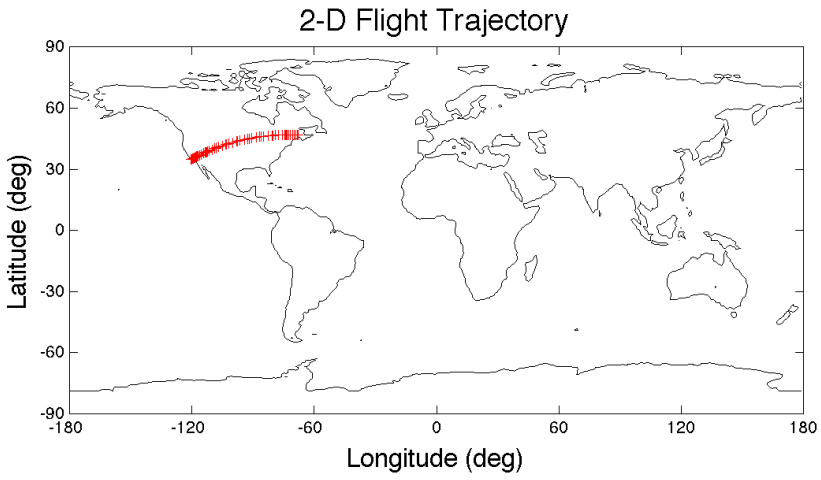

Mesh Interval History
(Interval Boundaries = Circles / Phase Boundaries = Squares)
Mesh Iteration
Time (Normalized for Each Mesh Iteration)
0
1
2
3
4
5
6
7
8
9
10
11
0
0.2
0.4
0.6
0.8
1

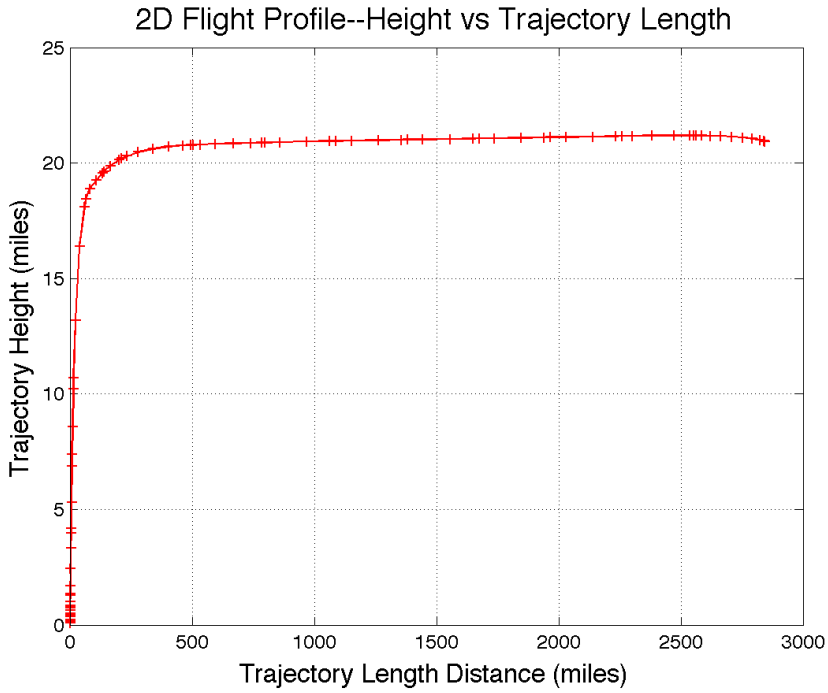


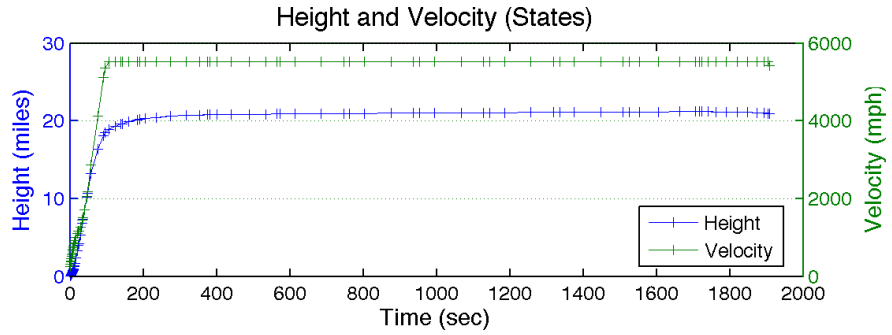
C.2 Minimum Time; Climb and Cruise

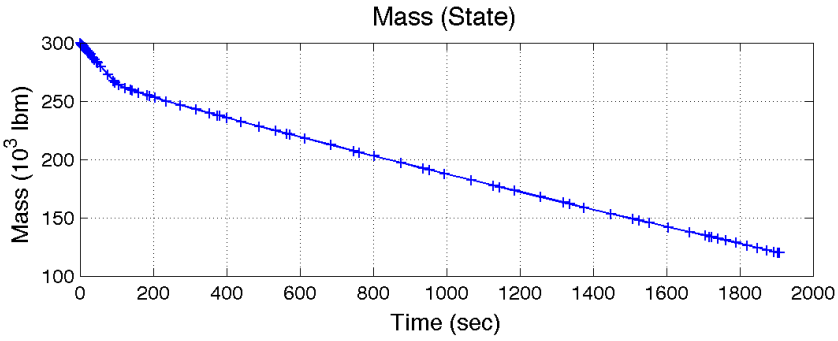


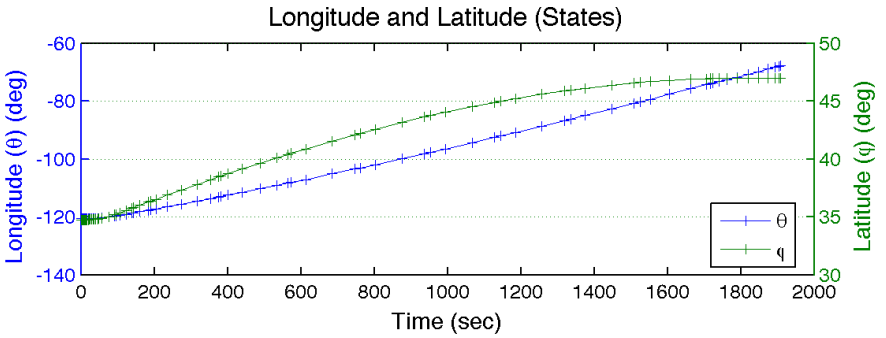

3D Flight Profile
Trajectory Height (miles)
20
15
10
5
0
46
44
42
40
38
36
Latitude (deg)
-70
-80
-90
-100
-110
-120
Longitude (deg)

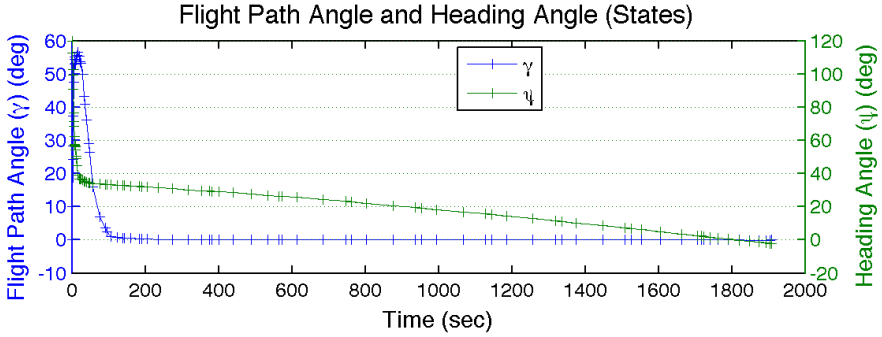

2-D Flight Trajectory
Latitude (deg)
90
60
30
0
-30
-60
-90
-180
-120
-60
0
60
120
180
Longitude (deg)

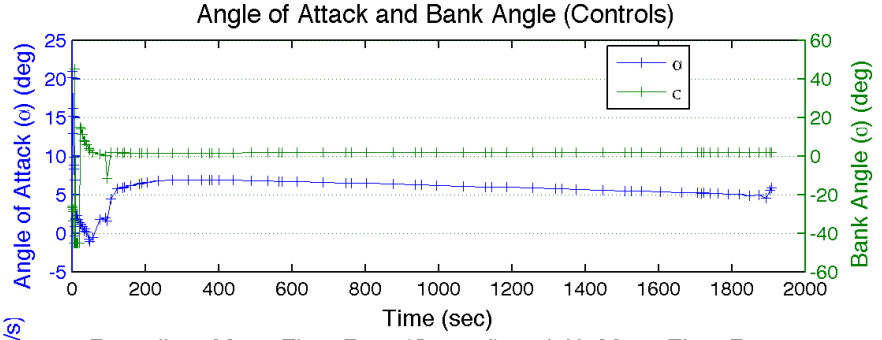

2D Flight Profile--Height vs Trajectory Length
Trajectory Height (miles)
Trajectory Length Distance (miles)
0
5
10
15
20
25
0
500
1000
1500
2000
2500
3000

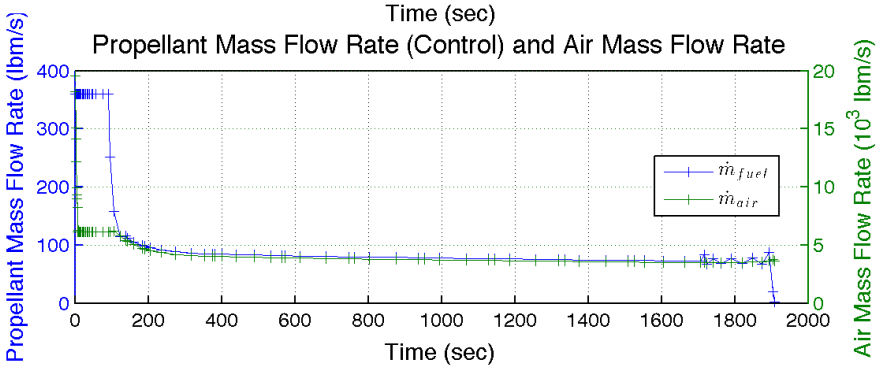

Height and Velocity (States)
Height (miles)
Velocity (mph)
Time (sec)
Height
Velocity
0
10
20
30
0
2000
4000
6000
0
200
400
600
800
1000
1200
1400
1600
1800
2000

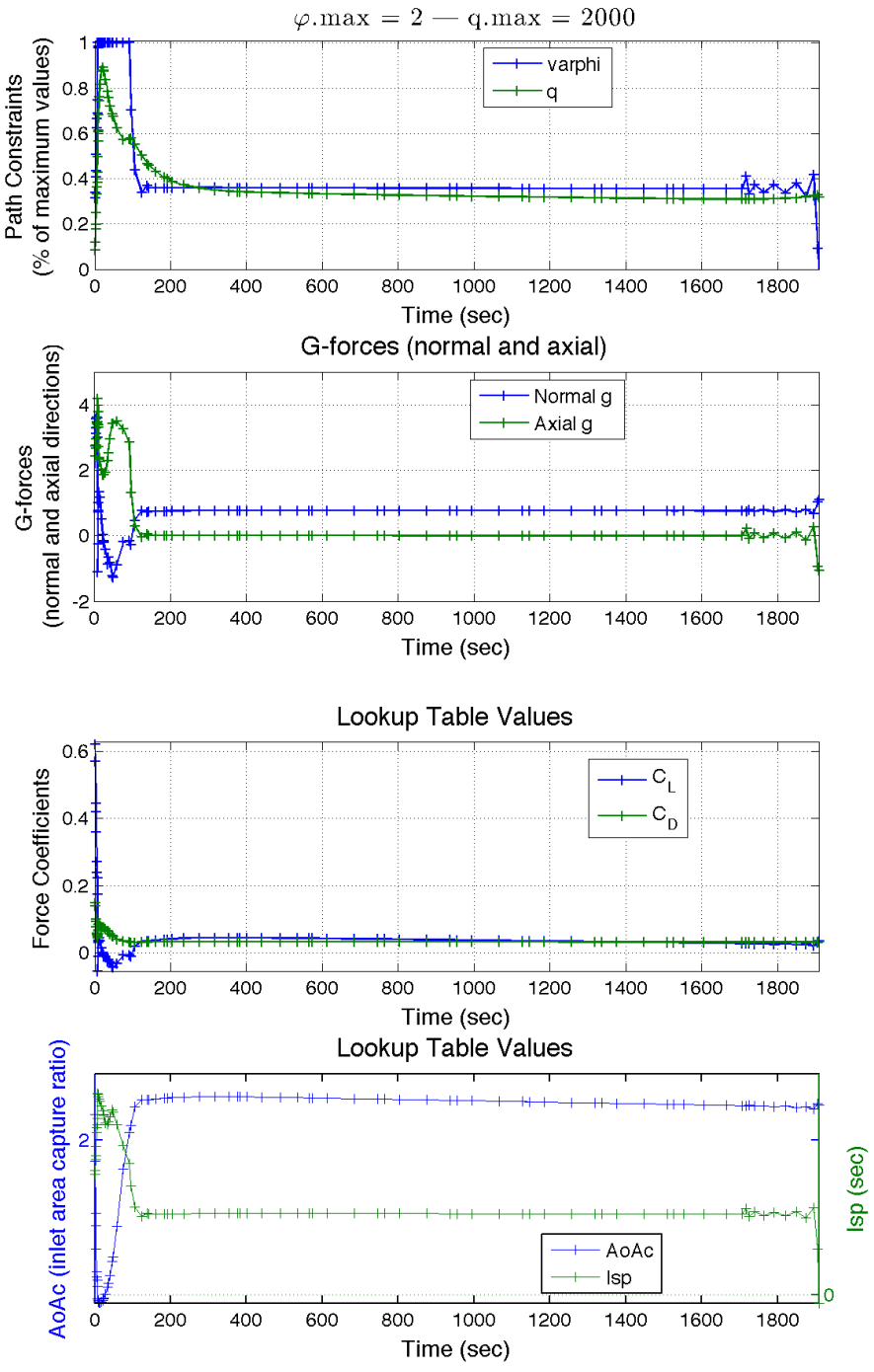

Mass (State)
Mass (10^3 lbm)
Time (sec)
100
150
200
250
300
0
200
400
600
800
1000
1200
1400
1600
1800
2000

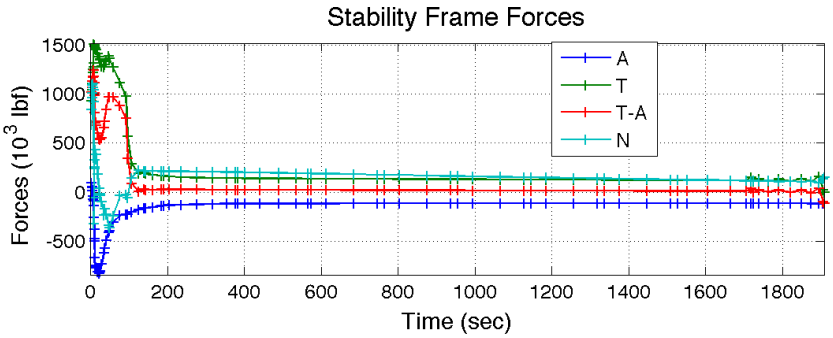

Longitude and Latitude (States)
Longitude (θ) (deg)
Latitude (φ) (deg)
Time (sec)
θ
φ

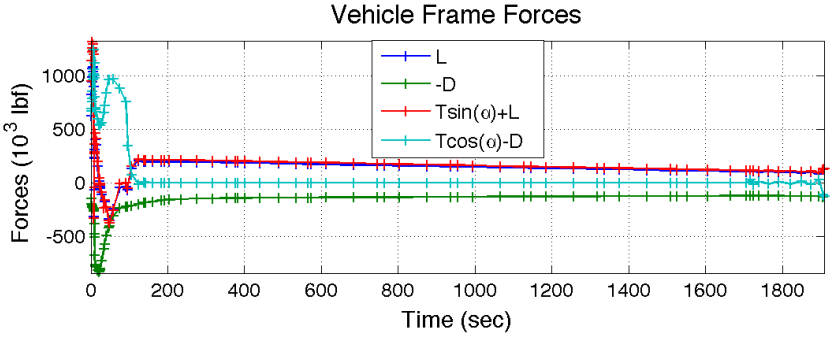

Flight Path Angle and Heading Angle (States)
Flight Path Angle (γ) (deg)
Heading Angle (ψ) (deg)
Time (sec)
γ
ψ

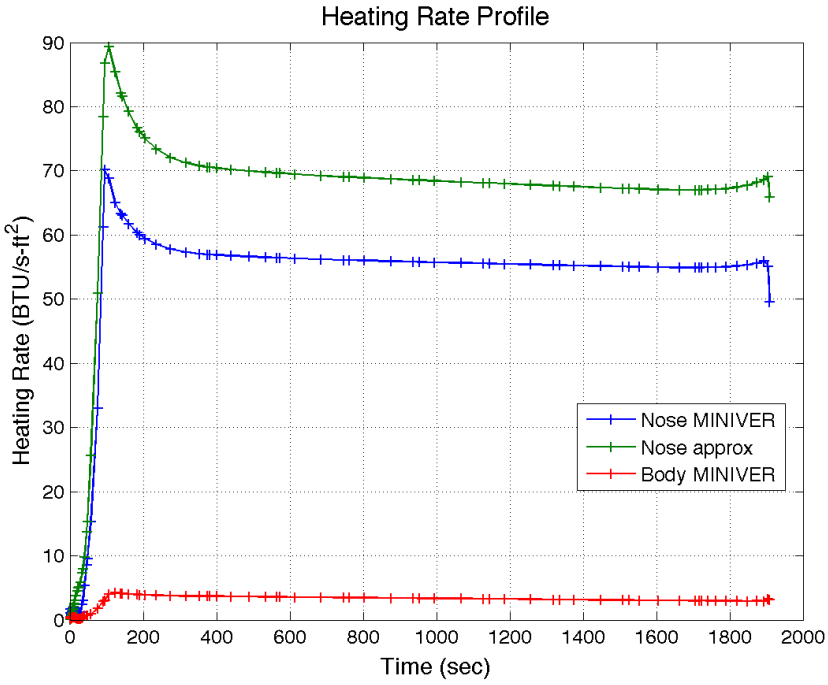

Angle of Attack and Bank Angle (Controls)
Angle of Attack (α) (deg)
Bank Angle (σ) (deg)
Time (sec)
α
σ

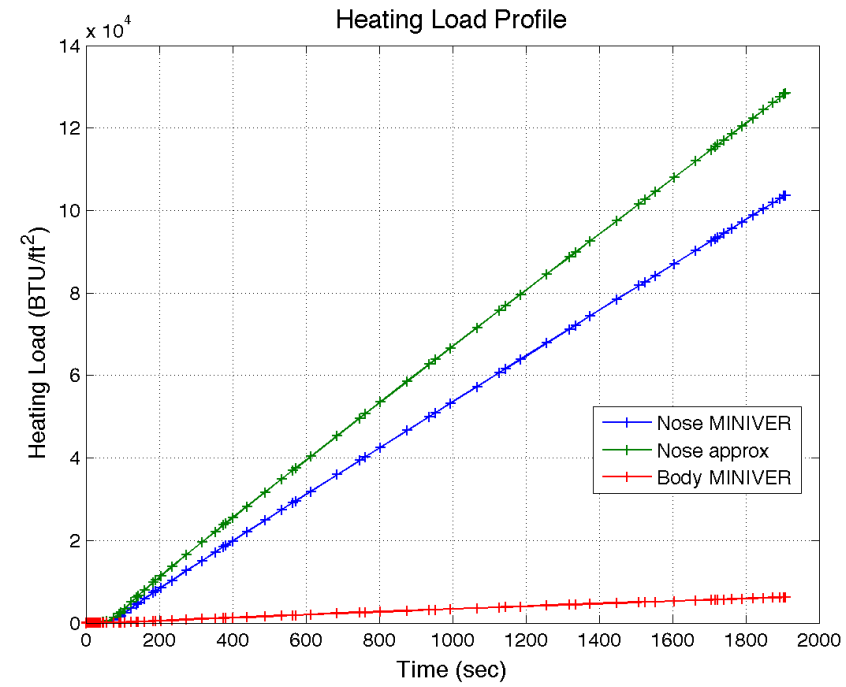

Propellant Mass Flow Rate (Control) and Air Mass Flow Rate
Propellant Mass Flow Rate (lbm/s)
Air Mass Flow Rate (10^3 lbm/s)
Time (sec)
$\dot{m}_{fuel}$
$\dot{m}_{air}$

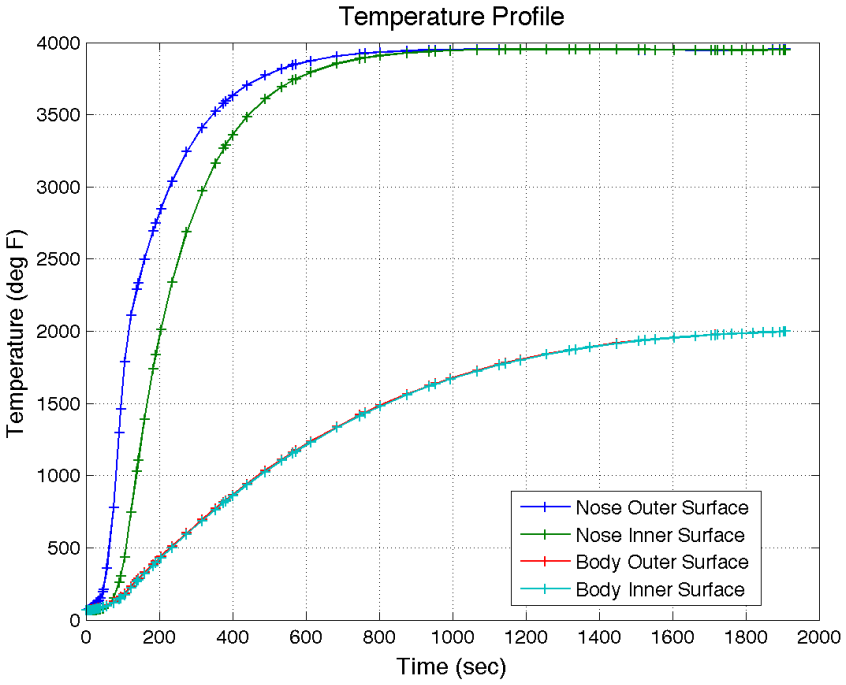

φ.max = 2 — q.max = 2000
Path Constraints
(% of maximum values)
varphi
q
Time (sec)
G-forces (normal and axial)
G-forces
(normal and axial directions)
Normal g
Axial g
Time (sec)
Lookup Table Values
Force Coefficients
C_L
C_D
Time (sec)
Lookup Table Values
AoAc (inlet area capture ratio)
Isp (sec)
AoAc
Isp
Time (sec)

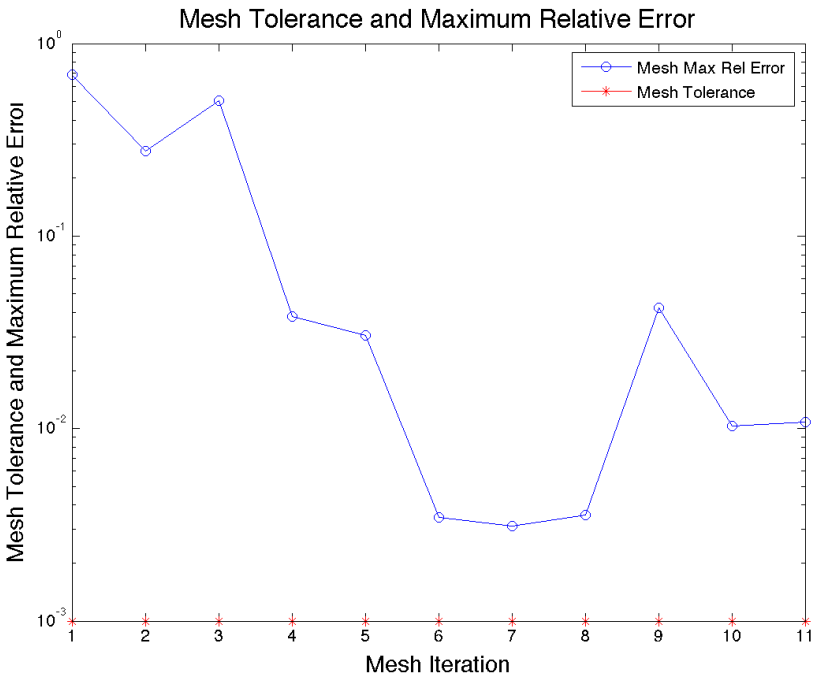

Stability Frame Forces
Forces (10^3 lbf)
Time (sec)
A
T
T-A
N

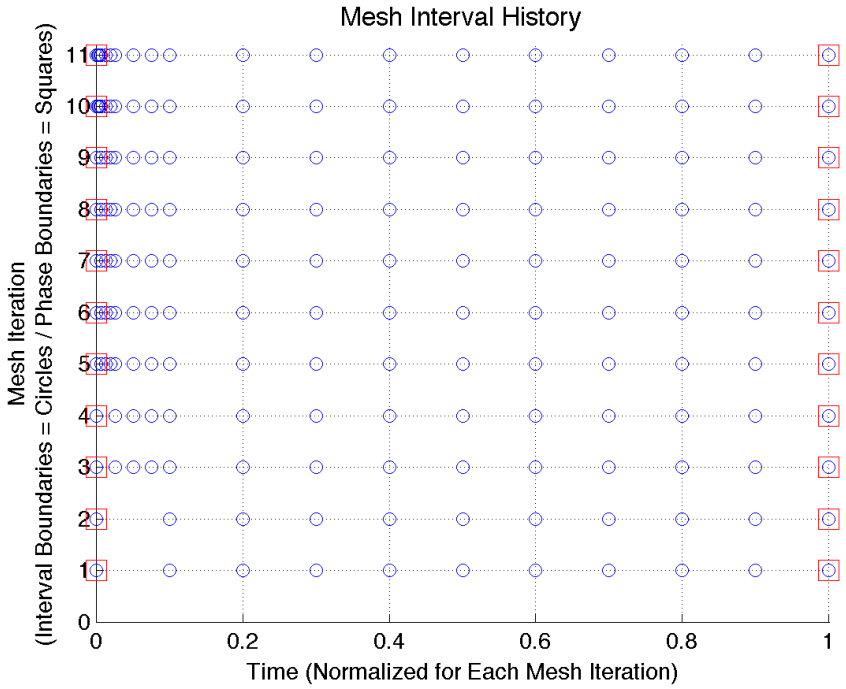

Vehicle Frame Forces
Forces (10^3 lbf)
Time (sec)
L
-D
Tsin(α)+L
Tcos(α)-D

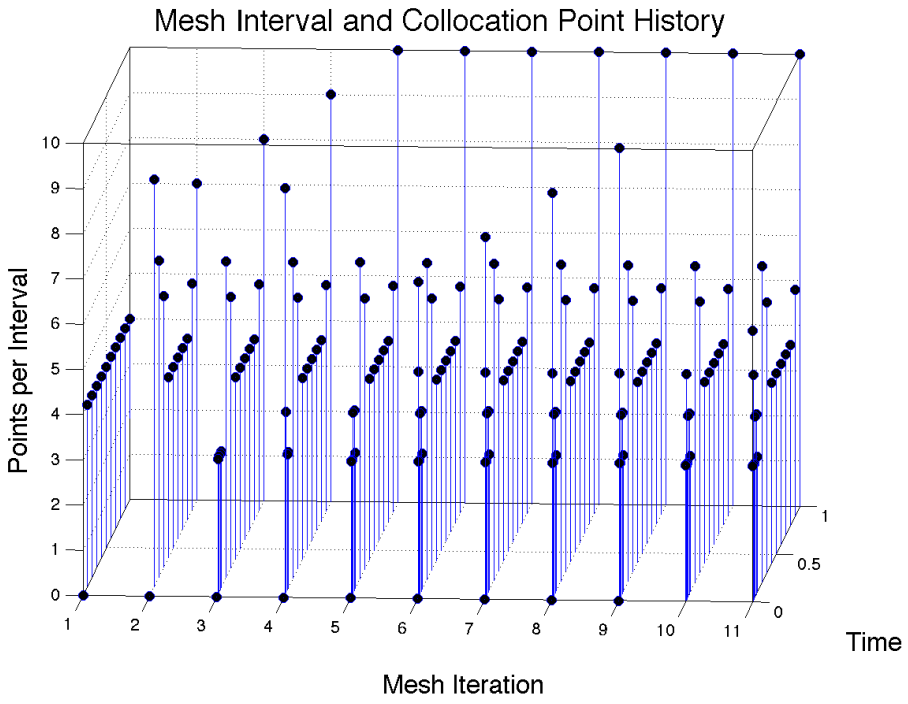

Heating Rate Profile
Heating Rate (BTU/s-ft^2)
Time (sec)
Nose MINIVER
Nose approx
Body MINIVER


Heating Load Profile
x 10^4
Heating Load (BTU/ft^2)
Time (sec)
Nose MINIVER
Nose approx
Body MINIVER


Temperature Profile
Temperature (deg F)
Time (sec)
Nose Outer Surface
Nose Inner Surface
Body Outer Surface
Body Inner Surface

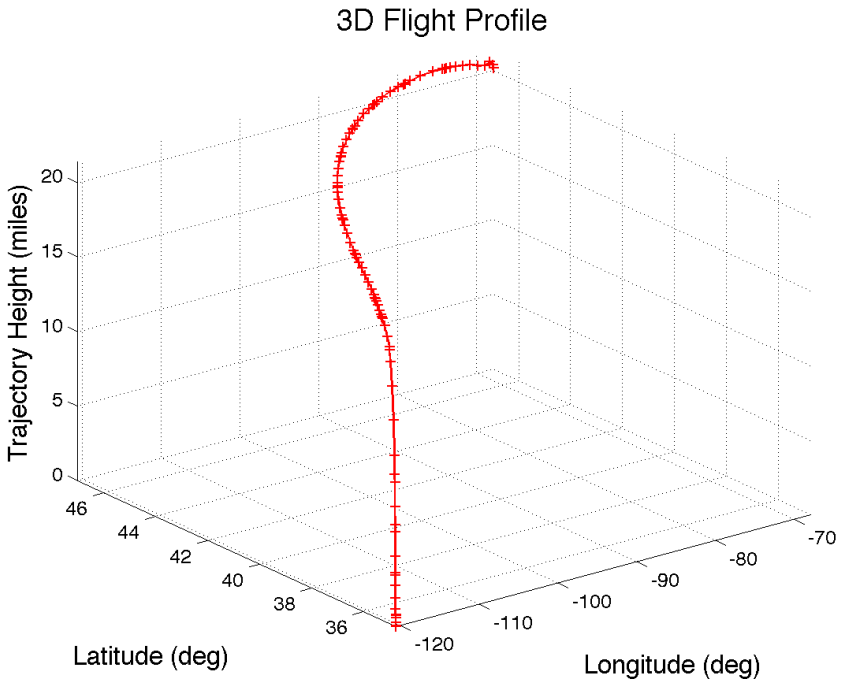

Mesh Tolerance and Maximum Relative Error
Mesh Max Rel Error
Mesh Tolerance
Mesh Tolerance and Maximum Relative Error
Mesh Iteration

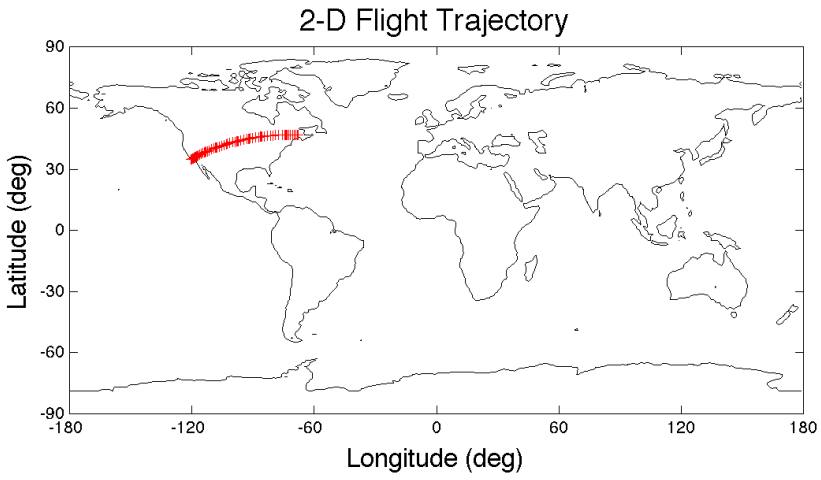

Mesh Interval History
(Interval Boundaries = Circles / Phase Boundaries = Squares)
Mesh Iteration
Time (Normalized for Each Mesh Iteration)

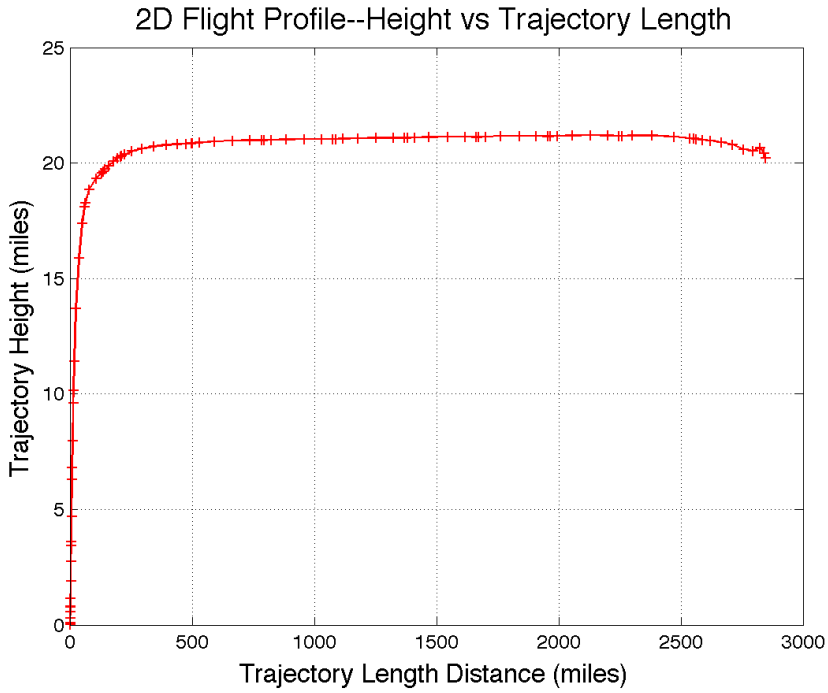


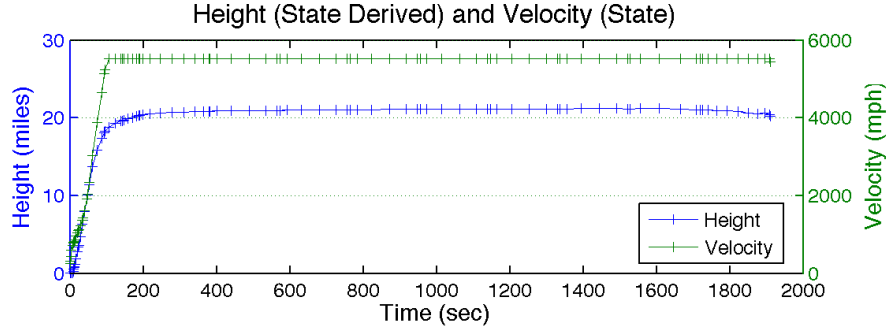
C.3 Minimum Time with Control Penalty; Climb and Cruise

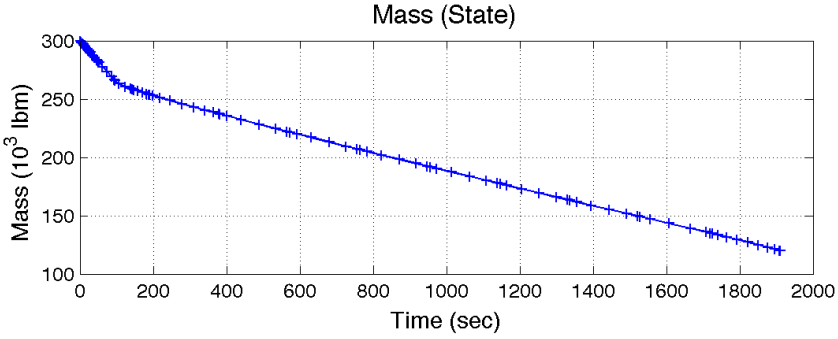


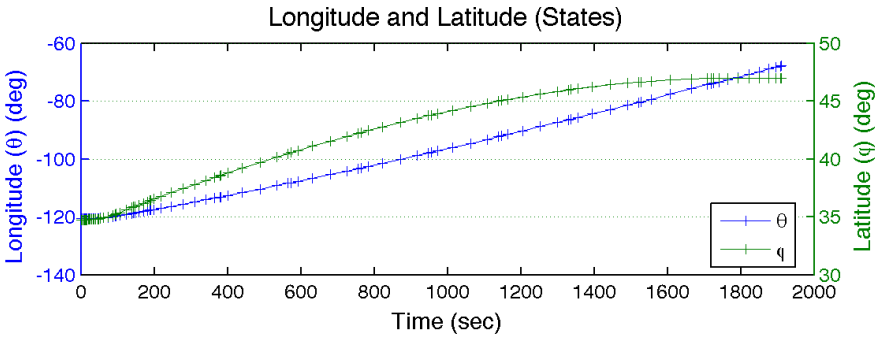

3D Flight Profile
Trajectory Height (miles)
20
15
10
5
0
46
44
42
40
38
36
Latitude (deg)
-70
-80
-90
-100
-110
-120
Longitude (deg)

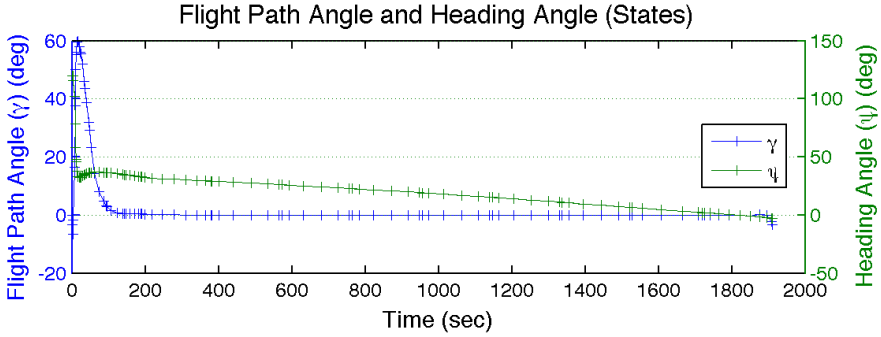

2-D Flight Trajectory
Latitude (deg)
90
60
30
0
-30
-60
-90
-180
-120
-60
0
60
120
180
Longitude (deg)

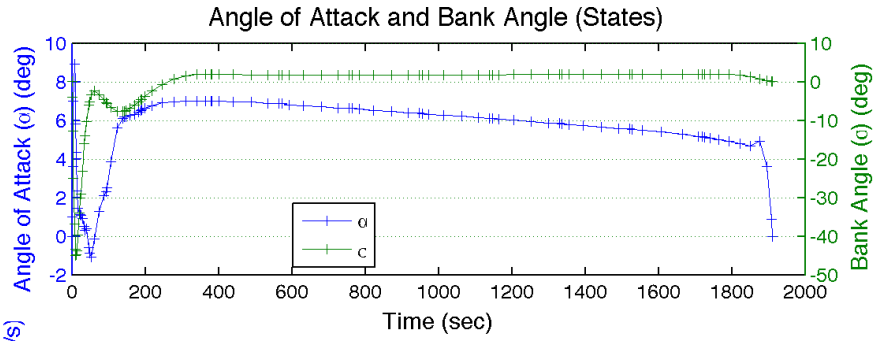

2D Flight Profile--Height vs Trajectory Length
Trajectory Height (miles)
Trajectory Length Distance (miles)
0
5
10
15
20
25
0
500
1000
1500
2000
2500
3000

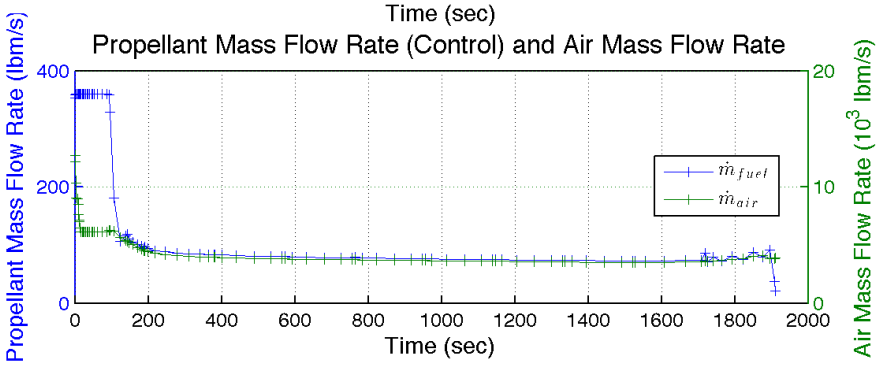

Height (State Derived) and Velocity (State)
Height (miles)
Velocity (mph)
Time (sec)
Height
Velocity
0
10
20
30
0
2000
4000
6000
0
200
400
600
800
1000
1200
1400
1600
1800
2000

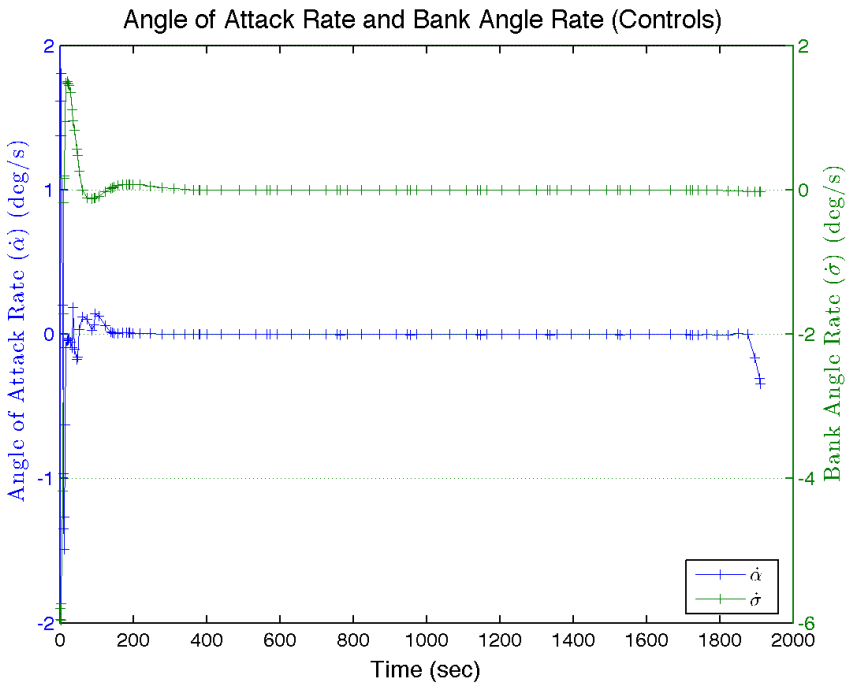

Mass (State)
Mass (10^3 lbm)
Time (sec)
100
150
200
250
300
0
200
400
600
800
1000
1200
1400
1600
1800
2000

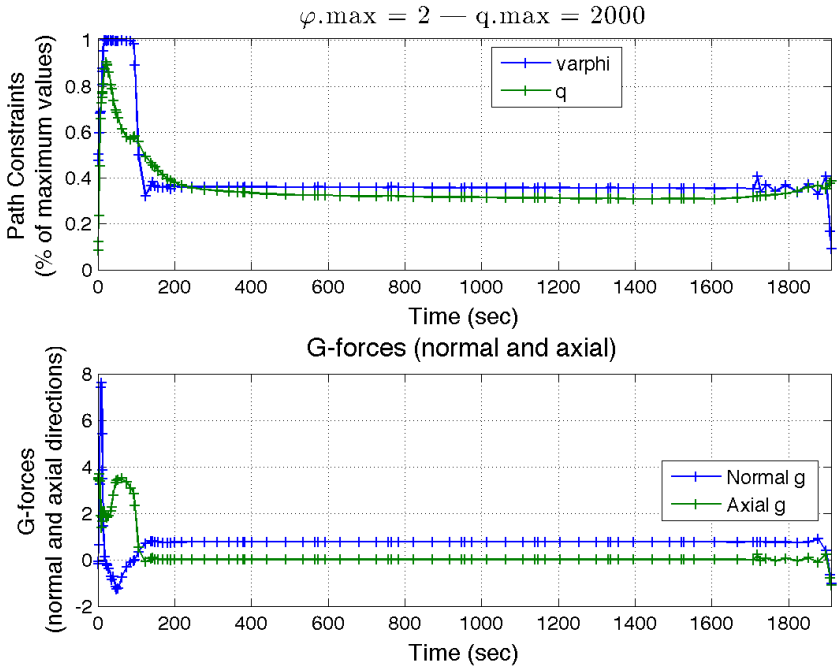

Longitude and Latitude (States)
Longitude (θ) (deg)
Latitude (φ) (deg)
Time (sec)
θ
φ

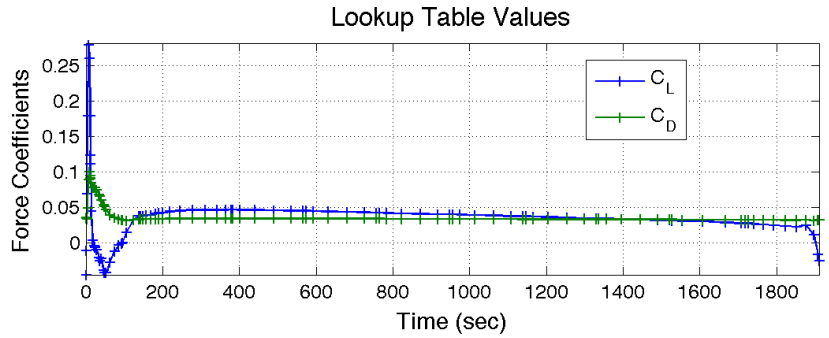

Flight Path Angle and Heading Angle (States)
Flight Path Angle (γ) (deg)
Heading Angle (ψ) (deg)
Time (sec)
γ
ψ

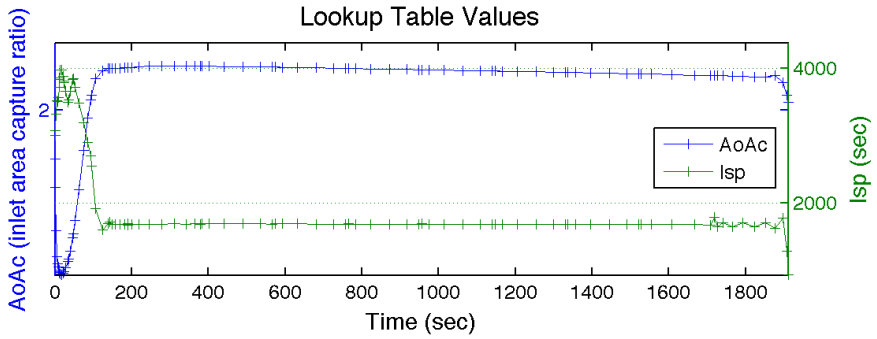

Angle of Attack and Bank Angle (States)
Angle of Attack (α) (deg)
Bank Angle (σ) (deg)
Time (sec)
α
σ

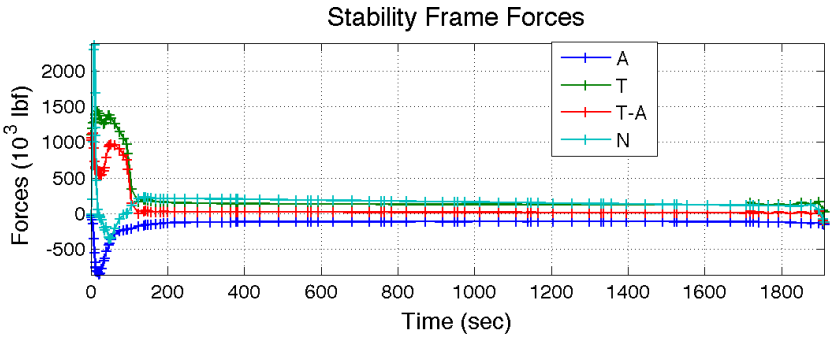

Propellant Mass Flow Rate (Control) and Air Mass Flow Rate
Propellant Mass Flow Rate (lbm/s)
Air Mass Flow Rate (10^3 lbm/s)
Time (sec)
$\dot{m}_{fuel}$
$\dot{m}_{air}$

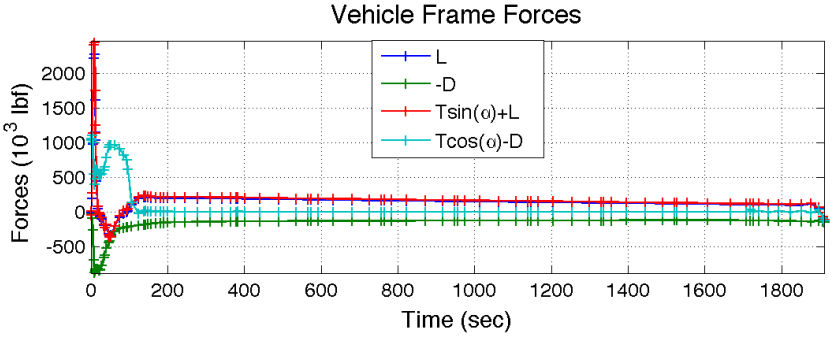

Angle of Attack Rate and Bank Angle Rate (Controls)
Angle of Attack Rate ($\dot{\alpha}$) (deg/s)
Bank Angle Rate ($\dot{\sigma}$) (deg/s)
Time (sec)
$\dot{\alpha}$
$\dot{\sigma}$

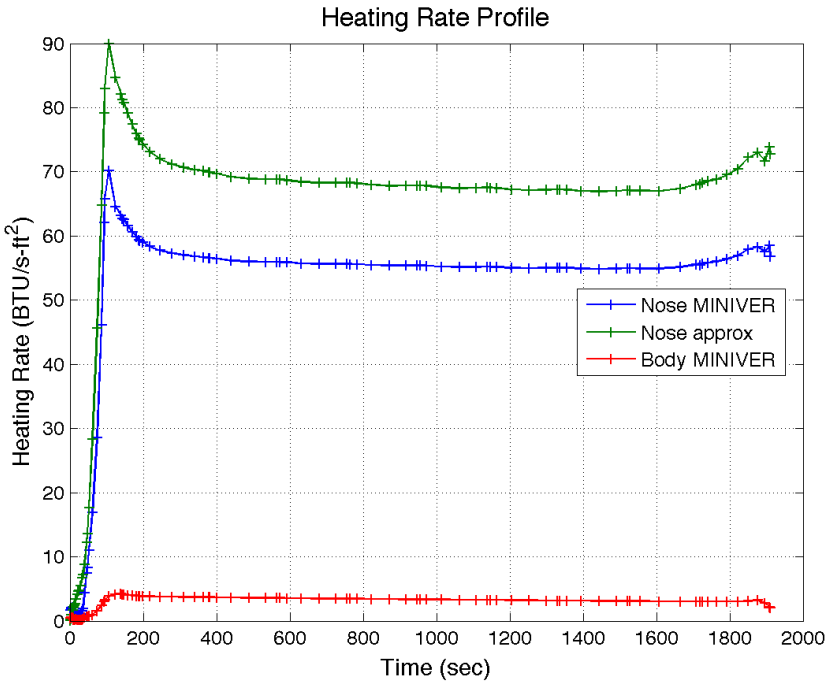

φ.max = 2 — q.max = 2000
Path Constraints
(% of maximum values)
varphi
q
Time (sec)
G-forces (normal and axial)
G-forces
(normal and axial directions)
Normal g
Axial g
Time (sec)

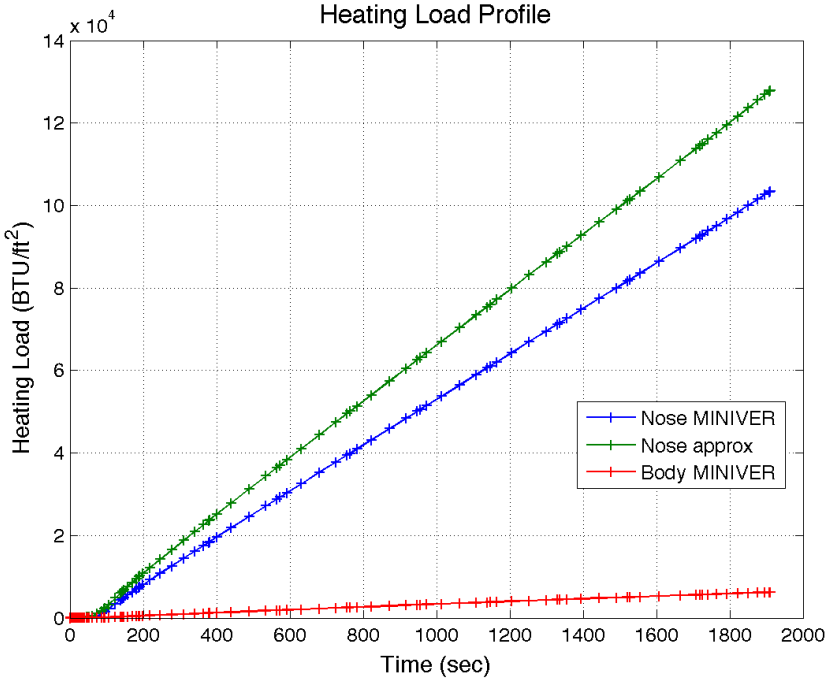

Lookup Table Values
Force Coefficients
0.25
0.2
0.15
0.1
0.05
0
C_L
C_D
0
200
400
600
800
1000
1200
1400
1600
1800
Time (sec)

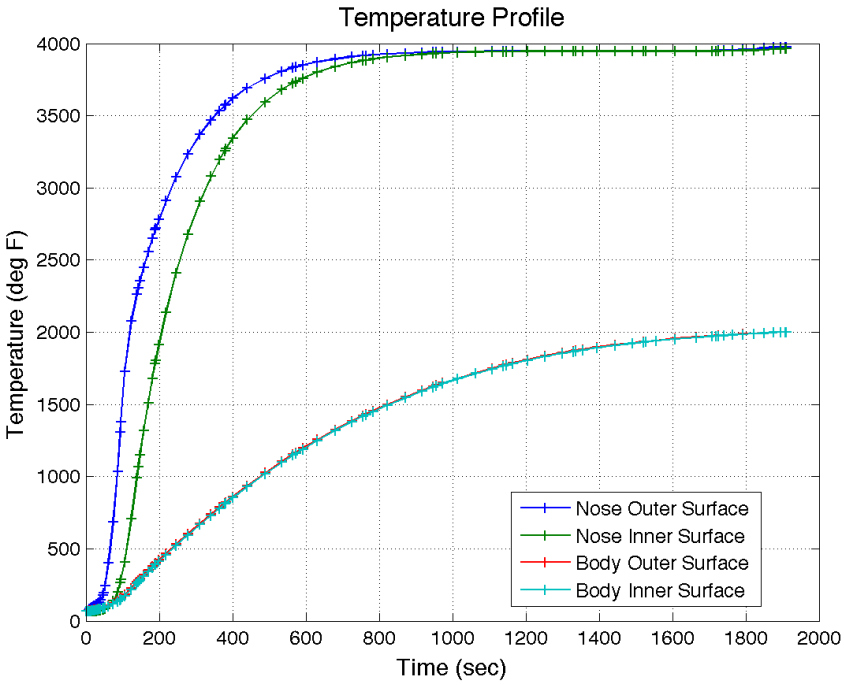

Lookup Table Values
AoAc (inlet area capture ratio)
2
4000
2000
Isp (sec)
AoAc
Isp
0
200
400
600
800
1000
1200
1400
1600
1800
Time (sec)

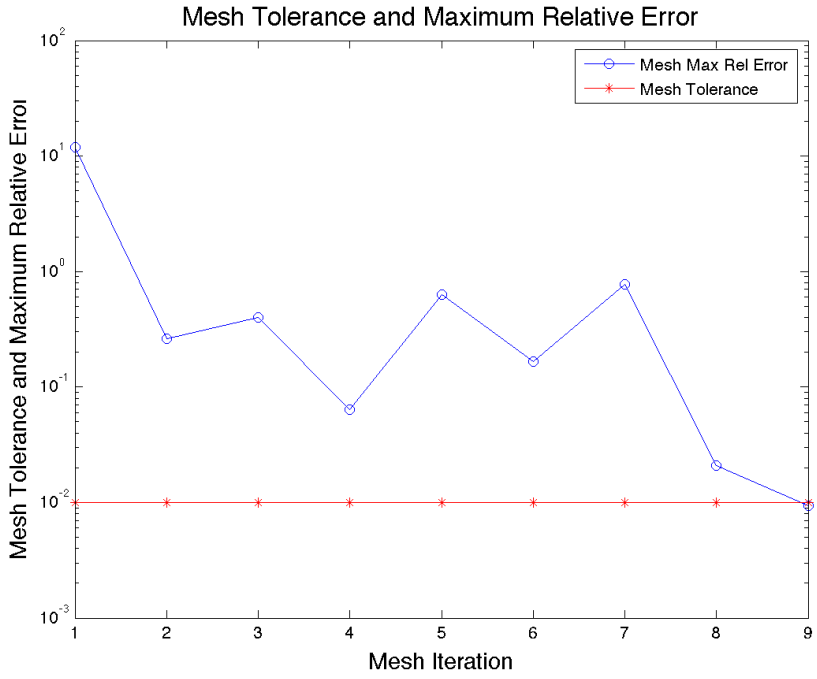

Stability Frame Forces
Forces (10^3 lbf)
2000
1500
1000
500
0
-500
A
T
T-A
N
0
200
400
600
800
1000
1200
1400
1600
1800
Time (sec)

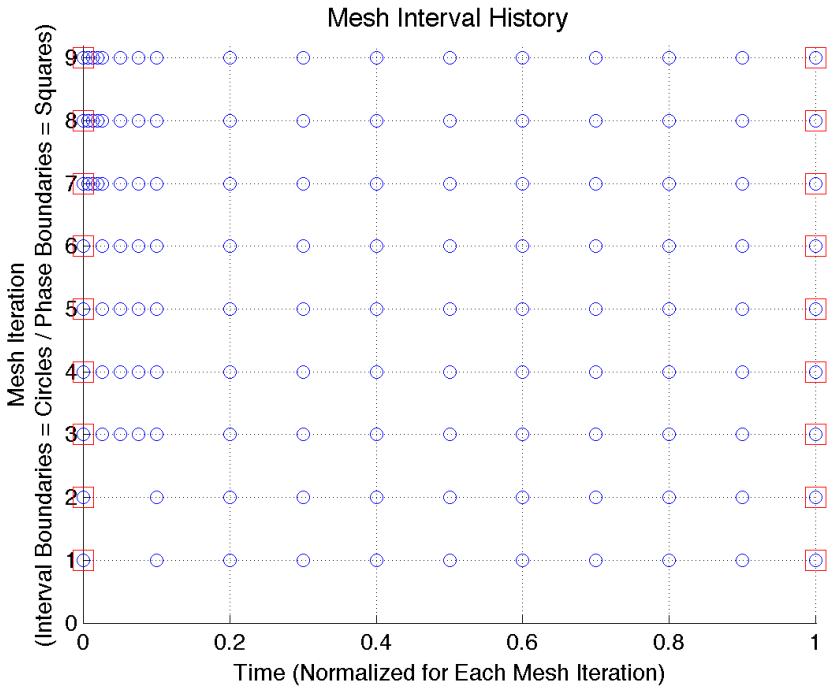

Vehicle Frame Forces
Forces (10^3 lbf)
2000
1500
1000
500
0
-500
L
-D
Tsin(α)+L
Tcos(α)-D
0
200
400
600
800
1000
1200
1400
1600
1800
Time (sec)

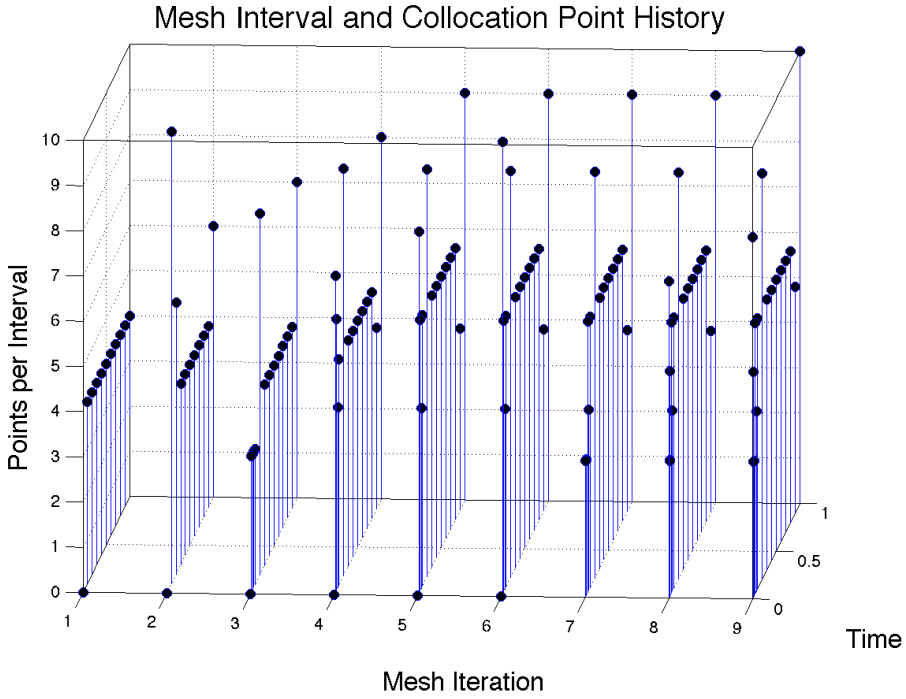

Heating Rate Profile
Heating Rate (BTU/s-ft^2)
Time (sec)
Nose MINIVER
Nose approx
Body MINIVER
0
10
20
30
40
50
60
70
80
90
0
200
400
600
800
1000
1200
1400
1600
1800
2000


Heating Load Profile
x 10^4
Heating Load (BTU/ft^2)
Time (sec)
Nose MINIVER
Nose approx
Body MINIVER
0
2
4
6
8
10
12
14
0
200
400
600
800
1000
1200
1400
1600
1800
2000


Temperature Profile
Temperature (deg F)
Time (sec)
Nose Outer Surface
Nose Inner Surface
Body Outer Surface
Body Inner Surface

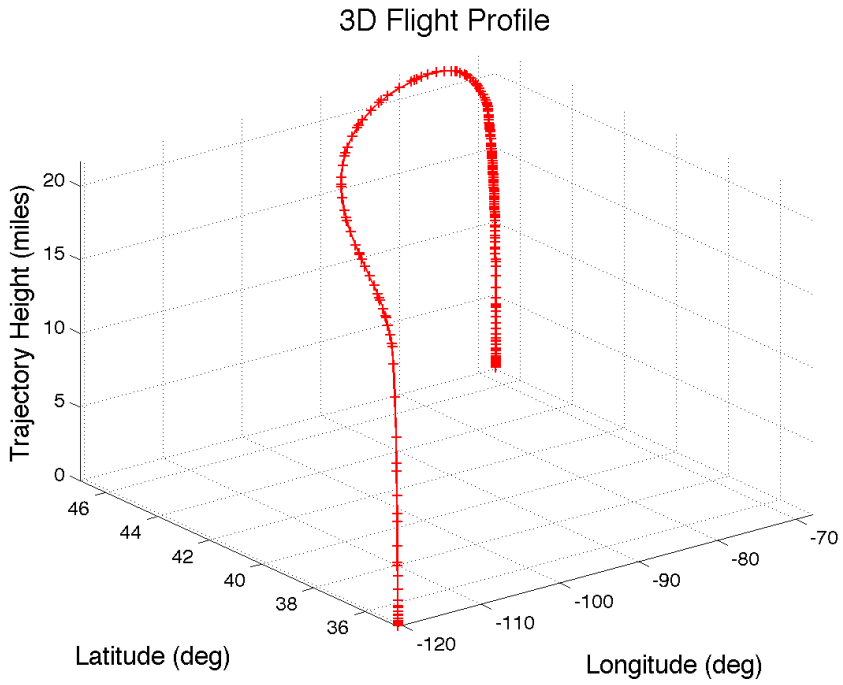

Mesh Tolerance and Maximum Relative Error
Mesh Tolerance and Maximum Relative Error
Mesh Iteration
Mesh Max Rel Error
Mesh Tolerance

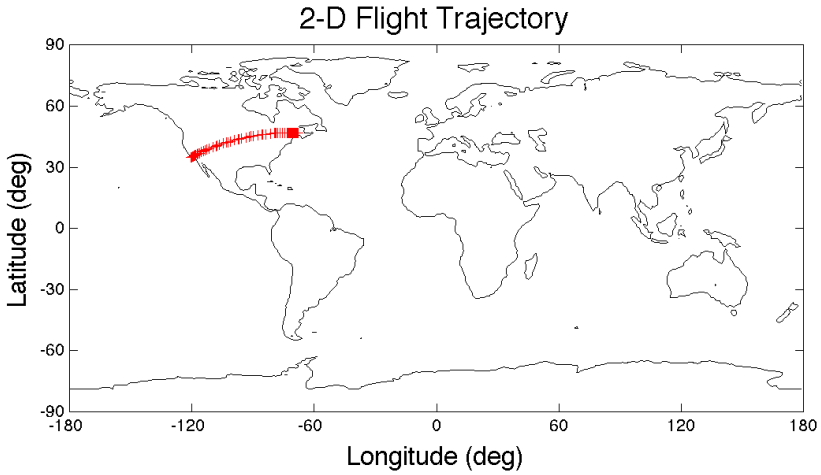

Mesh Interval History
Mesh Iteration
(Interval Boundaries = Circles / Phase Boundaries = Squares)
0
1
2
3
4
5
6
7
8
9
0
0.2
0.4
0.6
0.8
1
Time (Normalized for Each Mesh Iteration)

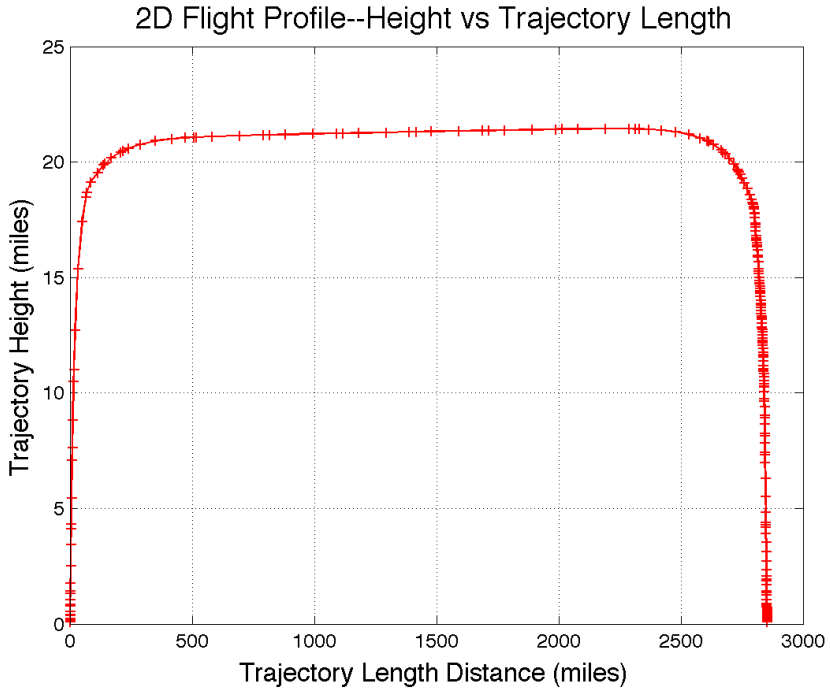

Mesh Interval and Collocation Point History
Points per Interval
0
1
2
3
4
5
6
7
8
9
10
1
2
3
4
5
6
7
8
9
Mesh Iteration
0
0.5
1
Time

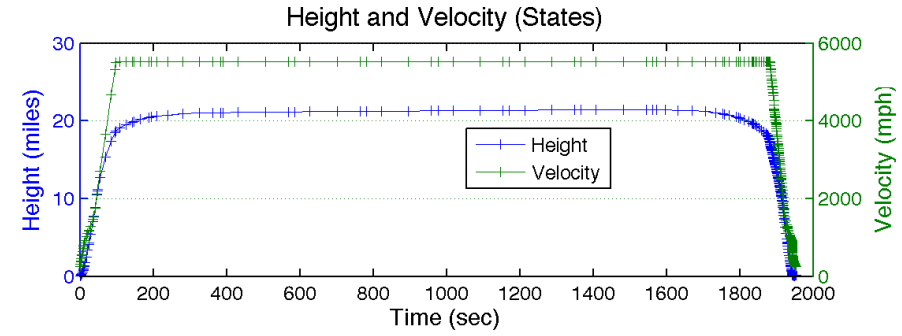
C.4 Minimum Time; Climb, Cruise, and Land

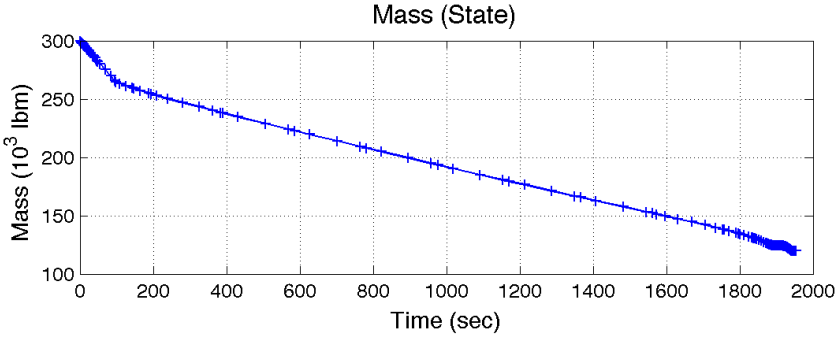


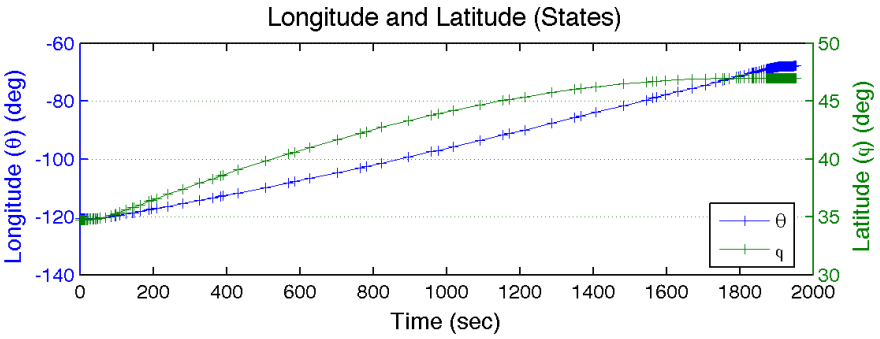


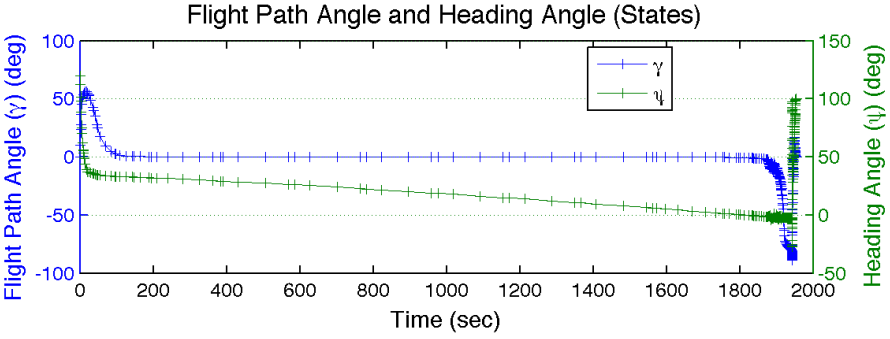

2-D Flight Trajectory
Latitude (deg)
Longitude (deg)
90
60
30
0
-30
-60
-90
-180
-120
-60
0
60
120
180

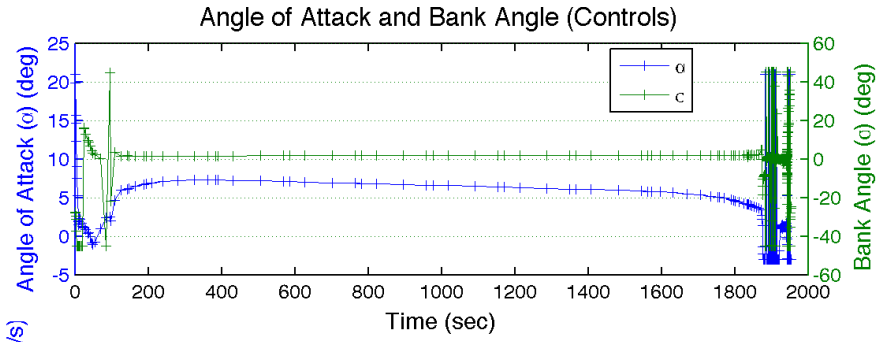

2D Flight Profile--Height vs Trajectory Length
Trajectory Height (miles)
Trajectory Length Distance (miles)
25
20
15
10
5
0
0
500
1000
1500
2000
2500
3000

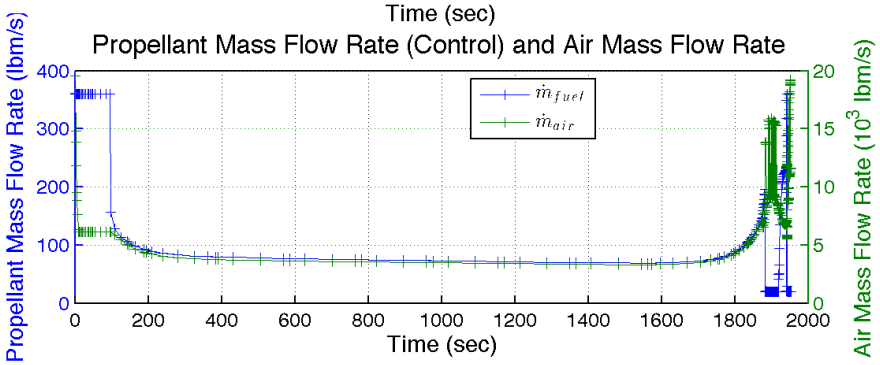

Height and Velocity (States)
Height (miles)
Velocity (mph)
Height
Velocity
Time (sec)

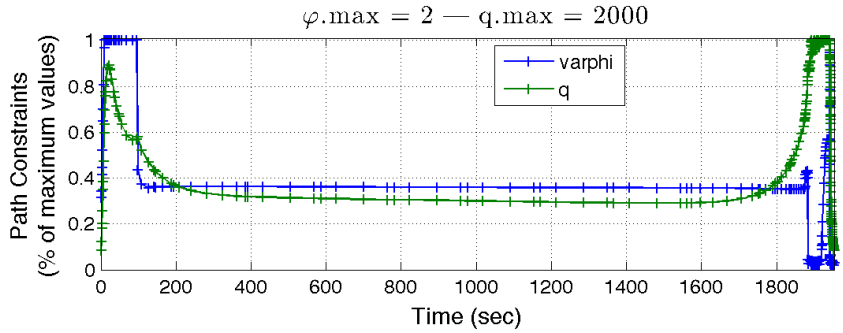

Mass (State)
Mass (10^3 lbm)
Time (sec)

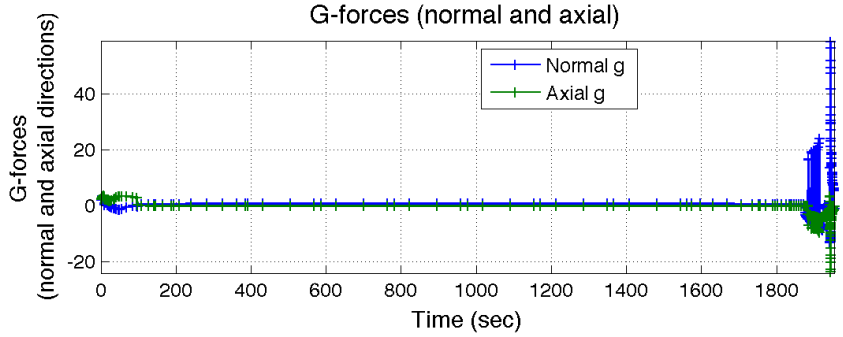

Longitude and Latitude (States)
Longitude (θ) (deg)
Latitude (φ) (deg)
θ
φ
Time (sec)

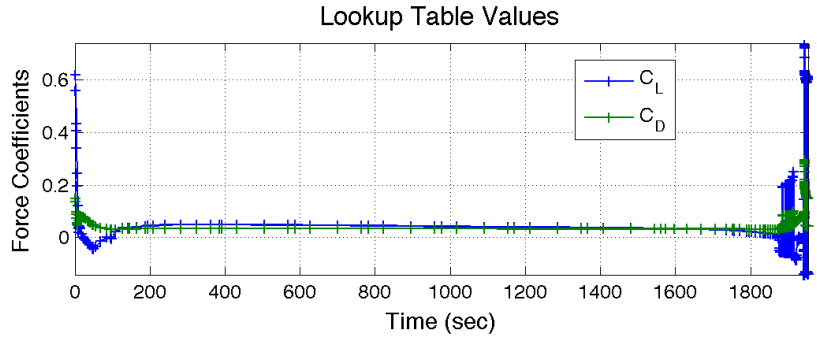

Flight Path Angle and Heading Angle (States)
Flight Path Angle (γ) (deg)
Heading Angle (ψ) (deg)
γ
ψ
Time (sec)

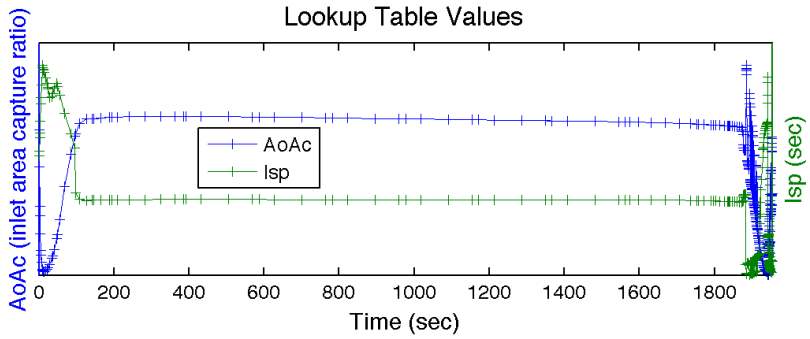

Angle of Attack and Bank Angle (Controls)
Angle of Attack (α) (deg)
Bank Angle (σ) (deg)
α
σ
Time (sec)

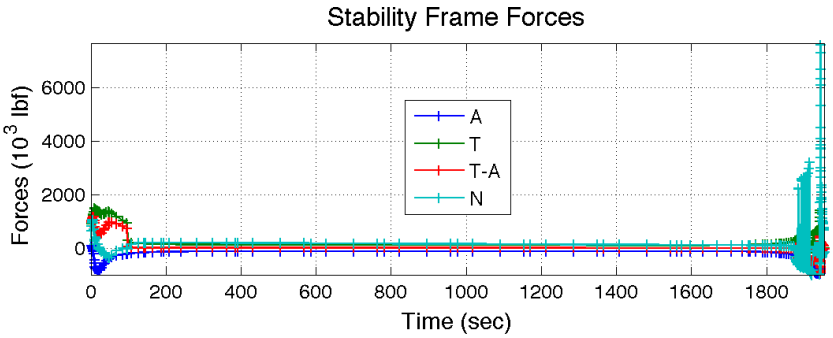

Propellant Mass Flow Rate (Control) and Air Mass Flow Rate
Propellant Mass Flow Rate (lbm/s)
Air Mass Flow Rate (10^3 lbm/s)
$\dot{m}_{fuel}$
$\dot{m}_{air}$
Time (sec)

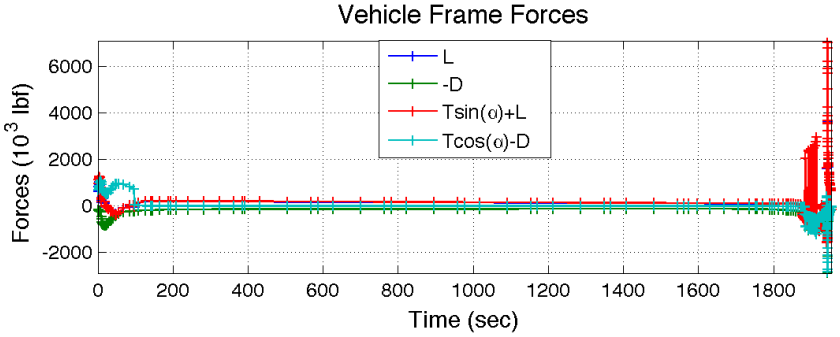

φ.max = 2 — q.max = 2000
Path Constraints
(% of maximum values)
varphi
q
Time (sec)

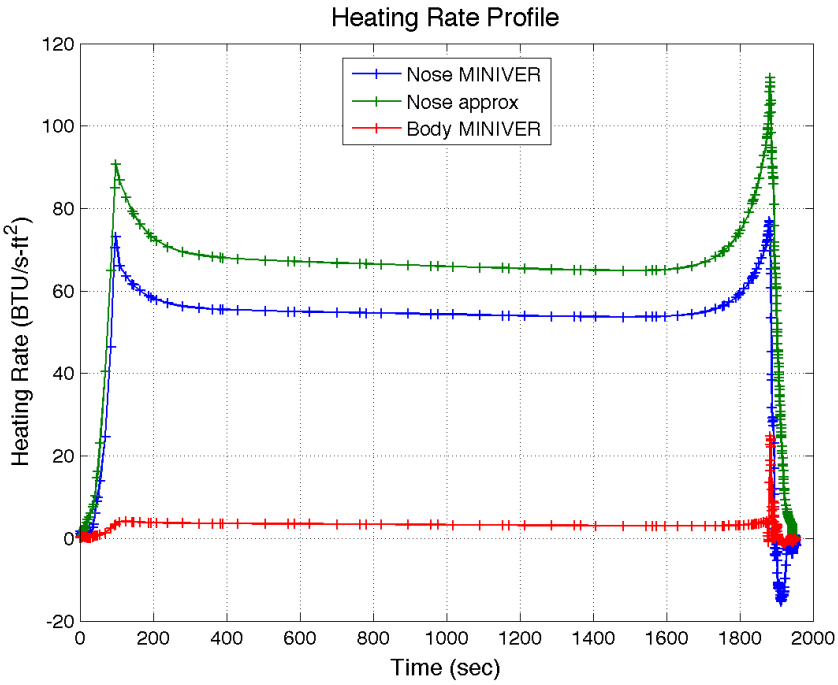

G-forces (normal and axial)
G-forces
(normal and axial directions)
Normal g
Axial g
Time (sec)

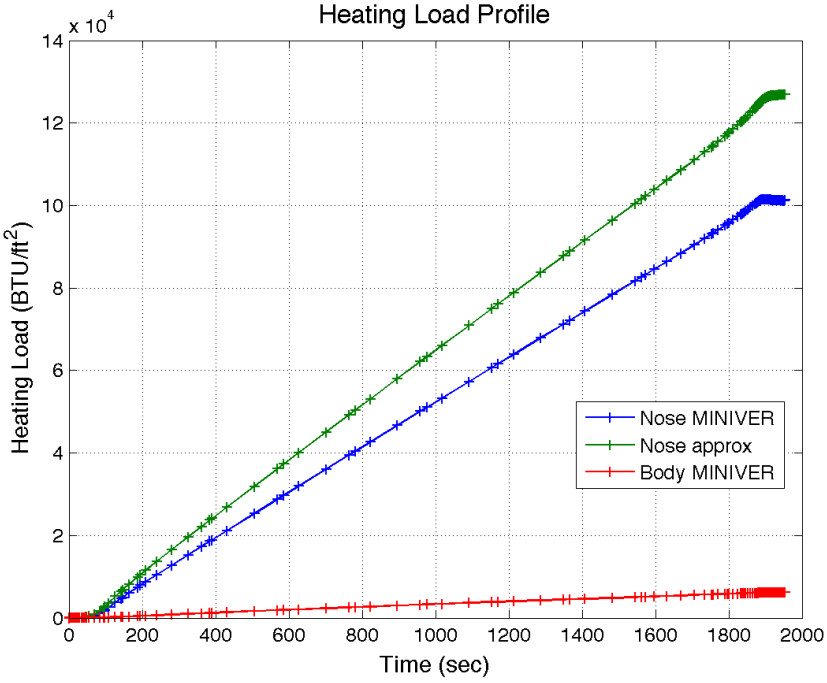

Lookup Table Values
Force Coefficients
0.6
0.4
0.2
0
C_L
C_D
0
200
400
600
800
1000
1200
1400
1600
1800
Time (sec)

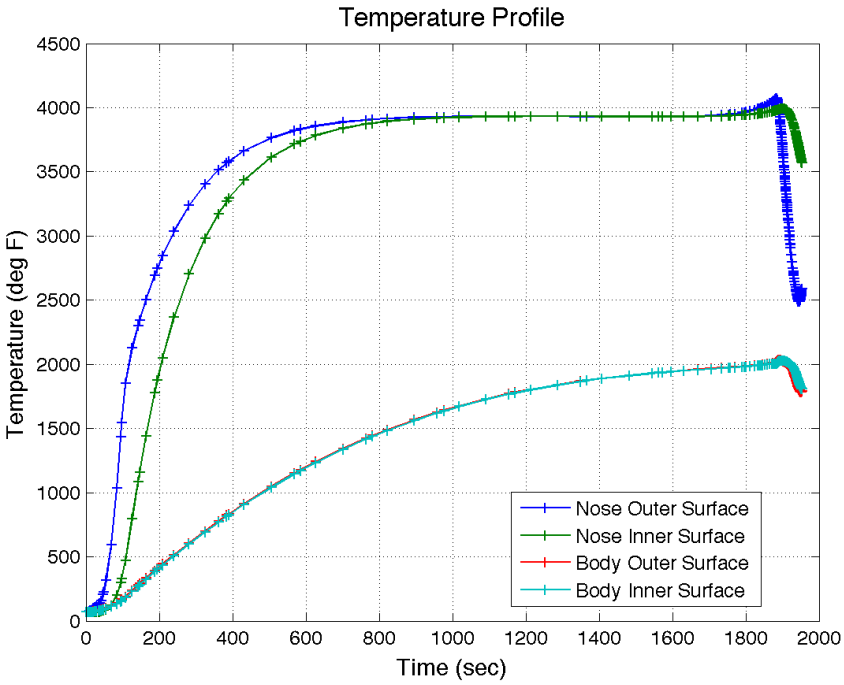

Lookup Table Values
AoAc (inlet area capture ratio)
Isp (sec)
AoAc
Isp
0
200
400
600
800
1000
1200
1400
1600
1800
Time (sec)

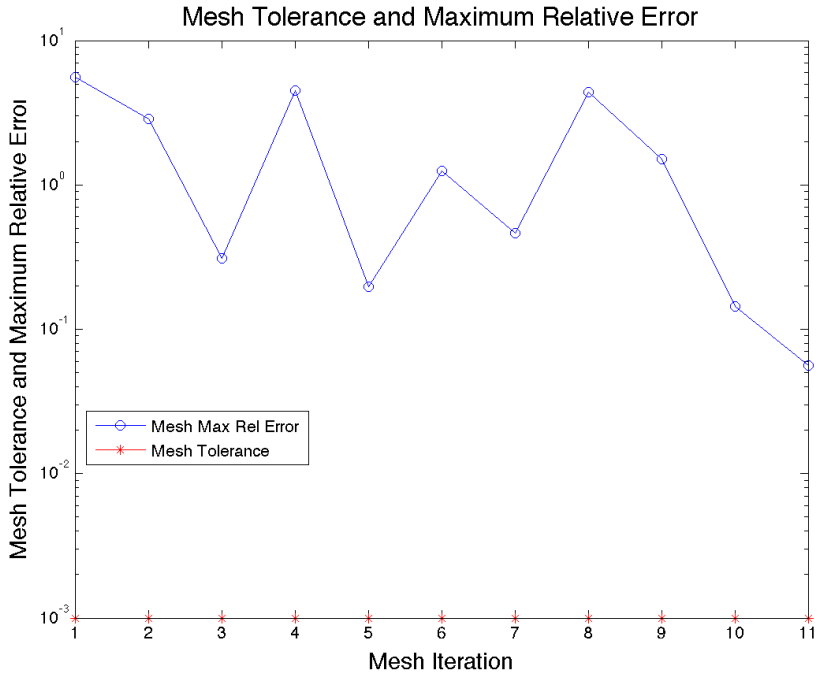

Stability Frame Forces
Forces (10^3 lbf)
6000
4000
2000
0
A
T
T-A
N
0
200
400
600
800
1000
1200
1400
1600
1800
Time (sec)

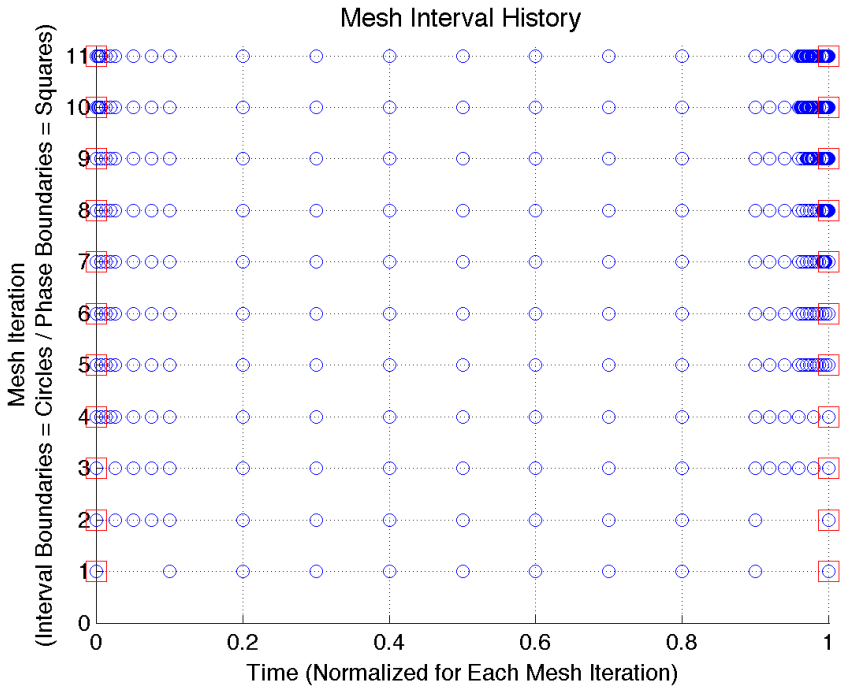

Vehicle Frame Forces
Forces (10^3 lbf)
6000
4000
2000
0
-2000
L
-D
Tsin(α)+L
Tcos(α)-D
0
200
400
600
800
1000
1200
1400
1600
1800
Time (sec)

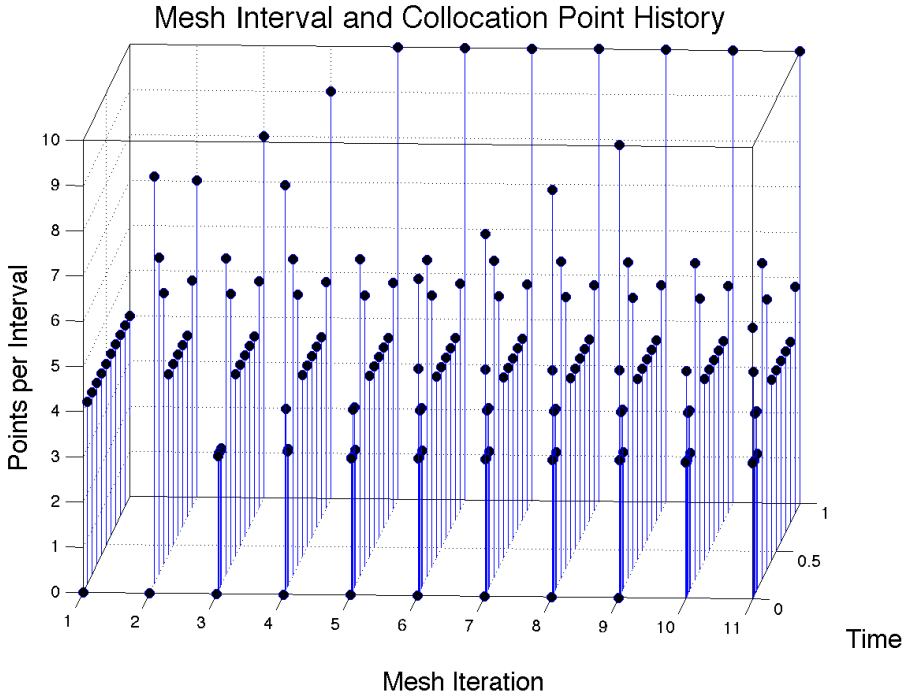

Heating Rate Profile
Nose MINIVER
Nose approx
Body MINIVER
Heating Rate (BTU/s-ft2)
Time (sec)
120
100
80
60
40
20
0
-20
0
200
400
600
800
1000
1200
1400
1600
1800
2000


Heating Load Profile
x 10^4
Heating Load (BTU/ft2)
Time (sec)
Nose MINIVER
Nose approx
Body MINIVER
14
12
10
8
6
4
2
0
0
200
400
600
800
1000
1200
1400
1600
1800
2000


Temperature Profile
Temperature (deg F)
Time (sec)
Nose Outer Surface
Nose Inner Surface
Body Outer Surface
Body Inner Surface

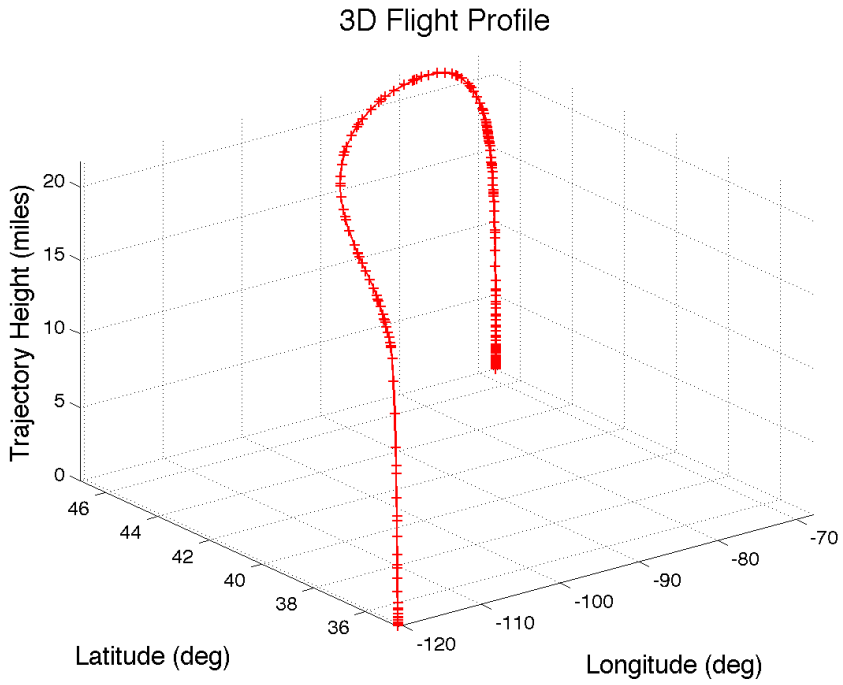

Mesh Tolerance and Maximum Relative Error
Mesh Tolerance and Maximum Relative Error
Mesh Iteration
Mesh Max Rel Error
Mesh Tolerance

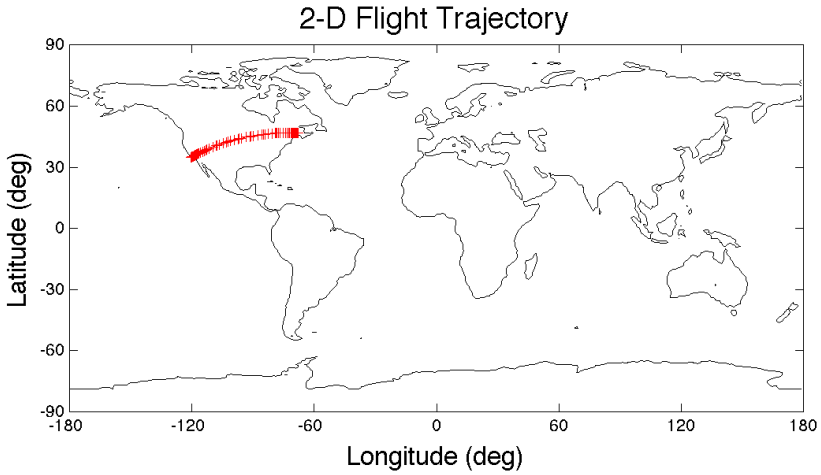

Mesh Interval History
Mesh Iteration
(Interval Boundaries = Circles / Phase Boundaries = Squares)
0
1
2
3
4
5
6
7
8
9
10
11
0
0.2
0.4
0.6
0.8
1
Time (Normalized for Each Mesh Iteration)

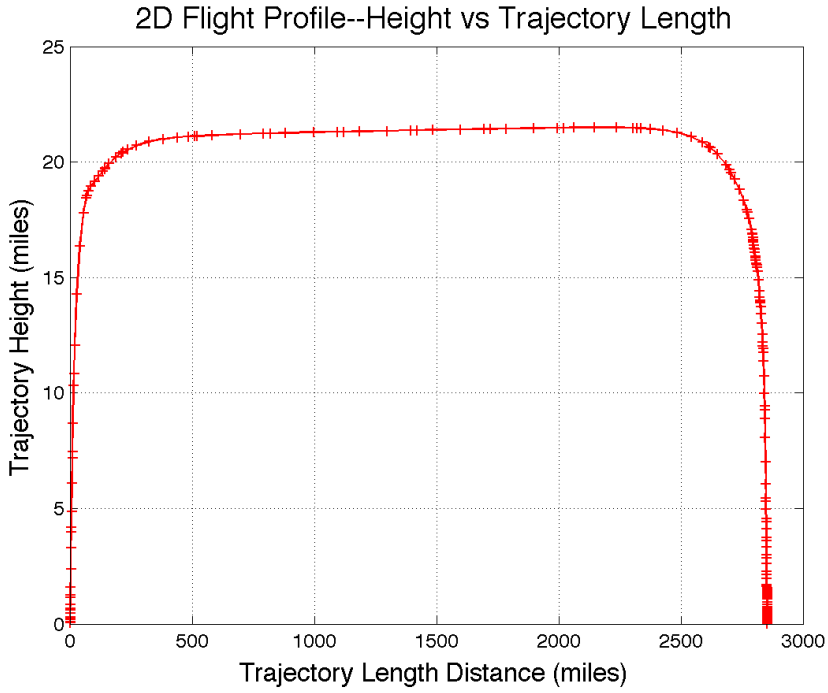

Mesh Interval and Collocation Point History
Points per Interval
0
1
2
3
4
5
6
7
8
9
10
1
2
3
4
5
6
7
8
9
10
11
Mesh Iteration
0
0.5
1
Time

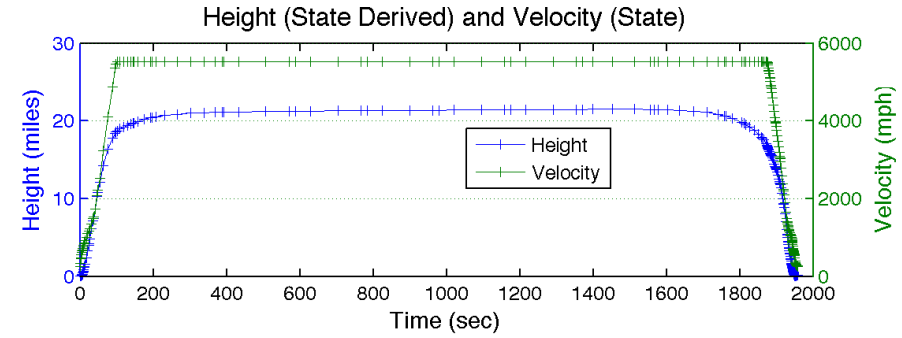
C.5 Minimum Time with Control Penalty; Climb, Cruise, and Land

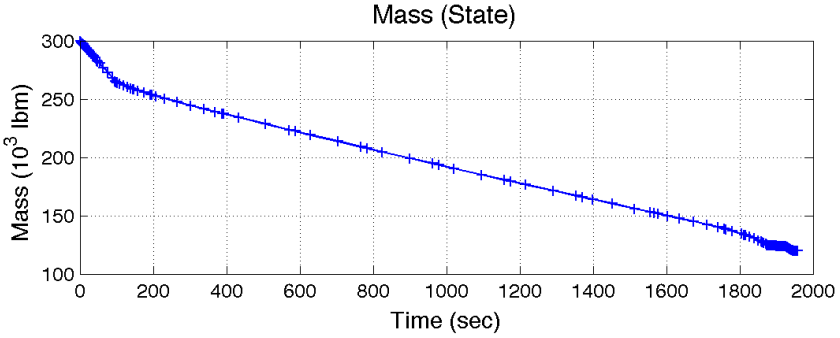


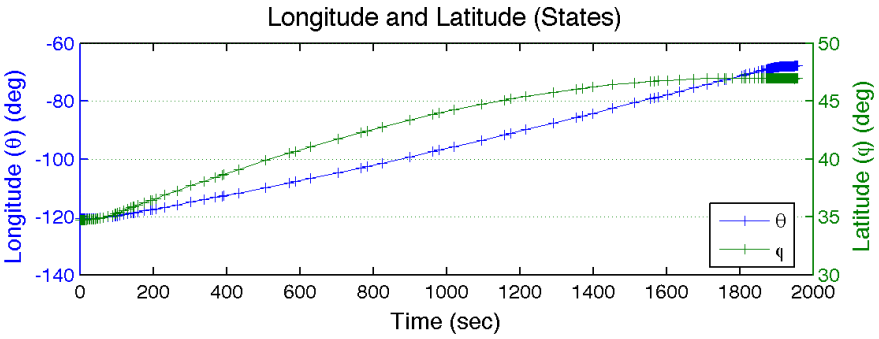


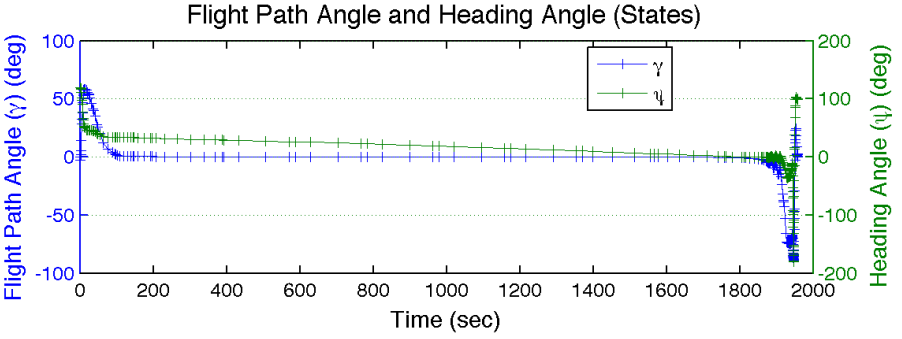

2-D Flight Trajectory
Latitude (deg)
Longitude (deg)
90
60
30
0
-30
-60
-90
-180
-120
-60
0
60
120
180

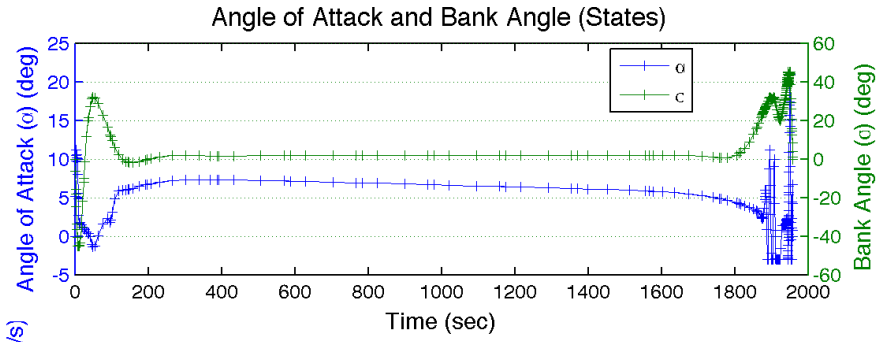

2D Flight Profile--Height vs Trajectory Length
Trajectory Height (miles)
Trajectory Length Distance (miles)
25
20
15
10
5
0
0
500
1000
1500
2000
2500
3000

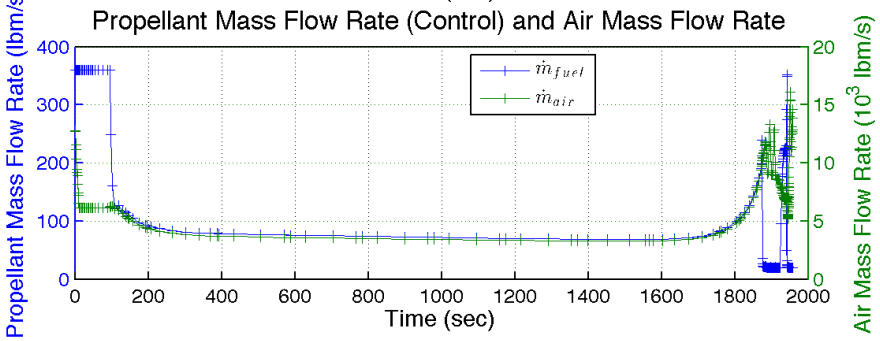

Height (State Derived) and Velocity (State)
Height (miles)
Velocity (mph)
Height
Velocity
Time (sec)

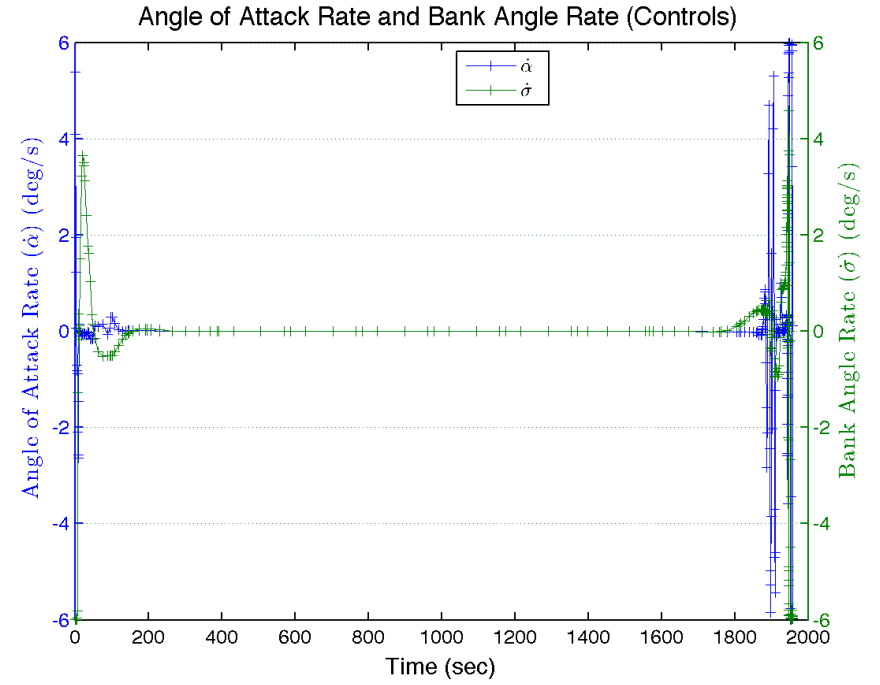

Mass (State)
Mass (10^3 lbm)
Time (sec)

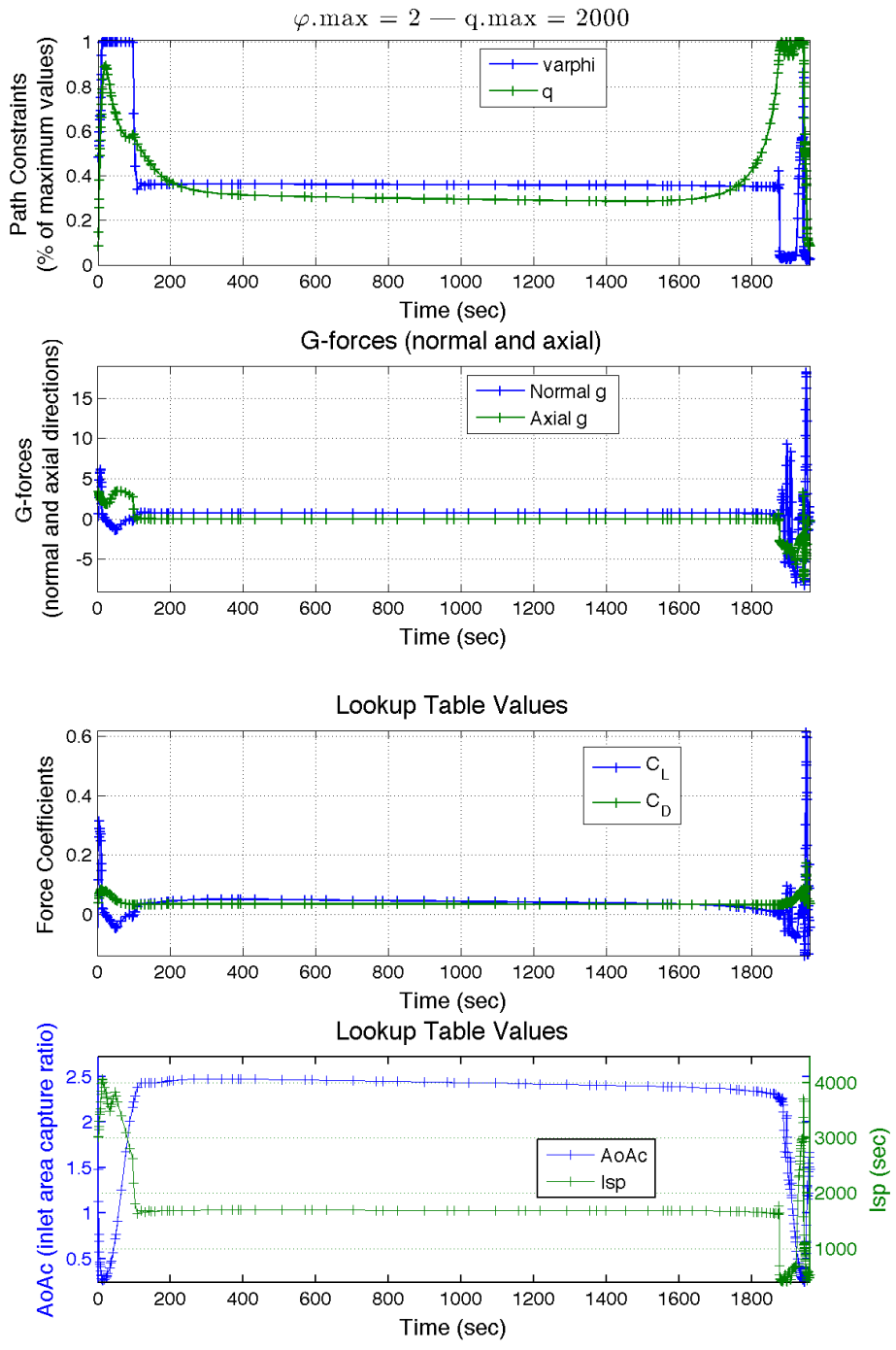

Longitude and Latitude (States)
Longitude (θ) (deg)
Latitude (φ) (deg)
θ
φ
Time (sec)

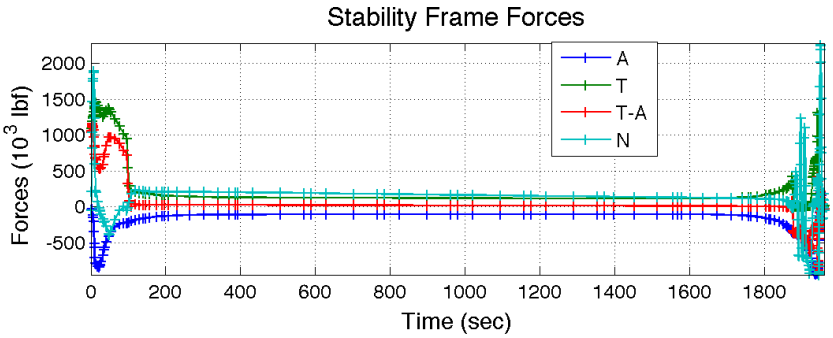

Flight Path Angle and Heading Angle (States)
Flight Path Angle (γ) (deg)
Heading Angle (ψ) (deg)
γ
ψ
Time (sec)

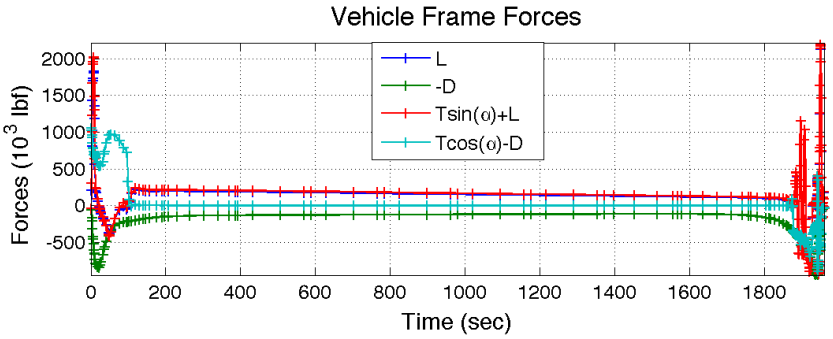

Angle of Attack and Bank Angle (States)
Angle of Attack (α) (deg)
Bank Angle (σ) (deg)
Time (sec)
α
c

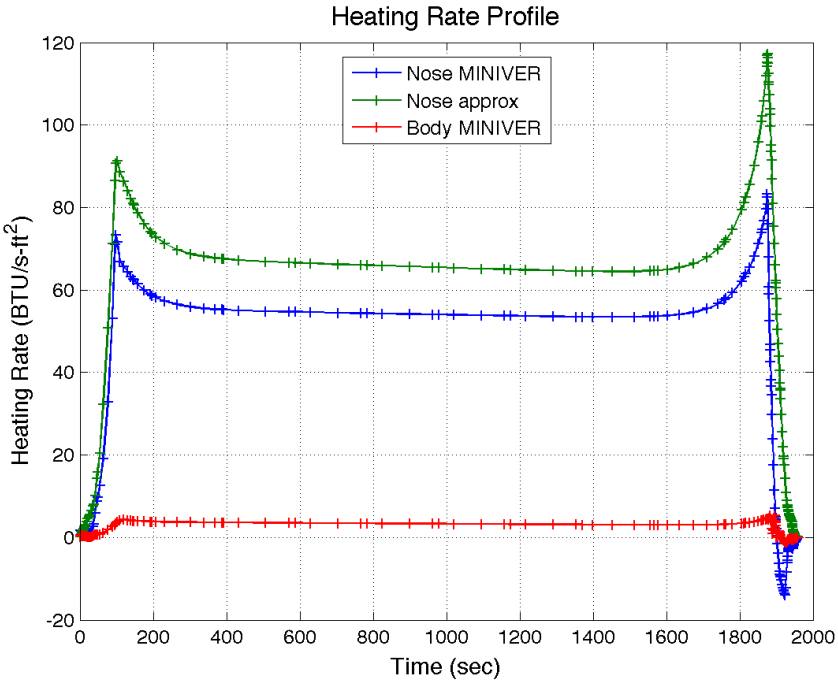

Propellant Mass Flow Rate (Control) and Air Mass Flow Rate
Propellant Mass Flow Rate (lbm/s)
Air Mass Flow Rate (10^3 lbm/s)
Time (sec)
$\dot{m}_{fuel}$
$\dot{m}_{air}$

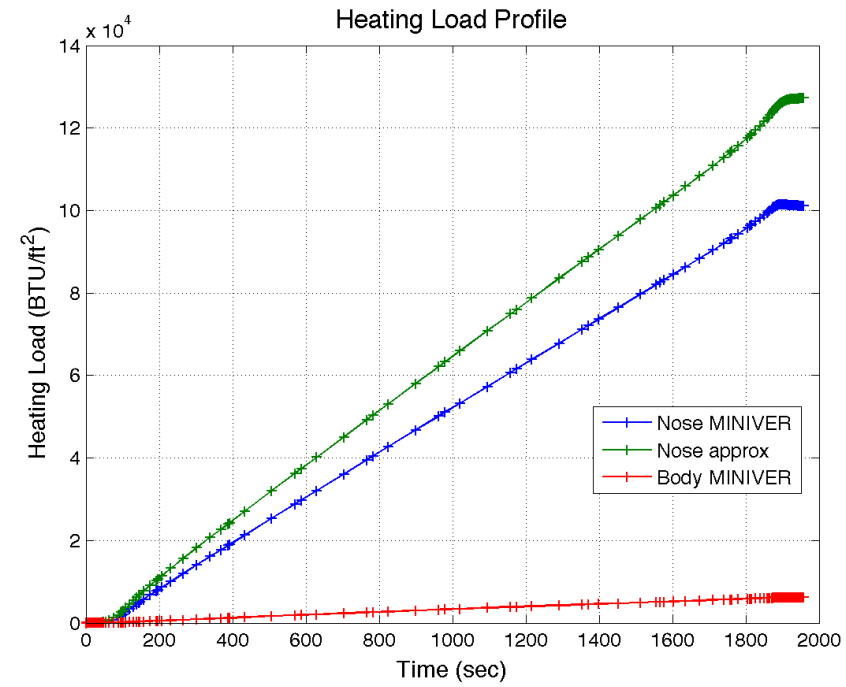

Angle of Attack Rate and Bank Angle Rate (Controls)
Angle of Attack Rate ($\dot{\alpha}$) (deg/s)
Bank Angle Rate ($\dot{\sigma}$) (deg/s)
Time (sec)
$\dot{\alpha}$
$\dot{\sigma}$

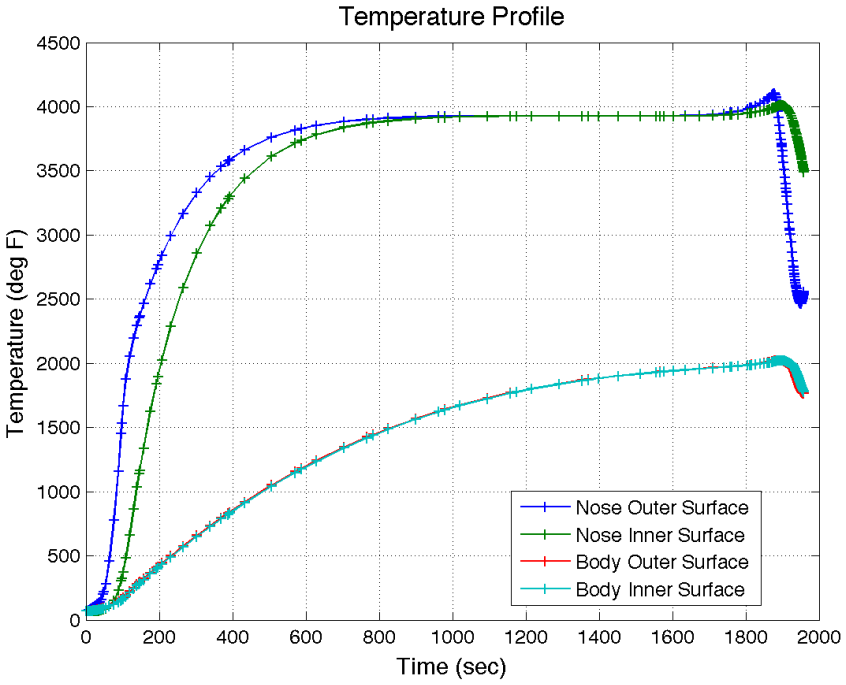

φ.max = 2 — q.max = 2000
Path Constraints
(% of maximum values)
varphi
q
Time (sec)
G-forces (normal and axial)
G-forces
(normal and axial directions)
Normal g
Axial g
Time (sec)
Lookup Table Values
Force Coefficients
C_L
C_D
Time (sec)
Lookup Table Values
AoAc (inlet area capture ratio)
Isp (sec)
AoAc
Isp
Time (sec)

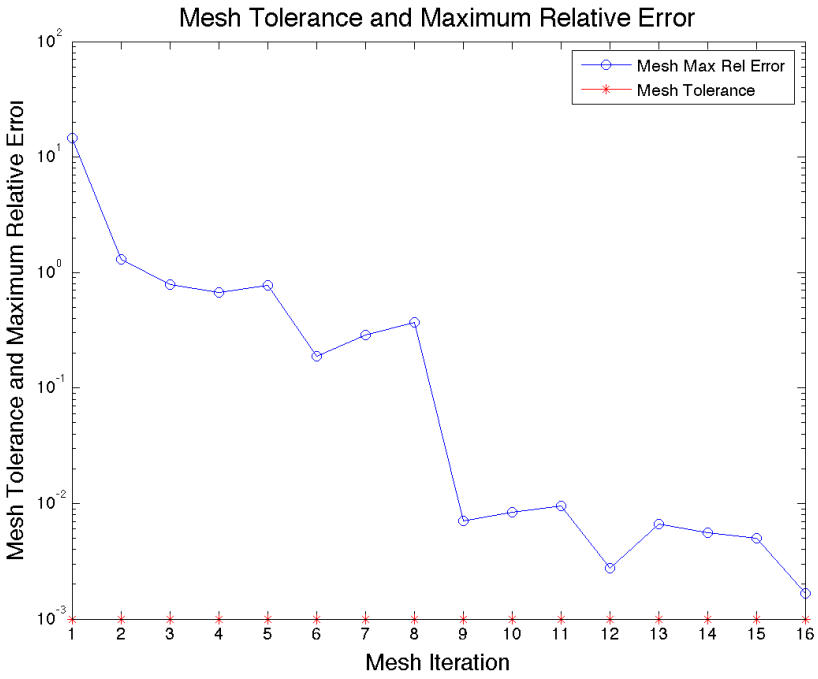

Stability Frame Forces
A
T
T-A
N
Forces (10^3 lbf)
Time (sec)

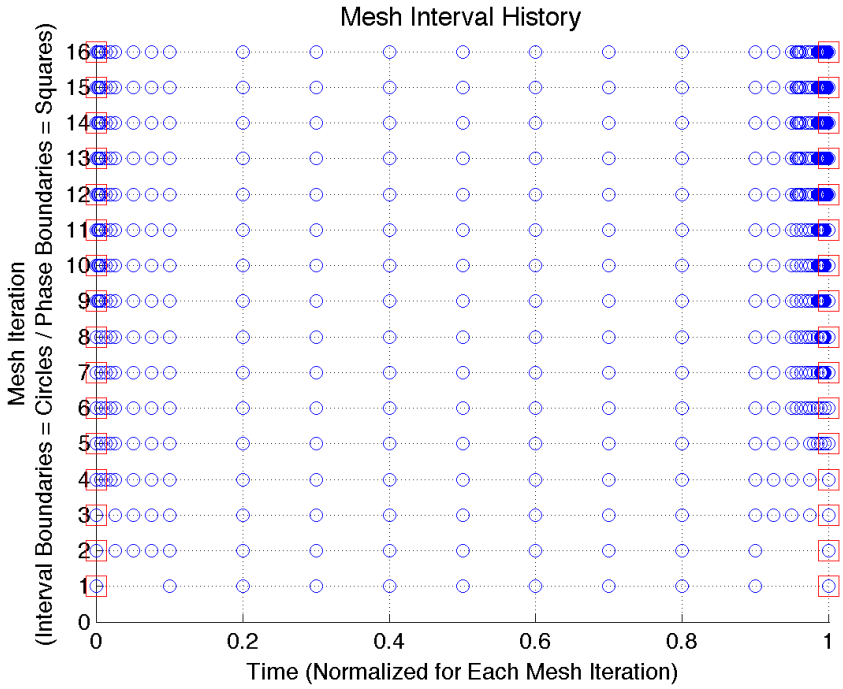

Vehicle Frame Forces
L
-D
Tsin(α)+L
Tcos(α)-D
Forces (10^3 lbf)
Time (sec)

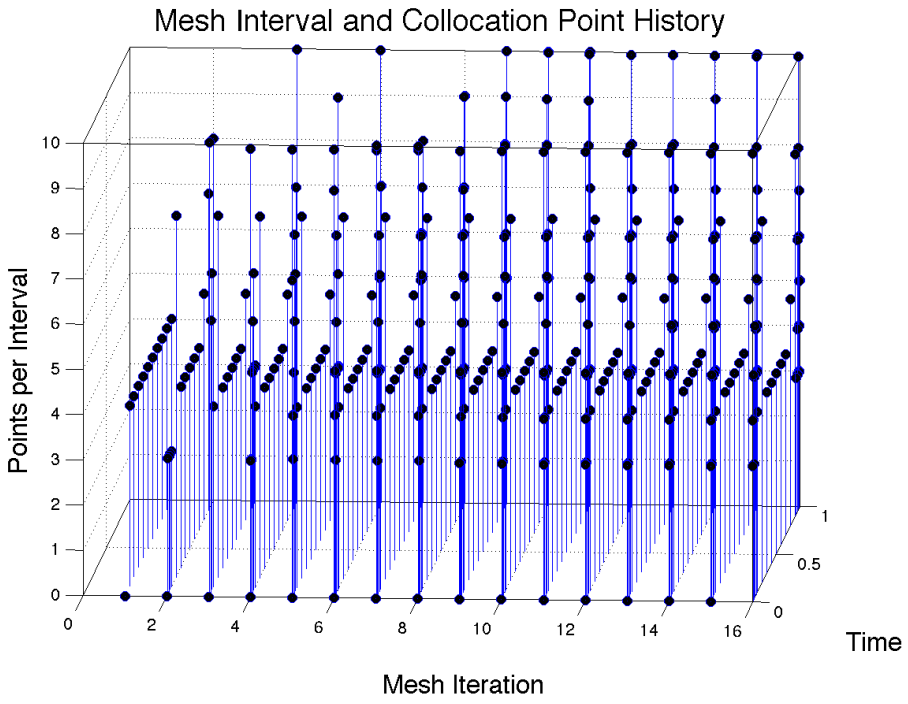

Heating Rate Profile
Nose MINIVER
Nose approx
Body MINIVER
Heating Rate (BTU/s-ft^2)
Time (sec)


Heating Load Profile
x 10^4
Heating Load (BTU/ft^2)
Time (sec)
Nose MINIVER
Nose approx
Body MINIVER

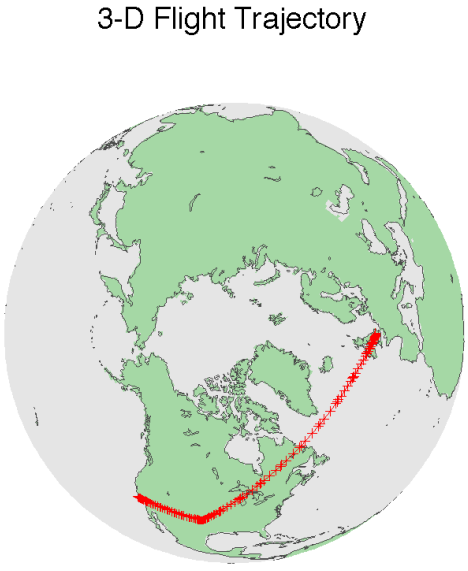

Temperature Profile
Temperature (deg F)
Time (sec)
Nose Outer Surface
Nose Inner Surface
Body Outer Surface
Body Inner Surface

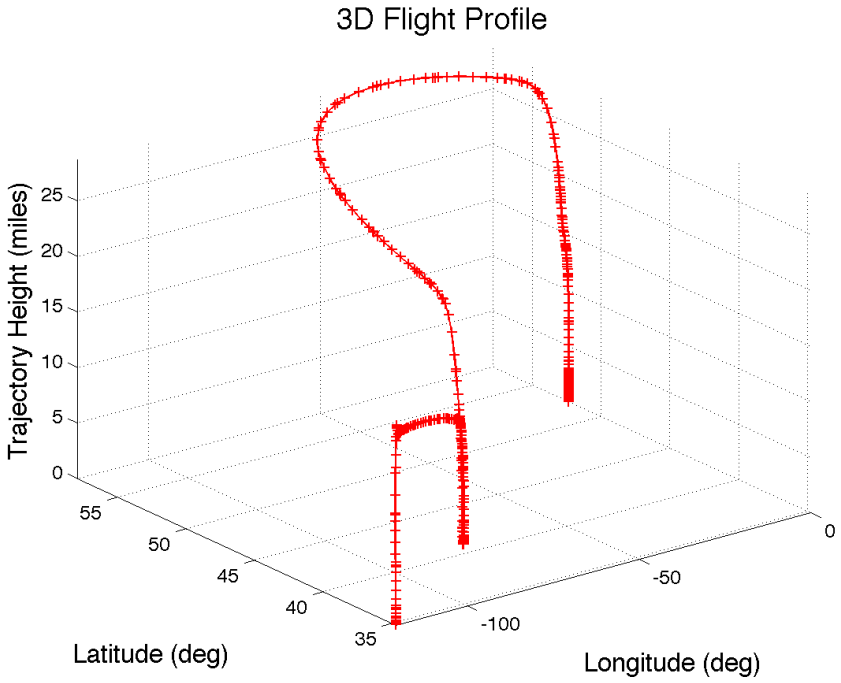

Mesh Tolerance and Maximum Relative Error
Mesh Max Rel Error
Mesh Tolerance
Mesh Tolerance and Maximum Relative Error
Mesh Iteration

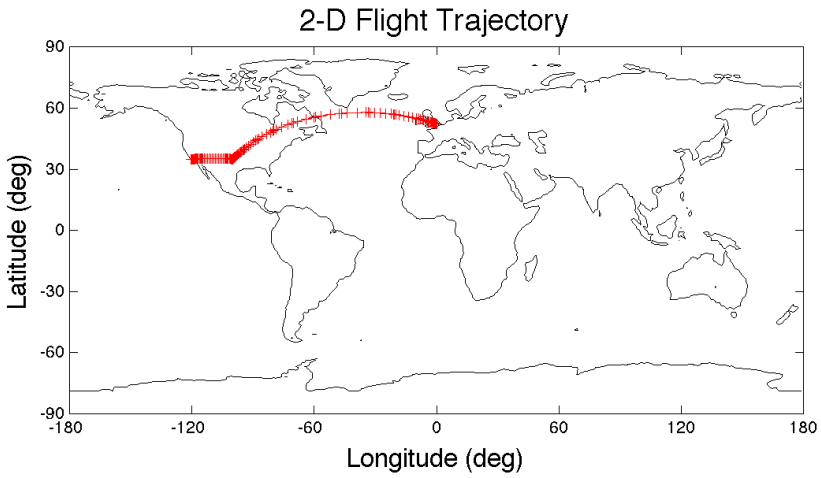

Mesh Interval History
Mesh Iteration
(Interval Boundaries = Circles / Phase Boundaries = Squares)
Time (Normalized for Each Mesh Iteration)

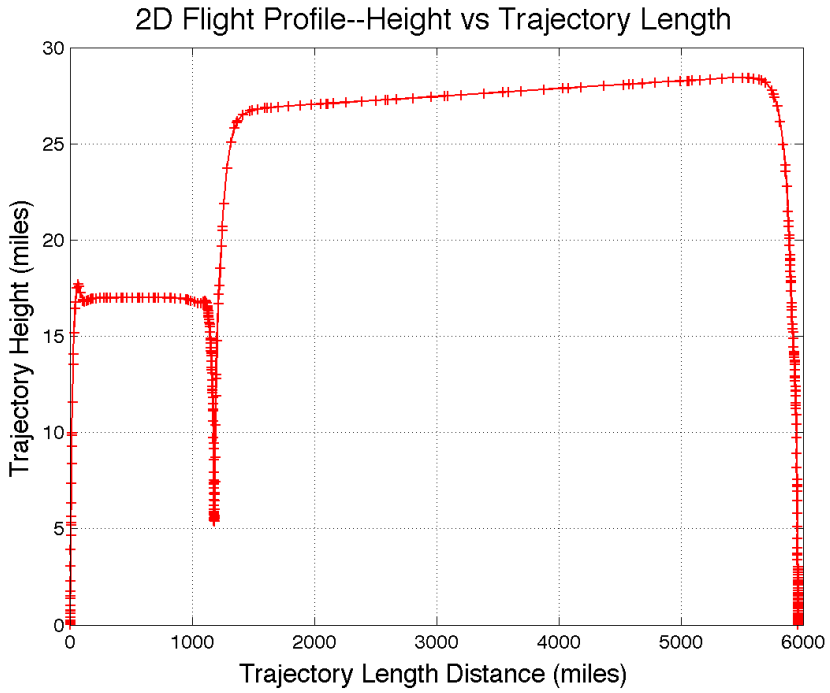


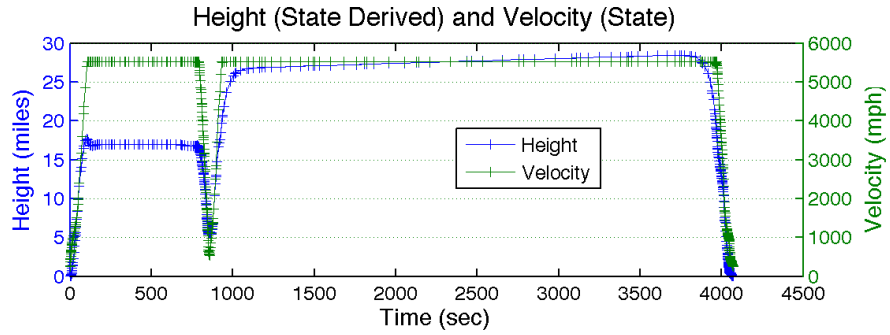
C.6 Minimum Time with Control Penalty; Climb, Cruise, Refuel, Cruise, and Land

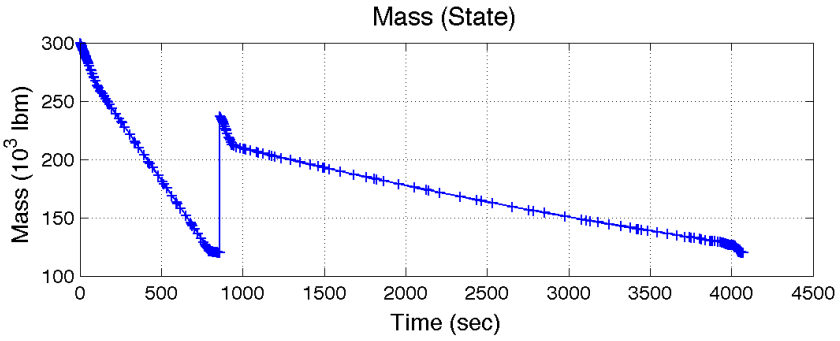


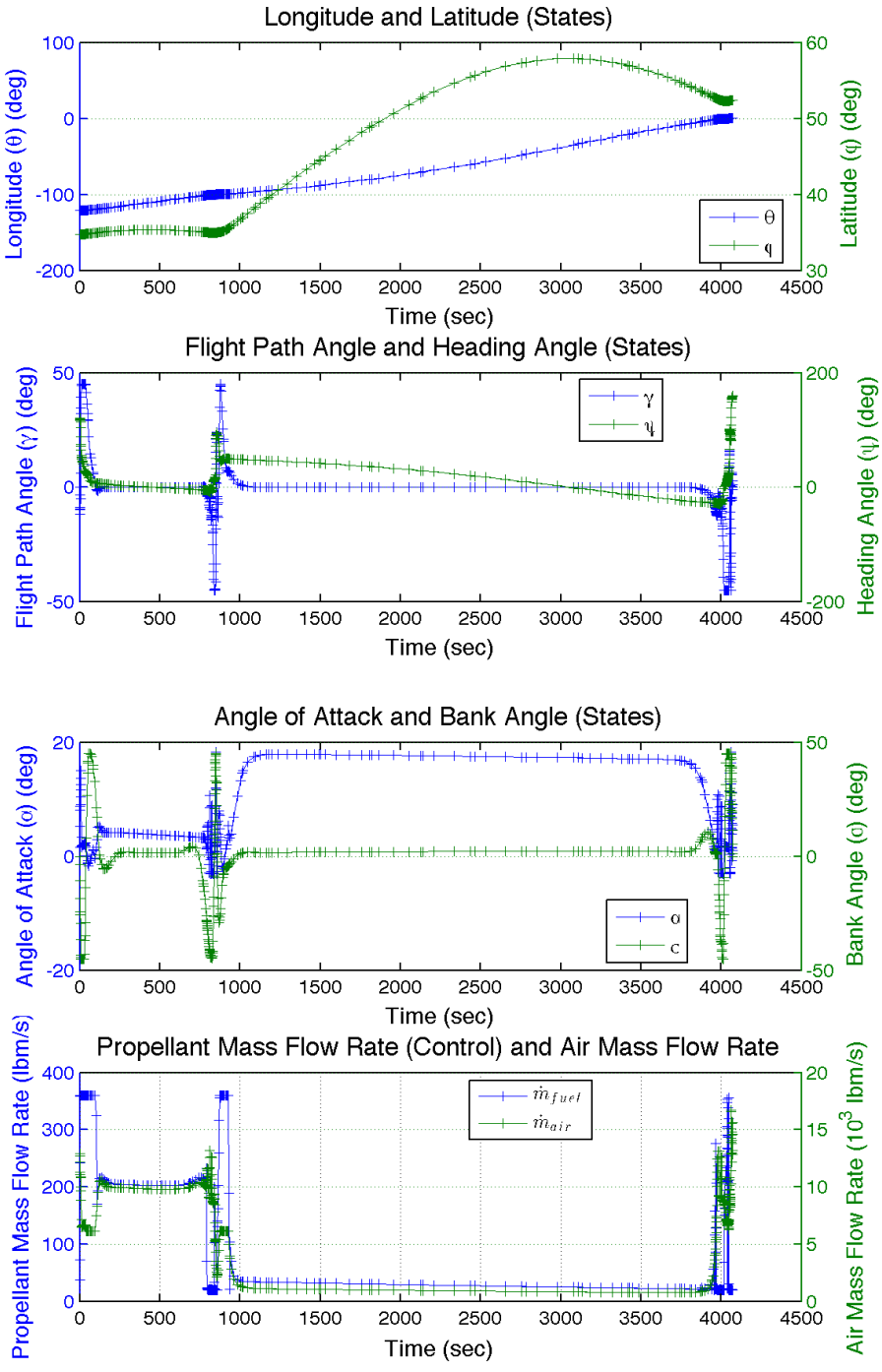

3D Flight Profile
Trajectory Height (miles)
25
20
15
10
5
0
55
50
45
40
35
Latitude (deg)
0
-50
-100
Longitude (deg)

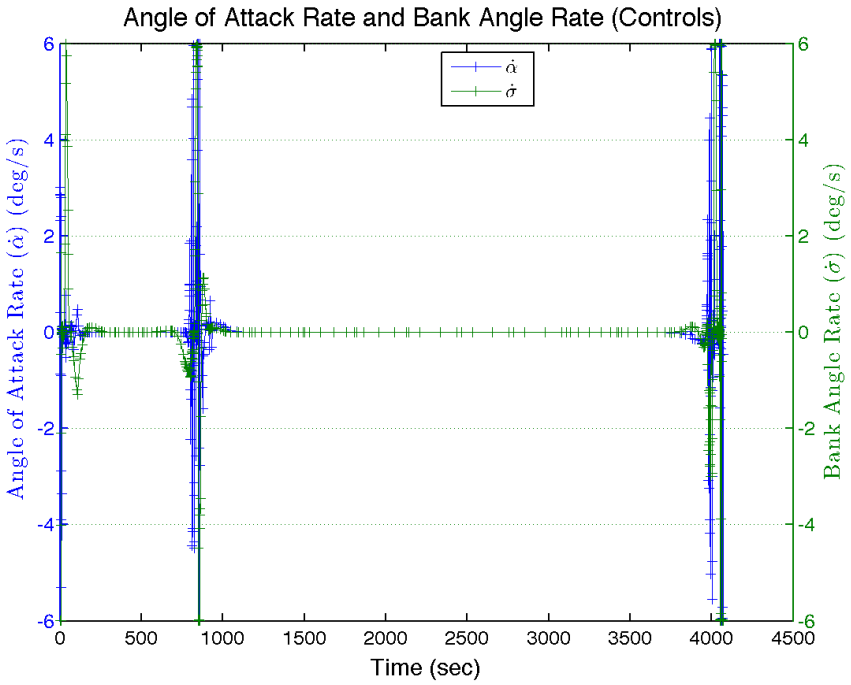

2-D Flight Trajectory
Latitude (deg)
90
60
30
0
-30
-60
-90
-180
-120
-60
0
60
120
180
Longitude (deg)

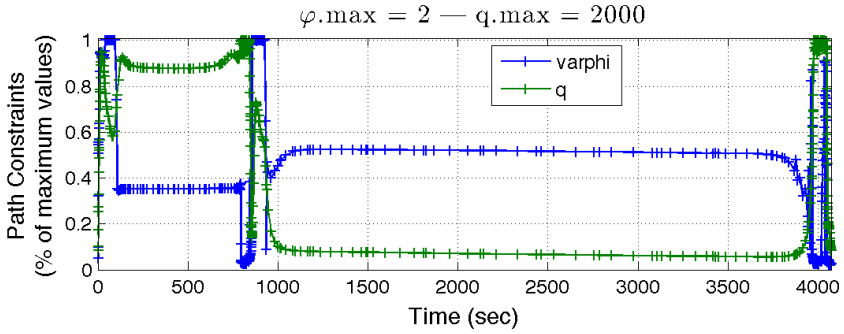

2D Flight Profile--Height vs Trajectory Length
Trajectory Height (miles)
Trajectory Length Distance (miles)
0
5
10
15
20
25
30
0
1000
2000
3000
4000
5000
6000

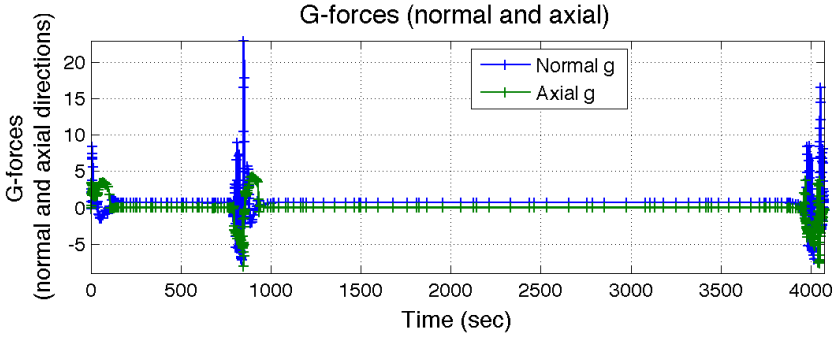

Height (State Derived) and Velocity (State)
Height (miles)
Velocity (mph)
Time (sec)
Height
Velocity
0
5
10
15
20
25
30
0
1000
2000
3000
4000
5000
6000
0
500
1000
1500
2000
2500
3000
3500
4000
4500

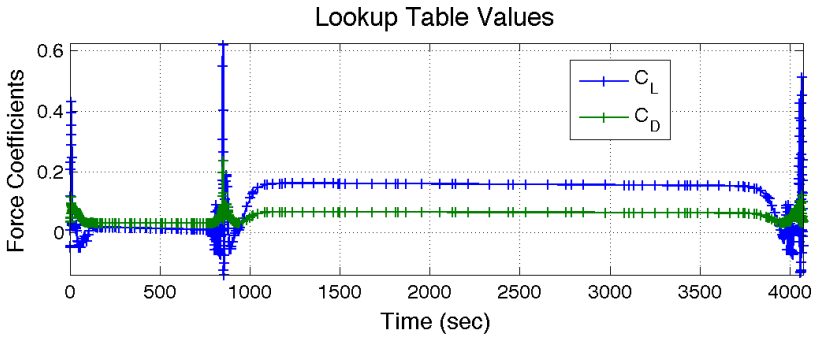

Mass (State)
Mass (10^3 lbm)
Time (sec)
100
150
200
250
300
0
500
1000
1500
2000
2500
3000
3500
4000
4500

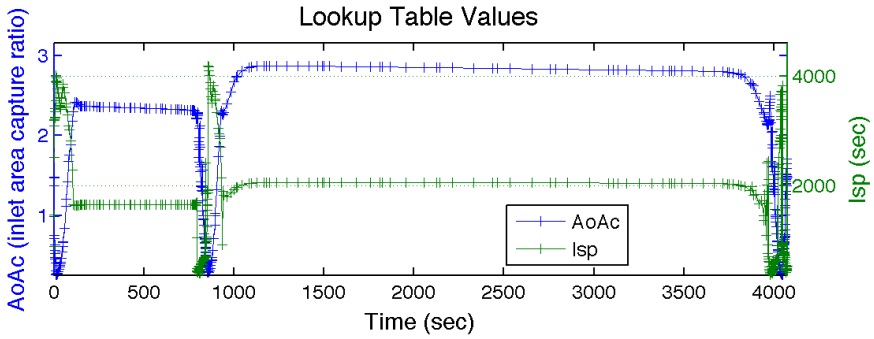

Longitude and Latitude (States)
Longitude (θ) (deg)
Latitude (φ) (deg)
Time (sec)
θ
φ
Flight Path Angle and Heading Angle (States)
Flight Path Angle (γ) (deg)
Heading Angle (ψ) (deg)
Time (sec)
γ
ψ
Angle of Attack and Bank Angle (States)
Angle of Attack (α) (deg)
Bank Angle (σ) (deg)
Time (sec)
α
σ
Propellant Mass Flow Rate (Control) and Air Mass Flow Rate
Propellant Mass Flow Rate (lbm/s)
Air Mass Flow Rate (10^3 lbm/s)
Time (sec)
$\dot{m}_{fuel}$
$\dot{m}_{air}$

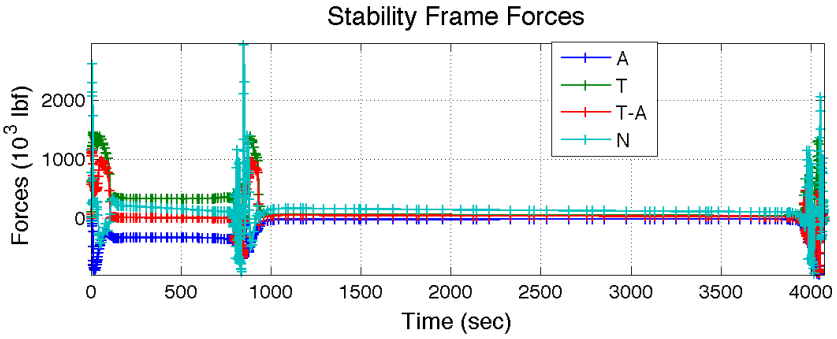

Angle of Attack Rate and Bank Angle Rate (Controls)
$\dot{\alpha}$
$\dot{\sigma}$
Angle of Attack Rate ($\dot{\alpha}$) (deg/s)
Bank Angle Rate ($\dot{\sigma}$) (deg/s)
Time (sec)

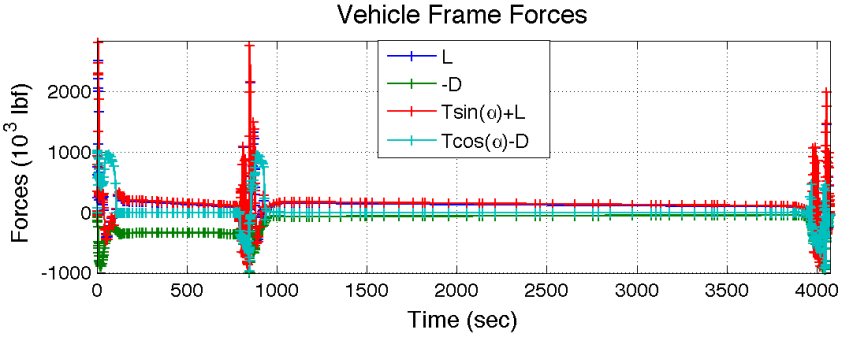

φ.max = 2 — q.max = 2000
varphi
q
Path Constraints
(% of maximum values)
Time (sec)

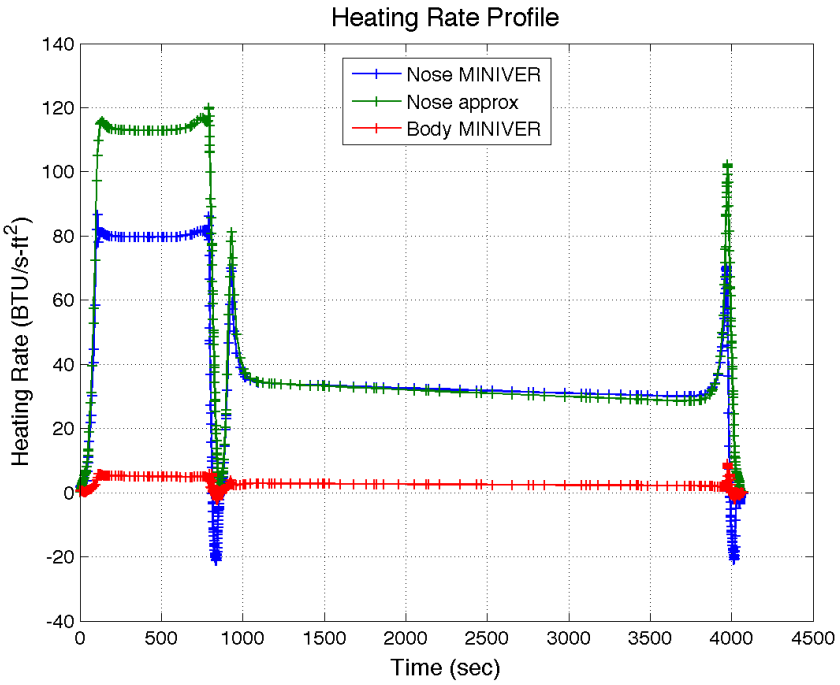

G-forces (normal and axial)
Normal g
Axial g
G-forces
(normal and axial directions)
Time (sec)

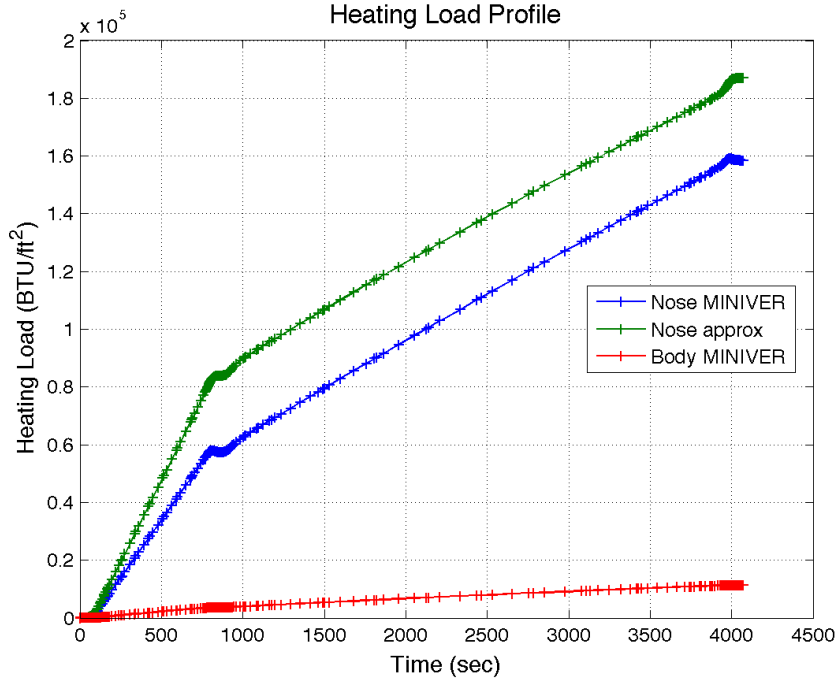

Lookup Table Values
Force Coefficients
0.6
0.4
0.2
0
C_L
C_D
0
500
1000
1500
2000
2500
3000
3500
4000
Time (sec)

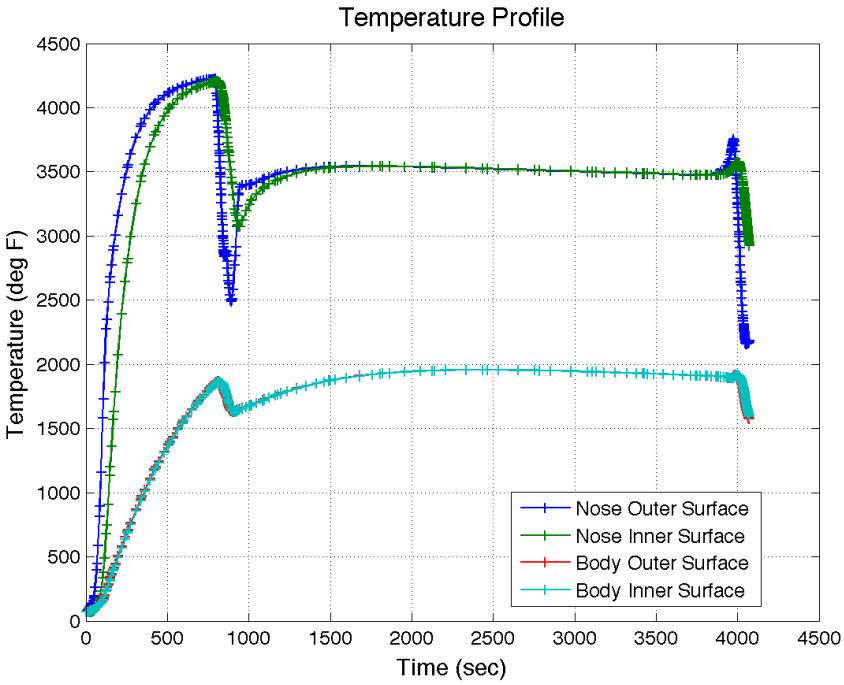

Lookup Table Values
AoAc (inlet area capture ratio)
3
2
1
4000
2000
Isp (sec)
AoAc
Isp
0
500
1000
1500
2000
2500
3000
3500
4000
Time (sec)

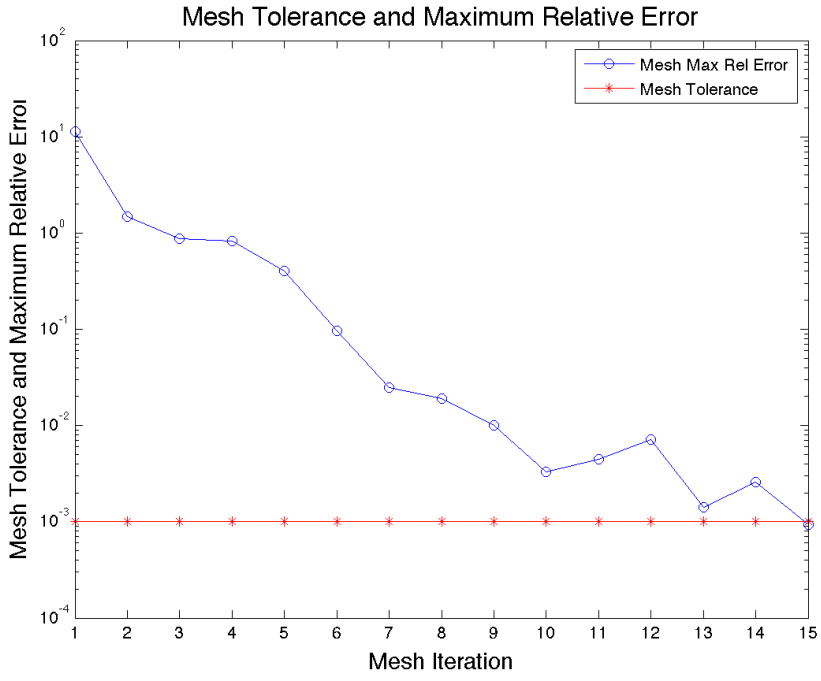

Stability Frame Forces
Forces (10^3 lbf)
2000
1000
0
A
T
T-A
N
0
500
1000
1500
2000
2500
3000
3500
4000
Time (sec)

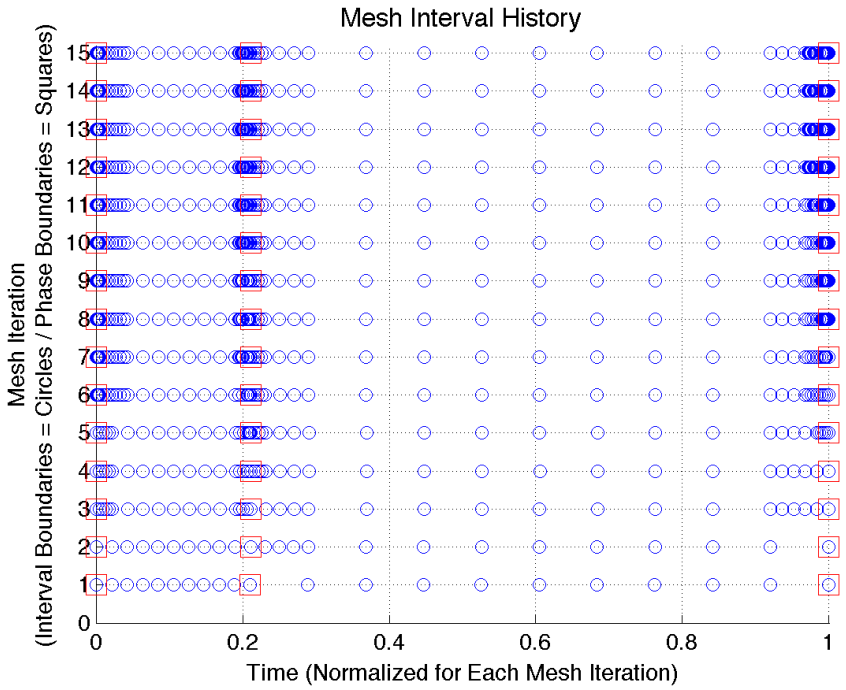

Vehicle Frame Forces
Forces (10^3 lbf)
2000
1000
0
-1000
L
-D
Tsin(α)+L
Tcos(α)-D
0
500
1000
1500
2000
2500
3000
3500
4000
Time (sec)

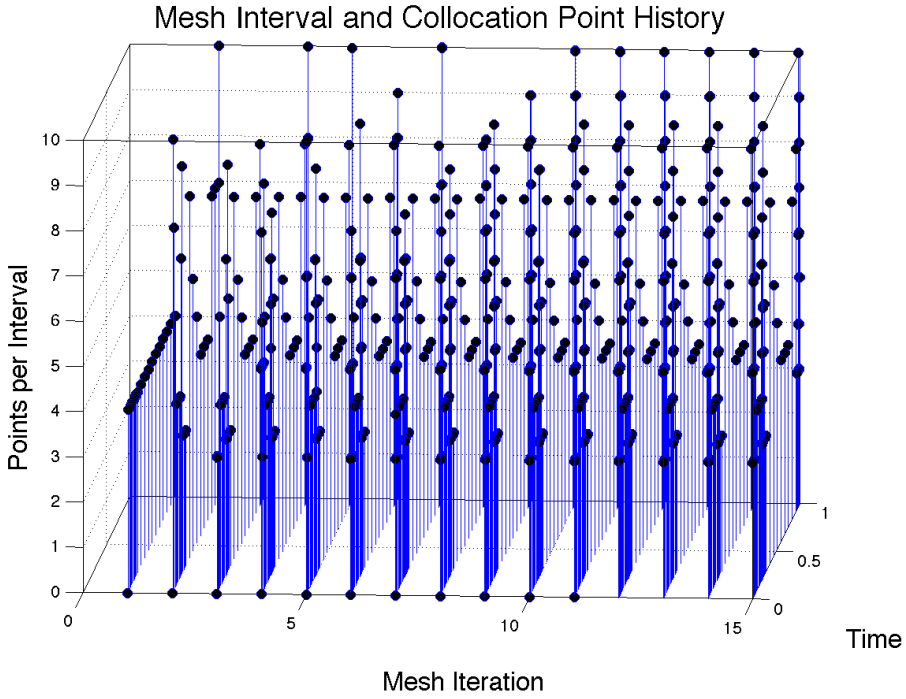

Heating Rate Profile
Nose MINIVER
Nose approx
Body MINIVER
Heating Rate (BTU/s-ft^2)
Time (sec)
140
120
100
80
60
40
20
0
-20
-40
0
500
1000
1500
2000
2500
3000
3500
4000
4500


Heating Load Profile
x 10^5
Nose MINIVER
Nose approx
Body MINIVER
Heating Load (BTU/ft^2)
Time (sec)
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0
500
1000
1500
2000
2500
3000
3500
4000
4500

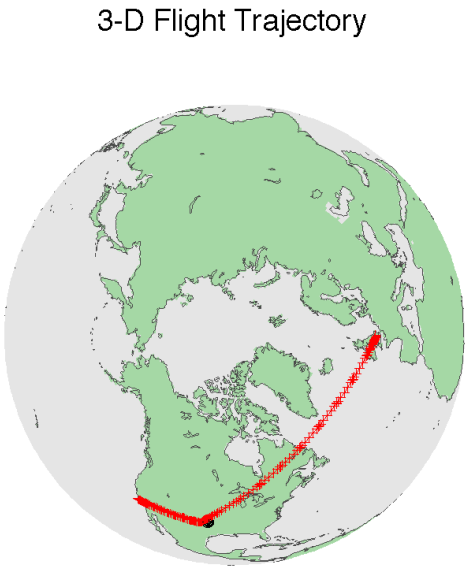

Temperature Profile
Temperature (deg F)
Time (sec)
Nose Outer Surface
Nose Inner Surface
Body Outer Surface
Body Inner Surface

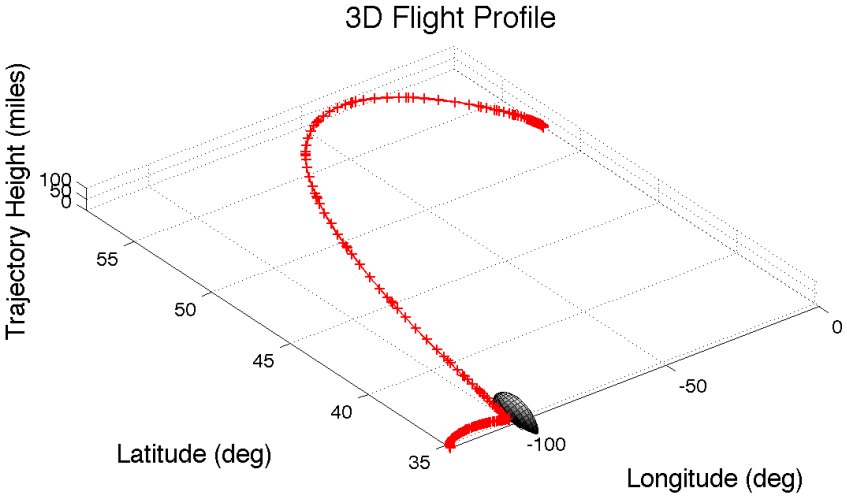

Mesh Tolerance and Maximum Relative Error
Mesh Tolerance and Maximum Relative Error
Mesh Iteration
Mesh Max Rel Error
Mesh Tolerance

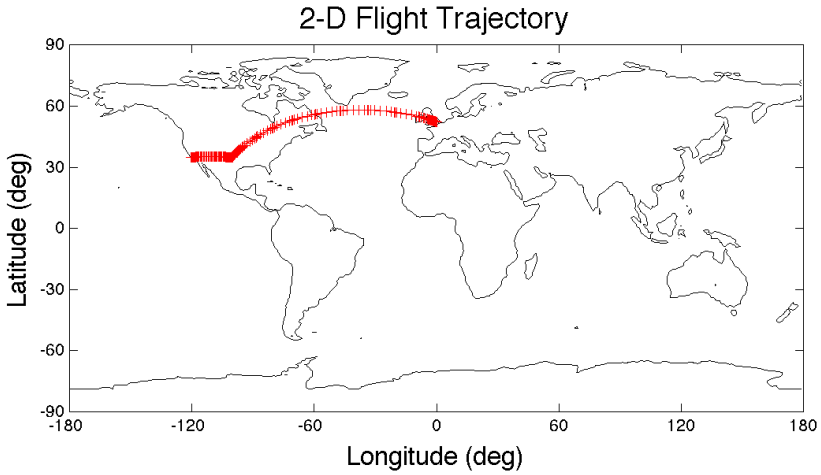

Mesh Interval History
(Interval Boundaries = Circles / Phase Boundaries = Squares)
Mesh Iteration
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
0
0.2
0.4
0.6
0.8
1
Time (Normalized for Each Mesh Iteration)

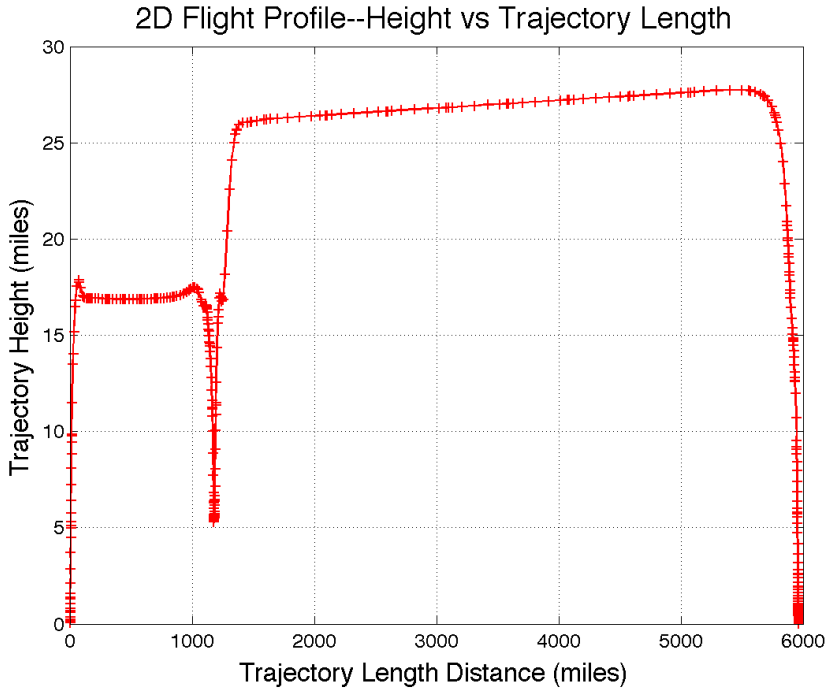

Mesh Interval and Collocation Point History
Points per Interval
0
1
2
3
4
5
6
7
8
9
10
0
5
10
15
Mesh Iteration
0
0.5
1
Time

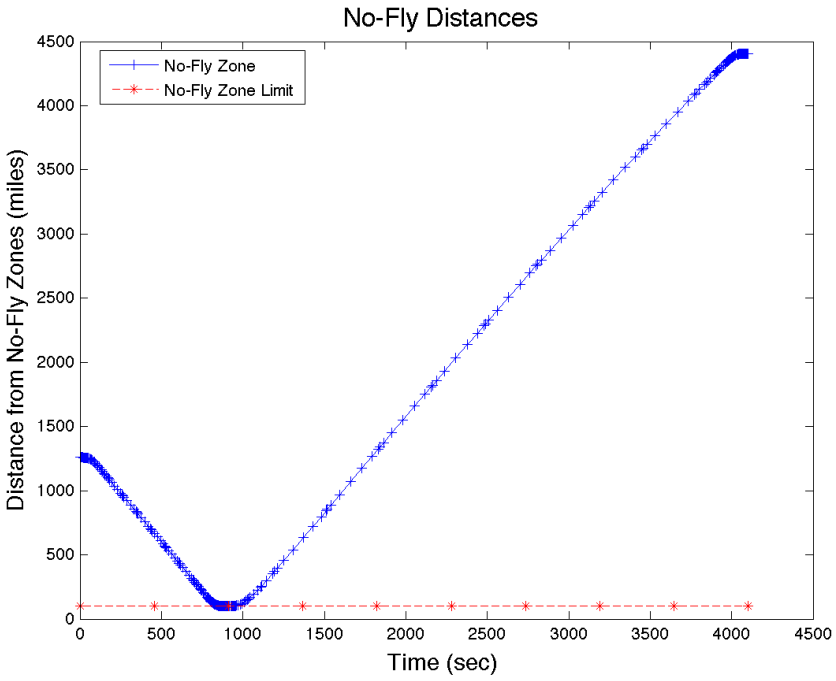
C.7 Minimum Time with Control Penalty, No-Fly Zone and g-limits; Climb, Cruise, Refuel, Cruise, and Land

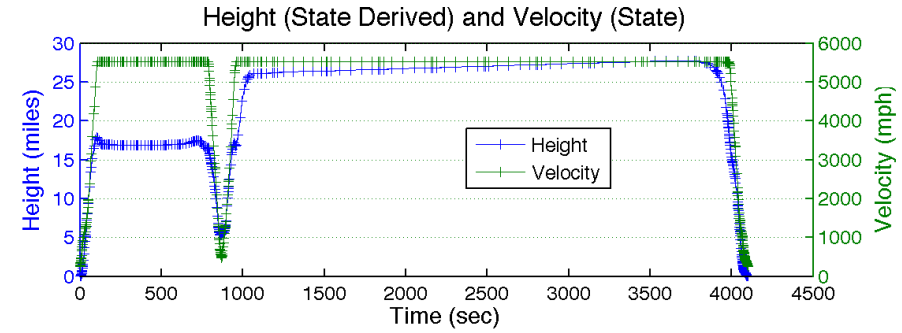


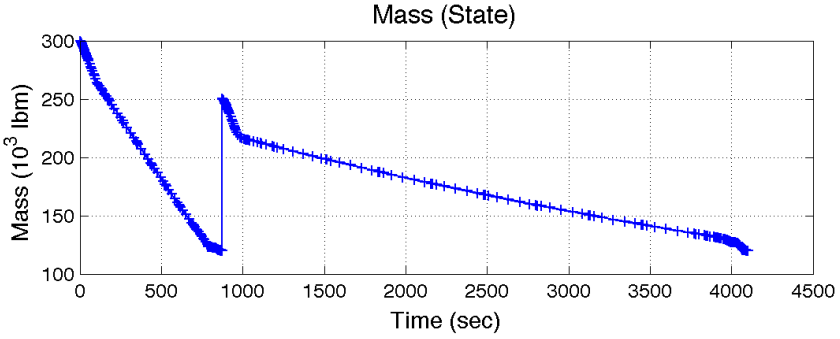


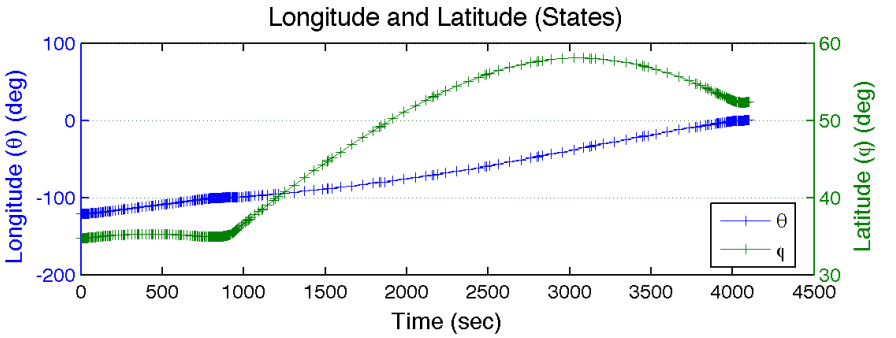

2-D Flight Trajectory
Latitude (deg)
Longitude (deg)
90
60
30
0
-30
-60
-90
-180
-120
-60
0
60
120
180

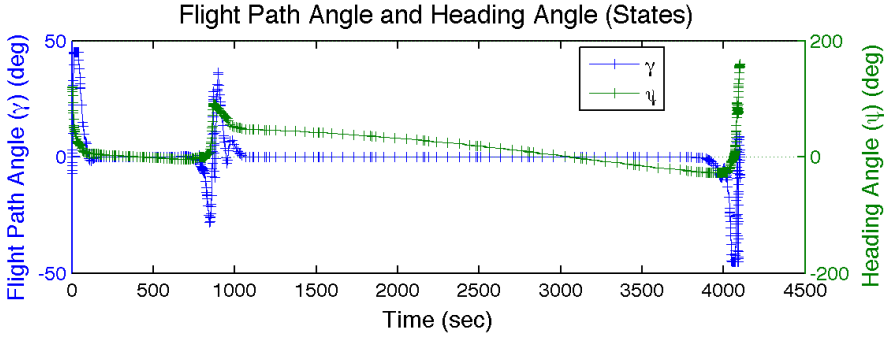

2D Flight Profile--Height vs Trajectory Length
Trajectory Height (miles)
Trajectory Length Distance (miles)
30
25
20
15
10
5
0
0
1000
2000
3000
4000
5000
6000

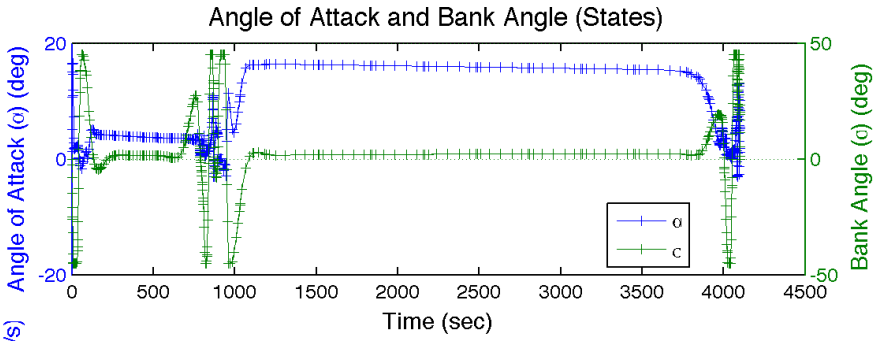

No-Fly Distances
No-Fly Zone
No-Fly Zone Limit
Distance from No-Fly Zones (miles)
Time (sec)
0
500
1000
1500
2000
2500
3000
3500
4000
4500

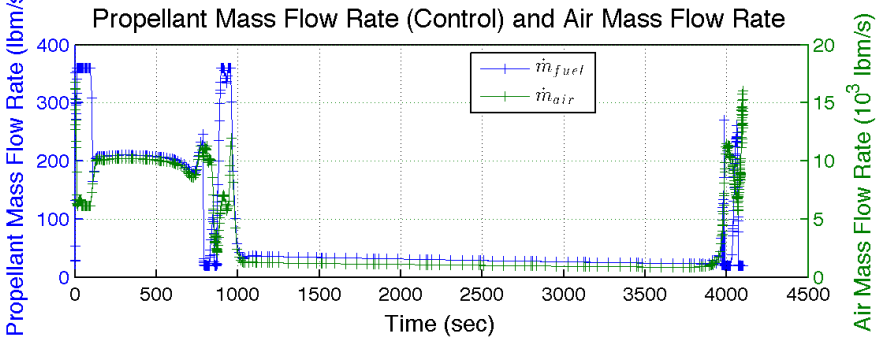

Height (State Derived) and Velocity (State)
Height
Velocity
Height (miles)
Velocity (mph)
Time (sec)
0
5
10
15
20
25
30
1000
2000
3000
4000
5000
6000

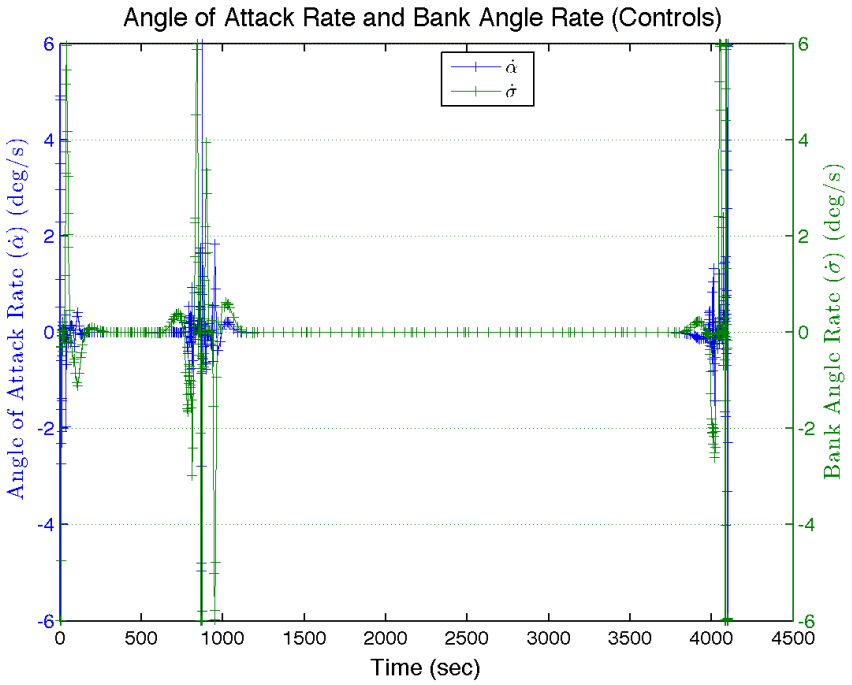

Mass (State)
Mass (10^3 lbm)
Time (sec)
100
150
200
250
300

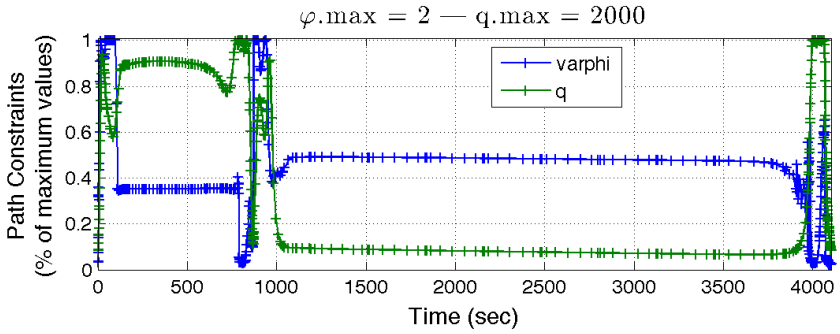

Longitude and Latitude (States)
Longitude (θ) (deg)
Latitude (φ) (deg)
Time (sec)
θ
φ

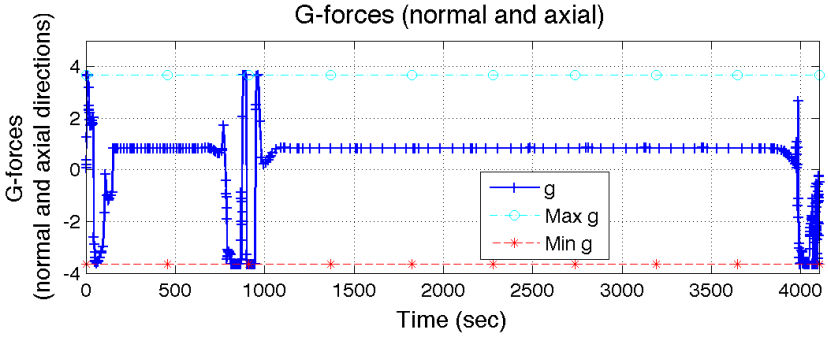

Flight Path Angle and Heading Angle (States)
Flight Path Angle (γ) (deg)
Heading Angle (ψ) (deg)
Time (sec)
γ
ψ

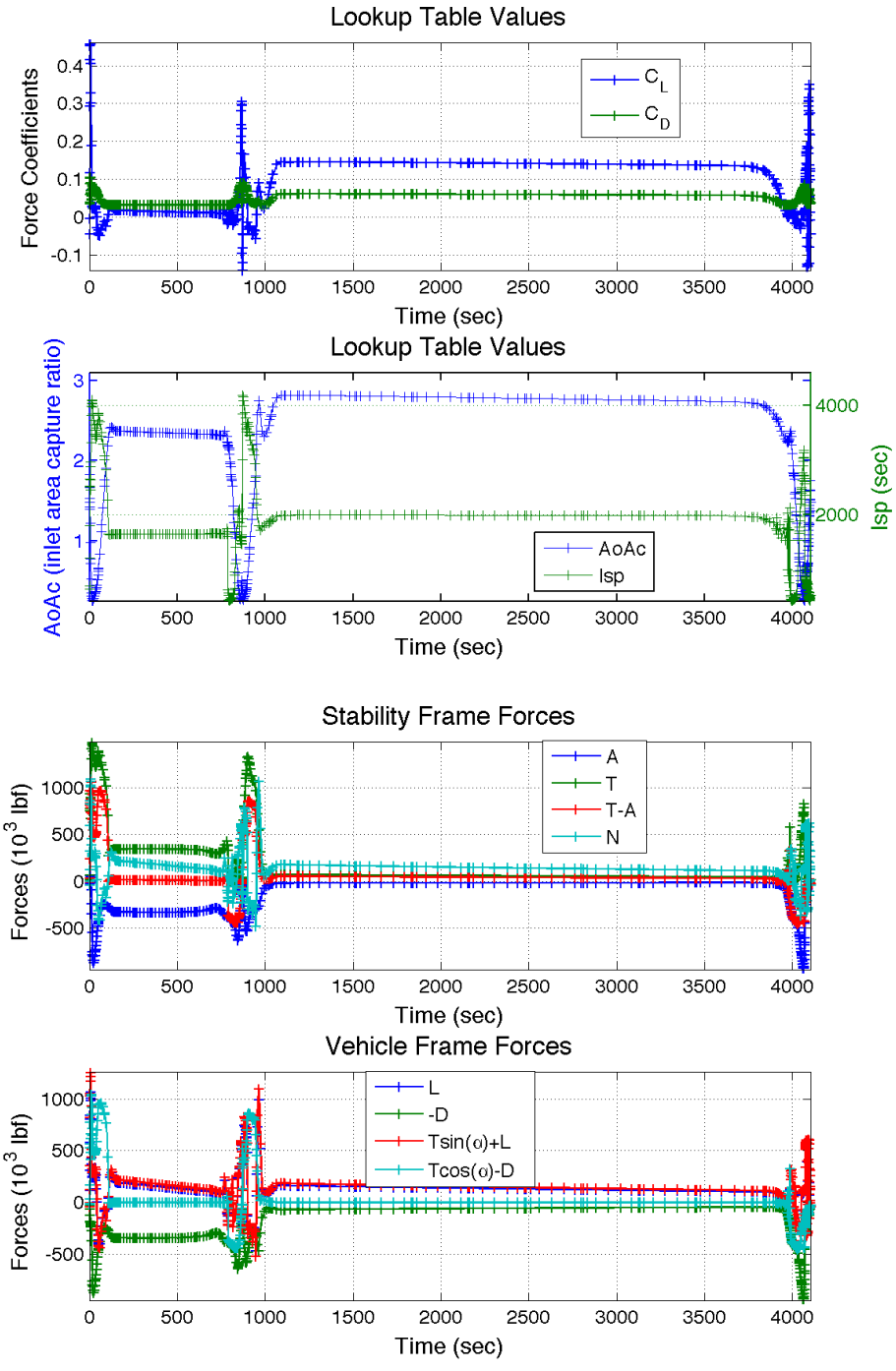

Angle of Attack and Bank Angle (States)
Angle of Attack (α) (deg)
Bank Angle (σ) (deg)
Time (sec)
α
σ

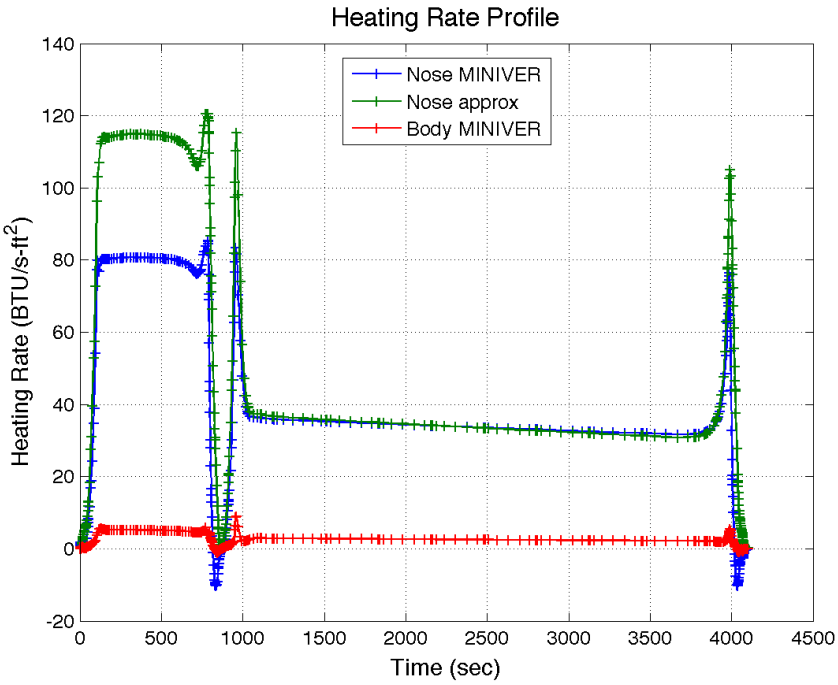

Propellant Mass Flow Rate (Control) and Air Mass Flow Rate
Propellant Mass Flow Rate (lbm/s)
Air Mass Flow Rate (10^3 lbm/s)
Time (sec)
$\dot{m}_{fuel}$
$\dot{m}_{air}$

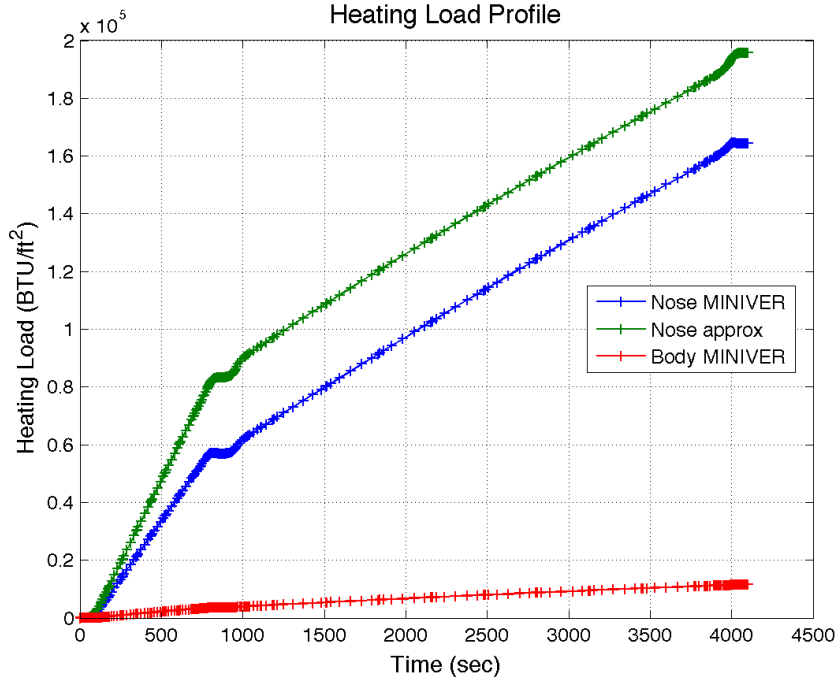

Angle of Attack Rate and Bank Angle Rate (Controls)
$\dot{\alpha}$
$\dot{\sigma}$
Angle of Attack Rate ($\dot{\alpha}$) (deg/s)
Bank Angle Rate ($\dot{\sigma}$) (deg/s)
Time (sec)

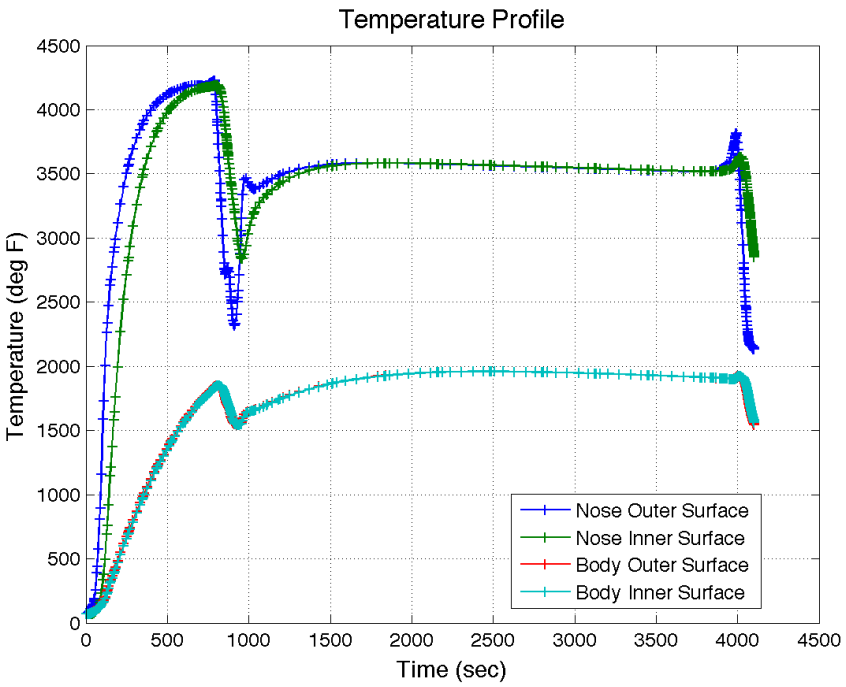

φ.max = 2 — q.max = 2000
varphi
q
Path Constraints
(% of maximum values)
Time (sec)

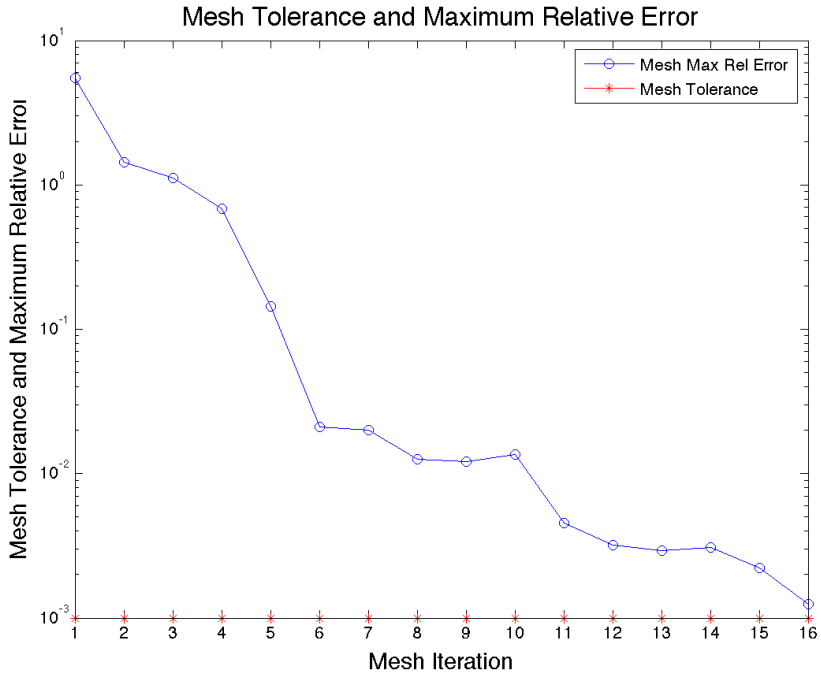

G-forces (normal and axial)
g
Max g
Min g
G-forces
(normal and axial directions)
Time (sec)

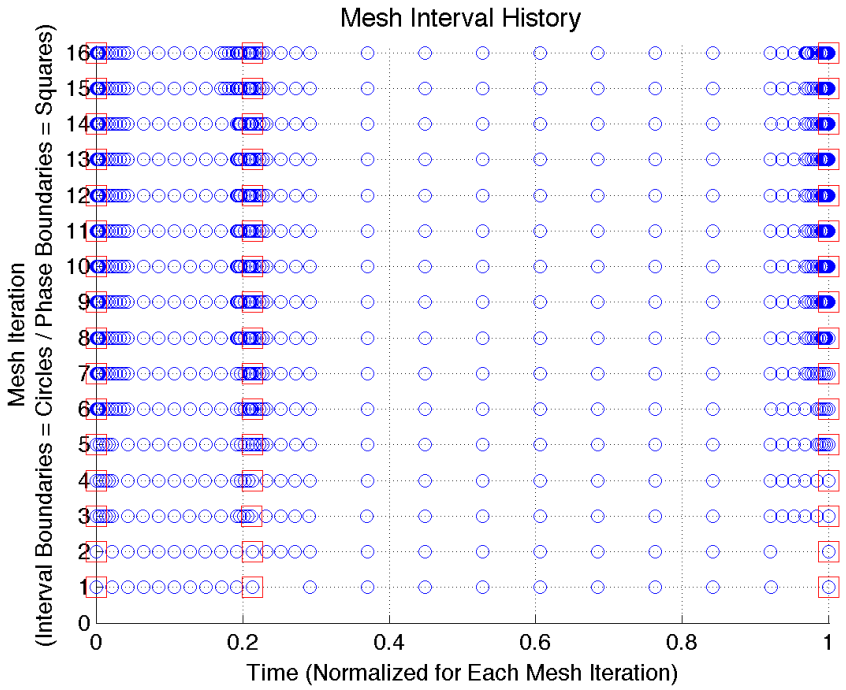

Lookup Table Values
Force Coefficients
Time (sec)
C_L
C_D
Lookup Table Values
AoAc (inlet area capture ratio)
Isp (sec)
Time (sec)
AoAc
Isp
Stability Frame Forces
Forces (10^3 lbf)
Time (sec)
A
T
T-A
N
Vehicle Frame Forces
Forces (10^3 lbf)
Time (sec)
L
-D
Tsin(α)+L
Tcos(α)-D

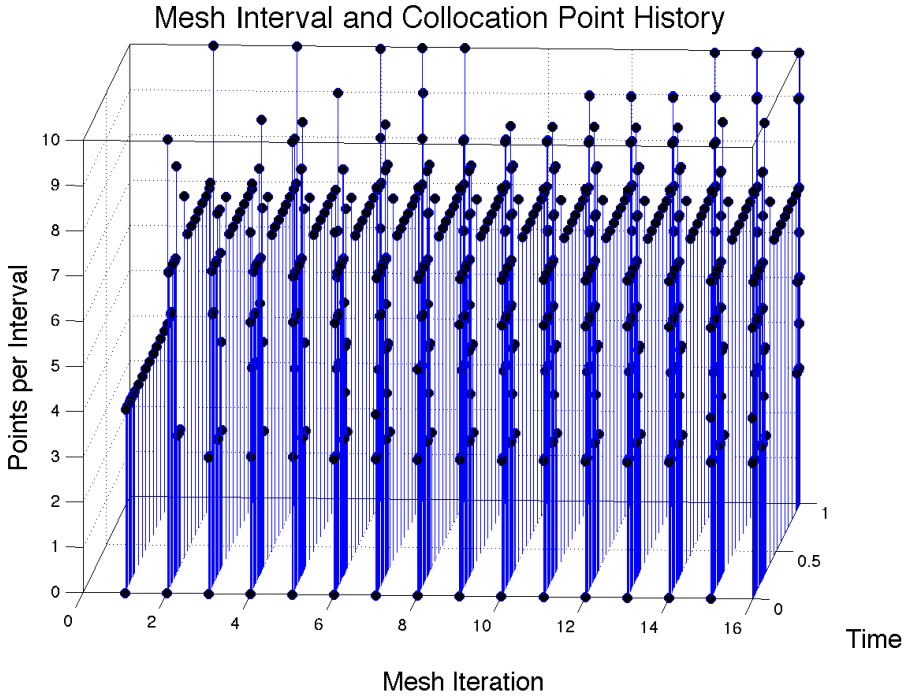

Heating Rate Profile
Nose MINIVER
Nose approx
Body MINIVER
Heating Rate (BTU/s-ft^2)
Time (sec)
140
120
100
80
60
40
20
0
-20
0
500
1000
1500
2000
2500
3000
3500
4000
4500


Heating Load Profile
x 10^5
Nose MINIVER
Nose approx
Body MINIVER
Heating Load (BTU/ft^2)
Time (sec)
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0
500
1000
1500
2000
2500
3000
3500
4000
4500

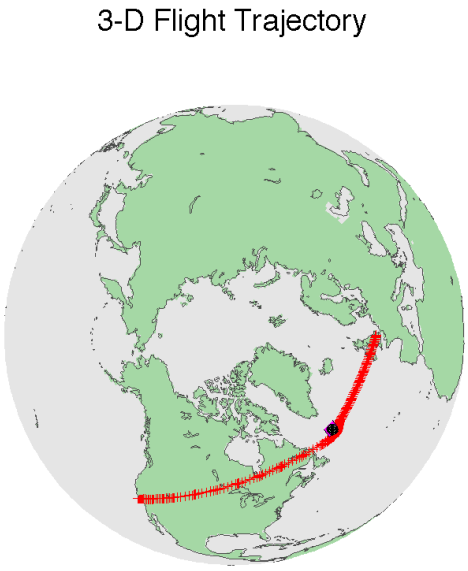

Temperature Profile
Temperature (deg F)
Time (sec)
Nose Outer Surface
Nose Inner Surface
Body Outer Surface
Body Inner Surface

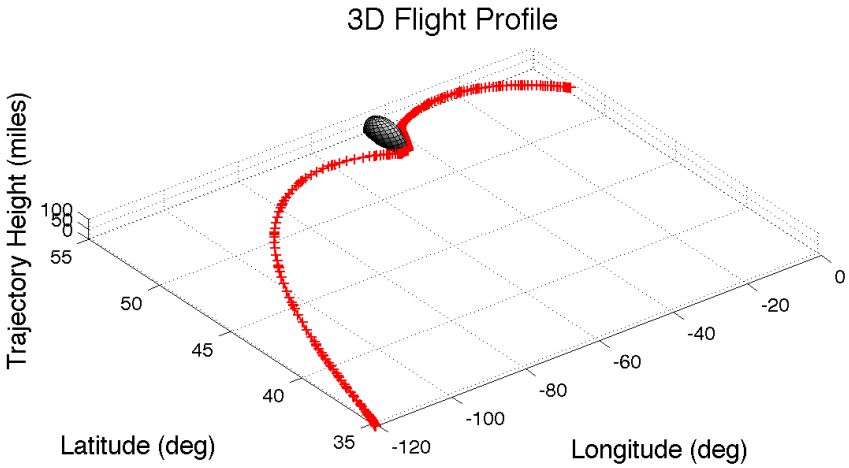

Mesh Tolerance and Maximum Relative Error
Mesh Tolerance and Maximum Relative Error
Mesh Iteration
Mesh Max Rel Error
Mesh Tolerance

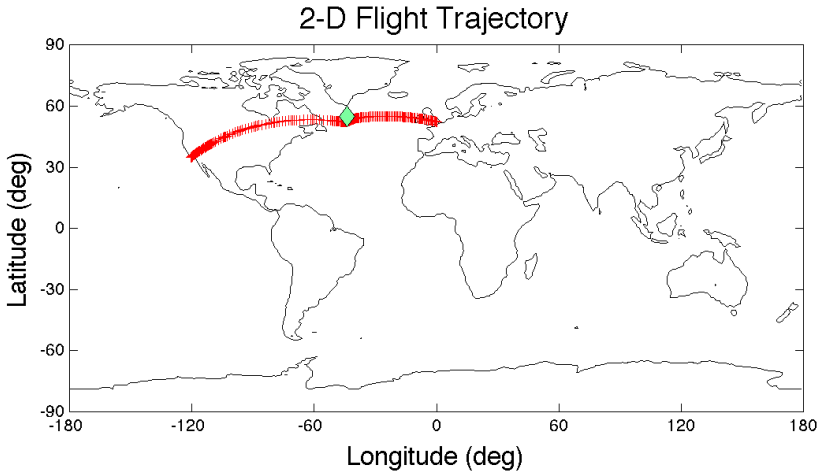

Mesh Interval History
(Interval Boundaries = Circles / Phase Boundaries = Squares)
Mesh Iteration
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
0
0.2
0.4
0.6
0.8
1
Time (Normalized for Each Mesh Iteration)

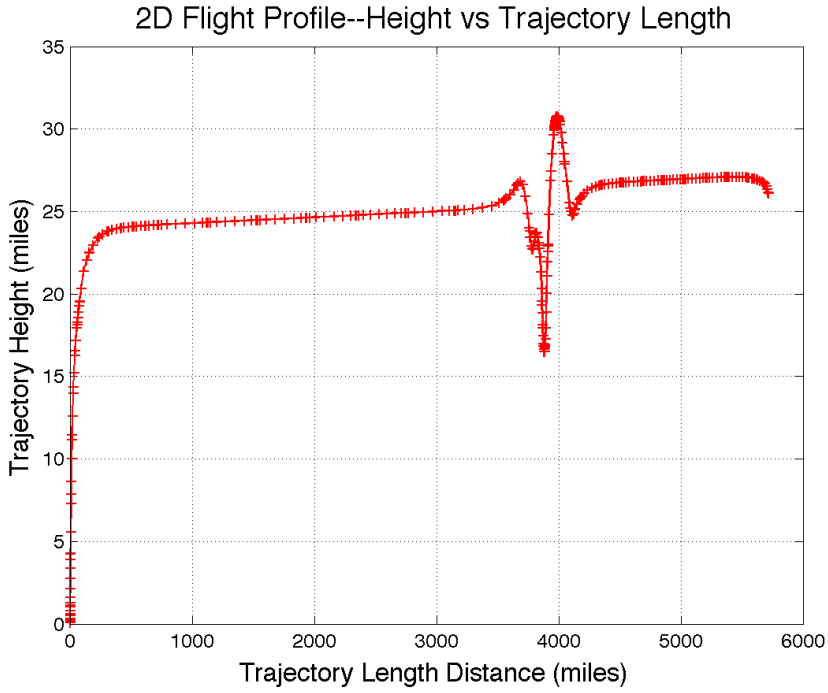

Mesh Interval and Collocation Point History
Points per Interval
0
1
2
3
4
5
6
7
8
9
10
0
2
4
6
8
10
12
14
16
Mesh Iteration
0
0.5
1
Time

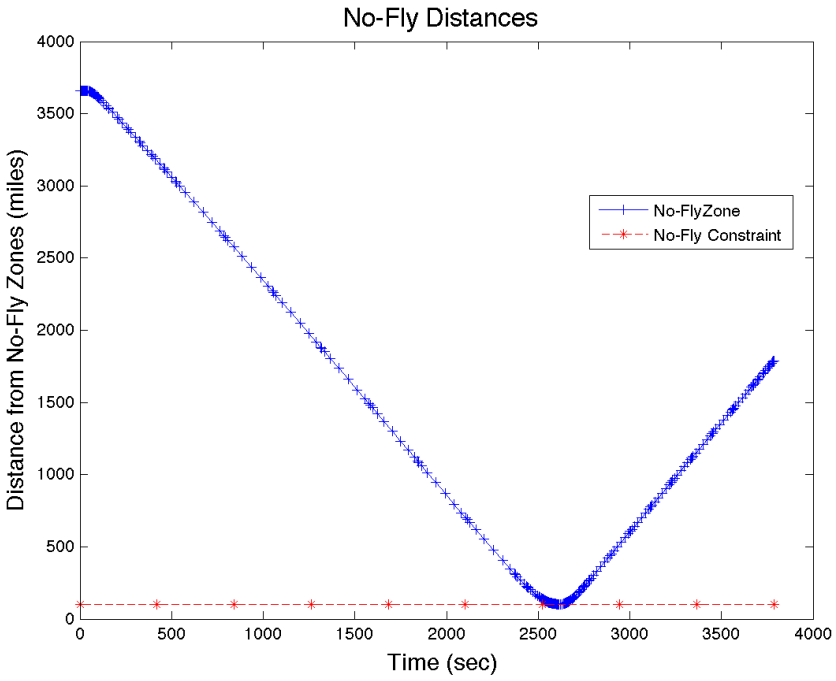
C.8 Minimum Time with Control Penalty, No-Fly Zone and Sensor Constraints; Climb, Cruise, Sense, and Cruise

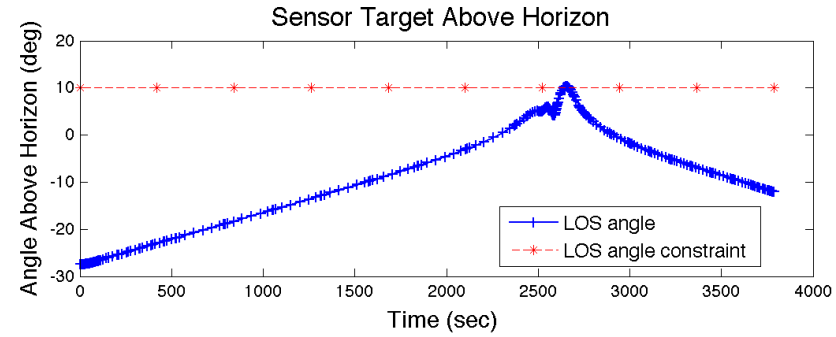


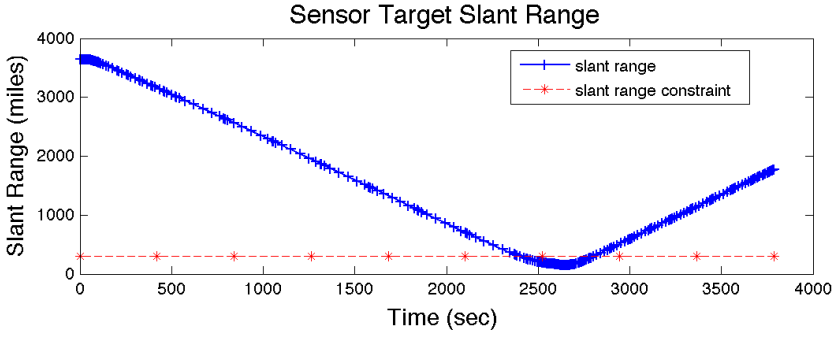


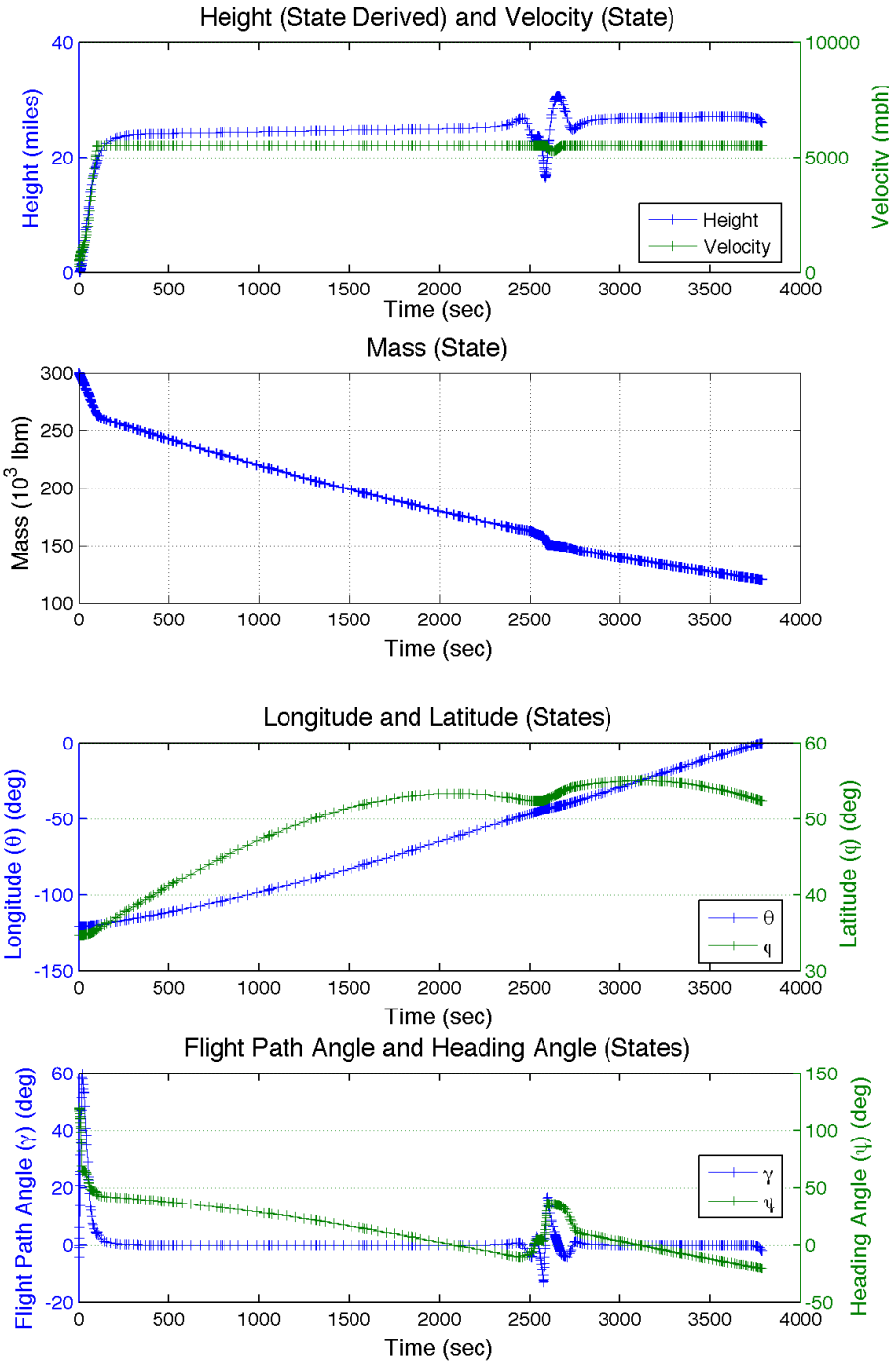

2-D Flight Trajectory
Latitude (deg)
Longitude (deg)
90
60
30
0
-30
-60
-90
-180
-120
-60
0
60
120
180

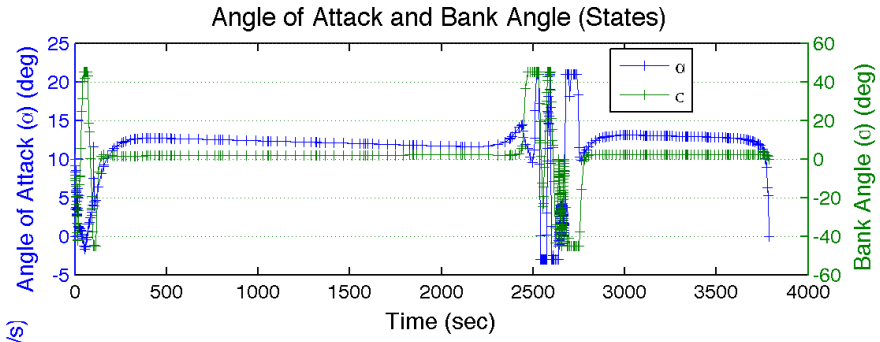

2D Flight Profile--Height vs Trajectory Length
Trajectory Height (miles)
Trajectory Length Distance (miles)
35
30
25
20
15
10
5
0
0
1000
2000
3000
4000
5000
6000

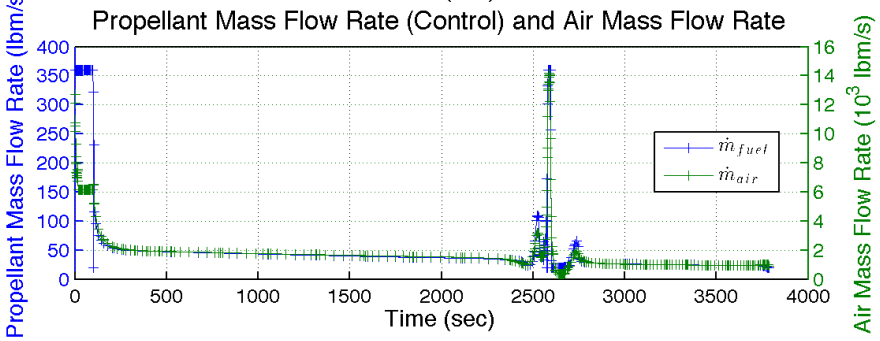

No-Fly Distances
Distance from No-Fly Zones (miles)
Time (sec)
No-FlyZone
No-Fly Constraint
0
500
1000
1500
2000
2500
3000
3500
4000

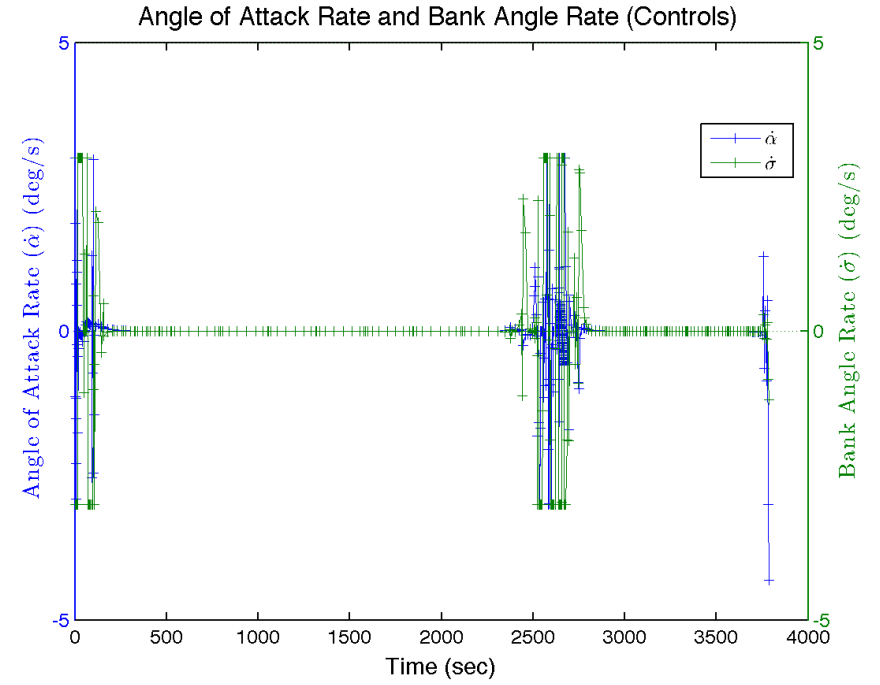

Sensor Target Above Horizon
Angle Above Horizon (deg)
Time (sec)
LOS angle
LOS angle constraint
20
10
0
-10
-20
-30
0
500
1000
1500
2000
2500
3000
3500
4000

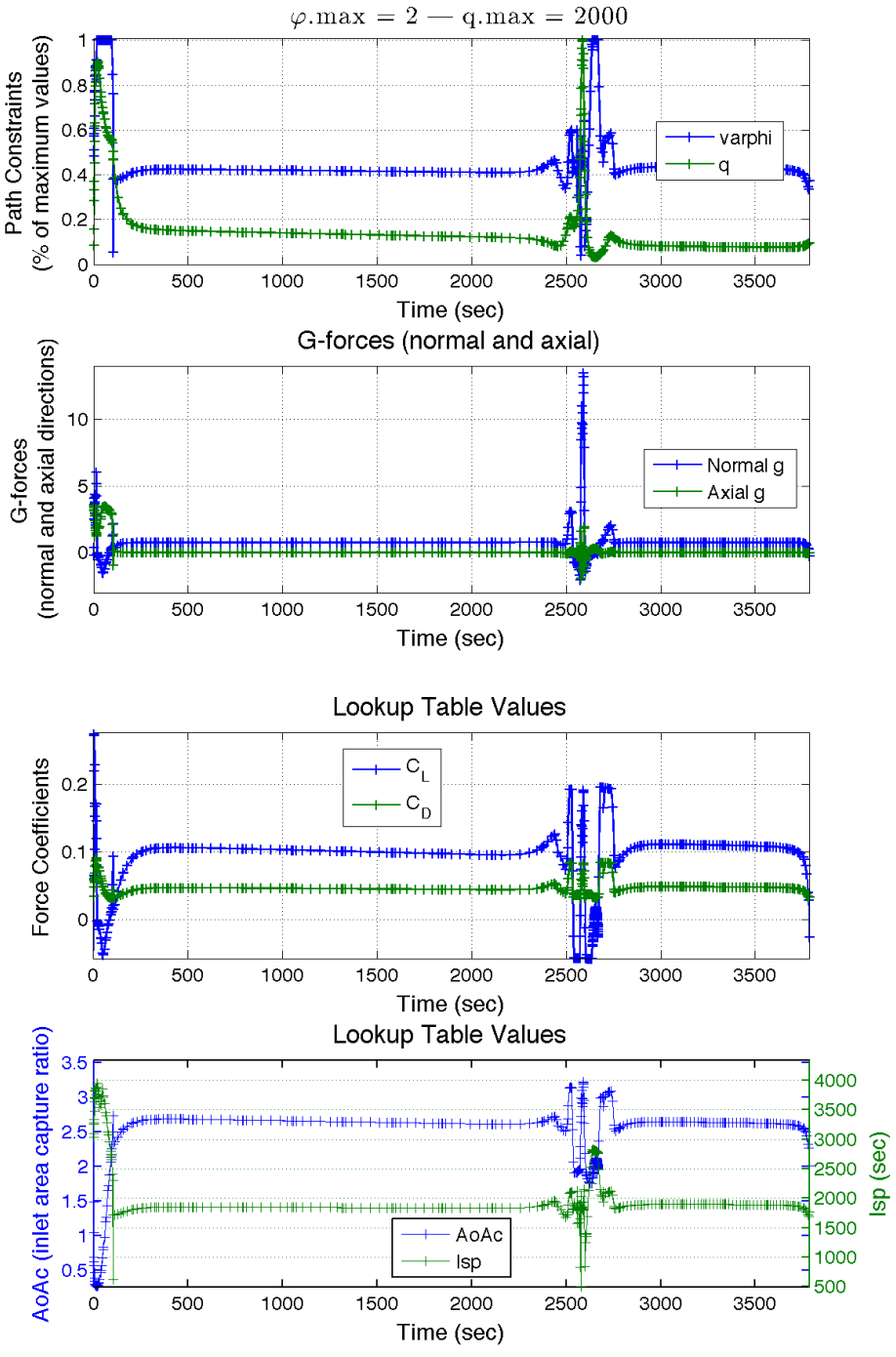

Sensor Target Slant Range
Slant Range (miles)
Time (sec)
slant range
slant range constraint
0
1000
2000
3000
4000
0
500
1000
1500
2000
2500
3000
3500
4000

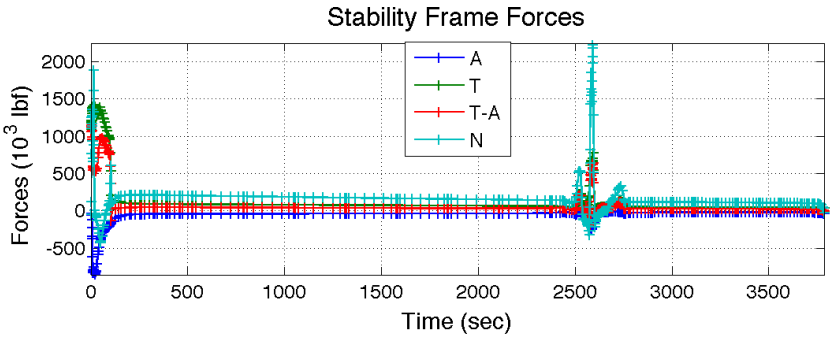

Height (State Derived) and Velocity (State)
Height (miles)
Velocity (mph)
Height
Velocity
Time (sec)
Mass (State)
Mass (10^3 lbm)
Time (sec)
Longitude and Latitude (States)
Longitude (θ) (deg)
Latitude (φ) (deg)
θ
φ
Time (sec)
Flight Path Angle and Heading Angle (States)
Flight Path Angle (γ) (deg)
Heading Angle (ψ) (deg)
γ
ψ
Time (sec)

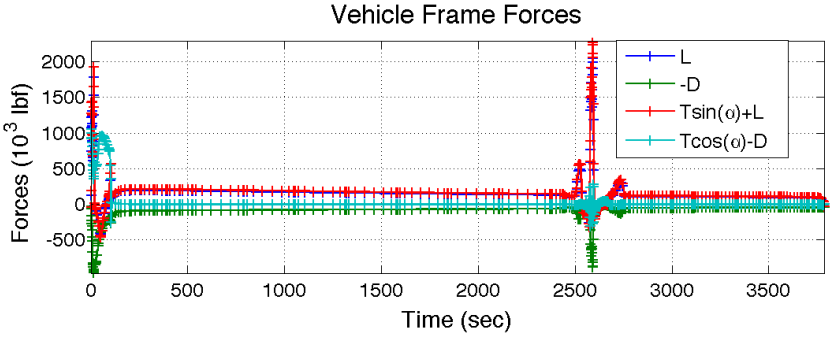

Angle of Attack and Bank Angle (States)
Angle of Attack (α) (deg)
Bank Angle (σ) (deg)
Time (sec)
α
c

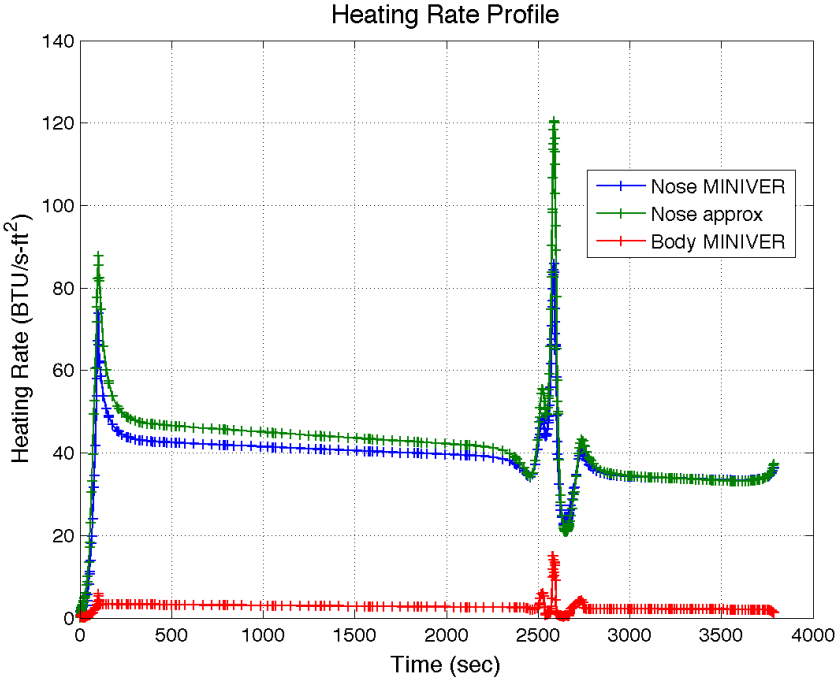

Propellant Mass Flow Rate (Control) and Air Mass Flow Rate
Propellant Mass Flow Rate (lbm/s)
Air Mass Flow Rate (10^3 lbm/s)
Time (sec)
$\dot{m}_{fuel}$
$\dot{m}_{air}$

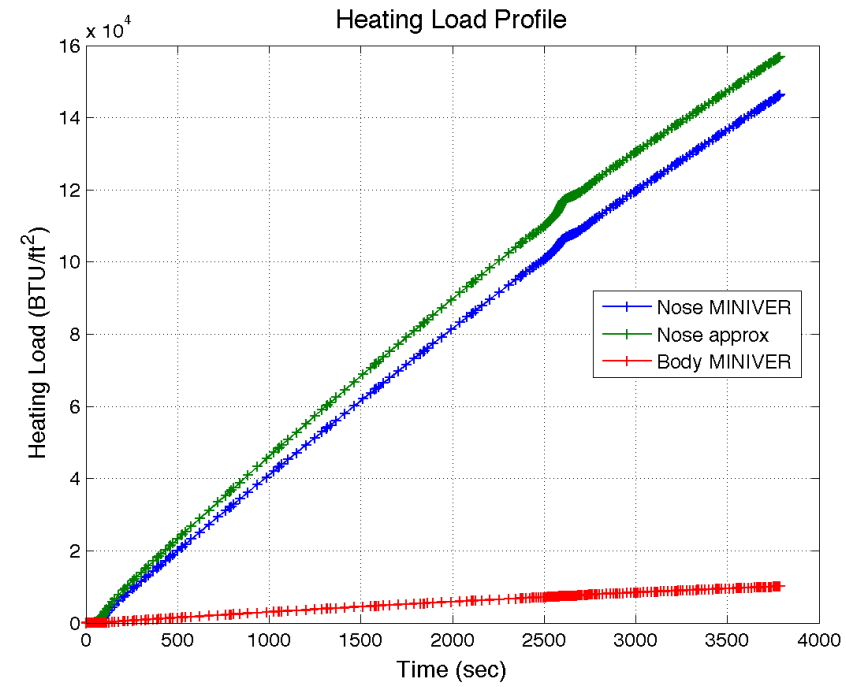

Angle of Attack Rate and Bank Angle Rate (Controls)
Angle of Attack Rate ($\dot{\alpha}$) (deg/s)
Bank Angle Rate ($\dot{\sigma}$) (deg/s)
Time (sec)
$\dot{\alpha}$
$\dot{\sigma}$

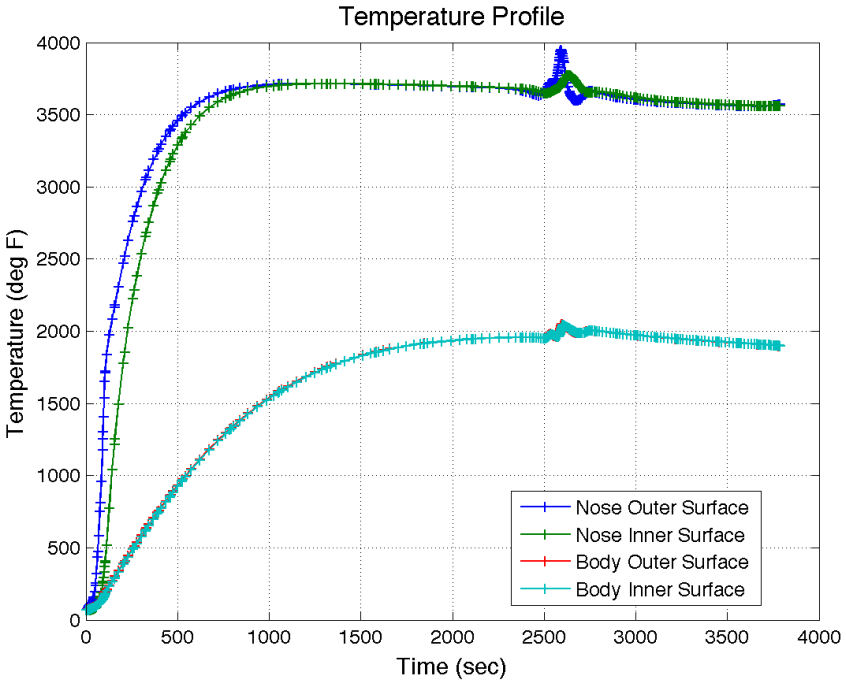

φ.max = 2 — q.max = 2000
Path Constraints
(% of maximum values)
varphi
q
Time (sec)
G-forces (normal and axial)
G-forces
(normal and axial directions)
Normal g
Axial g
Time (sec)
Lookup Table Values
Force Coefficients
C_L
C_D
Time (sec)
Lookup Table Values
AoAc (inlet area capture ratio)
Isp (sec)
AoAc
Isp
Time (sec)

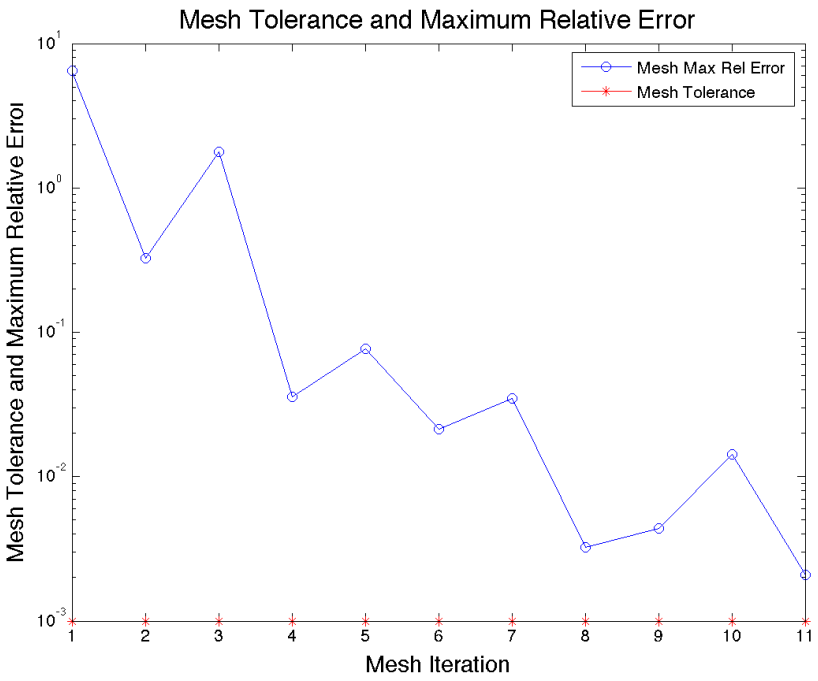

Stability Frame Forces
A
T
T-A
N
Forces (10^3 lbf)
Time (sec)

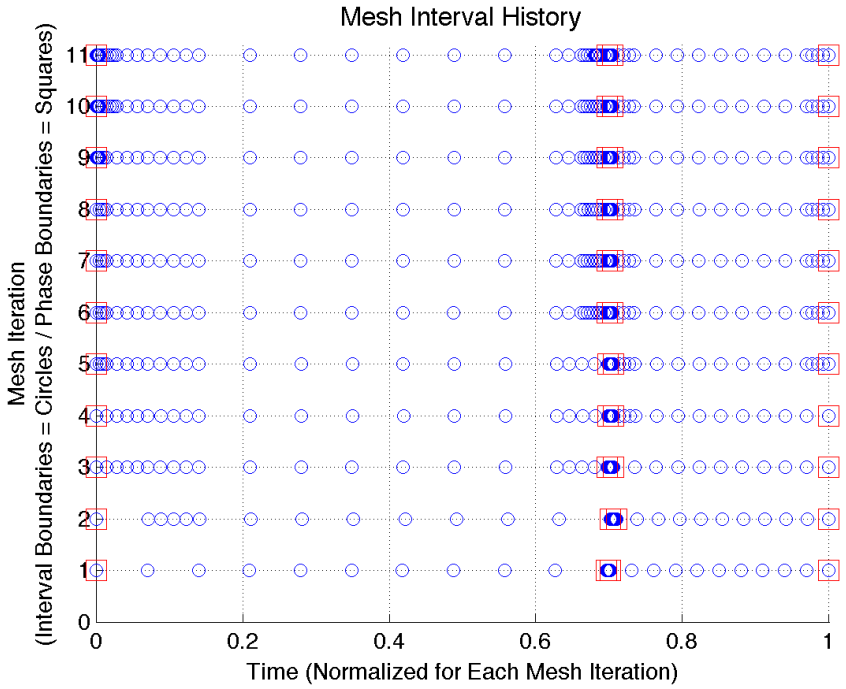

Vehicle Frame Forces
L
-D
Tsin(α)+L
Tcos(α)-D
Forces (10^3 lbf)
Time (sec)

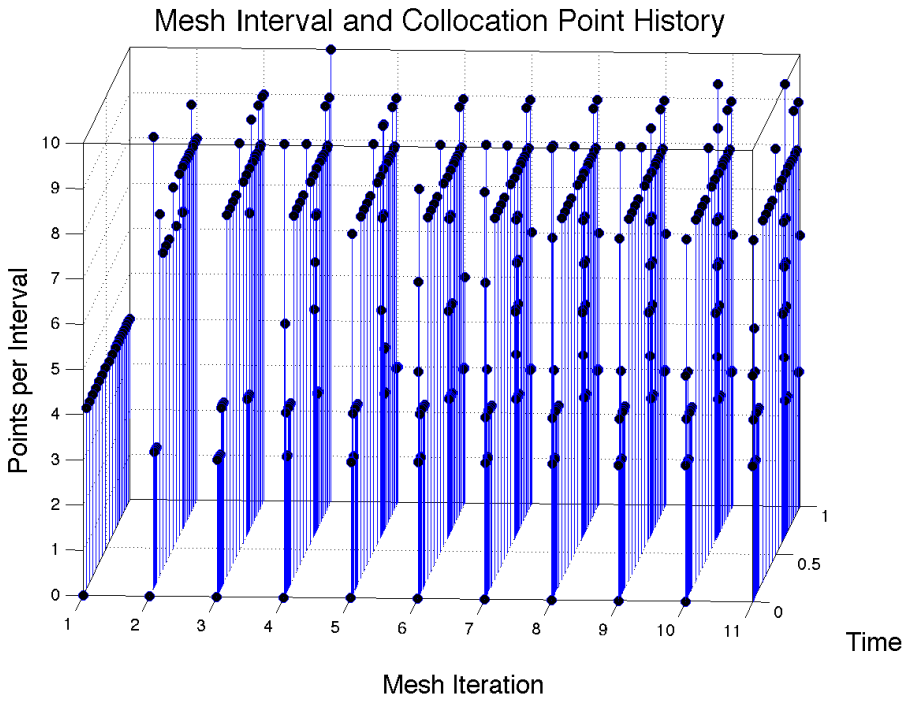

Heating Rate Profile
Nose MINIVER
Nose approx
Body MINIVER
Heating Rate (BTU/s-ft^2)
Time (sec)


Heating Load Profile
x 10^4
Heating Load (BTU/ft^2)
Time (sec)
Nose MINIVER
Nose approx
Body MINIVER


Temperature Profile
Temperature (deg F)
Time (sec)
Nose Outer Surface
Nose Inner Surface
Body Outer Surface
Body Inner Surface

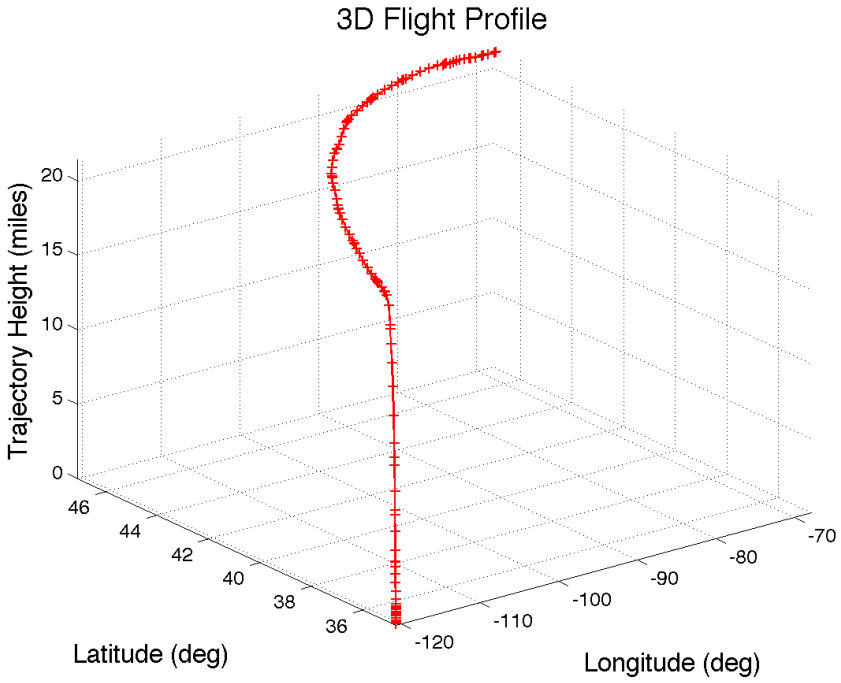

Mesh Tolerance and Maximum Relative Error
Mesh Max Rel Error
Mesh Tolerance
Mesh Tolerance and Maximum Relative Error
Mesh Iteration

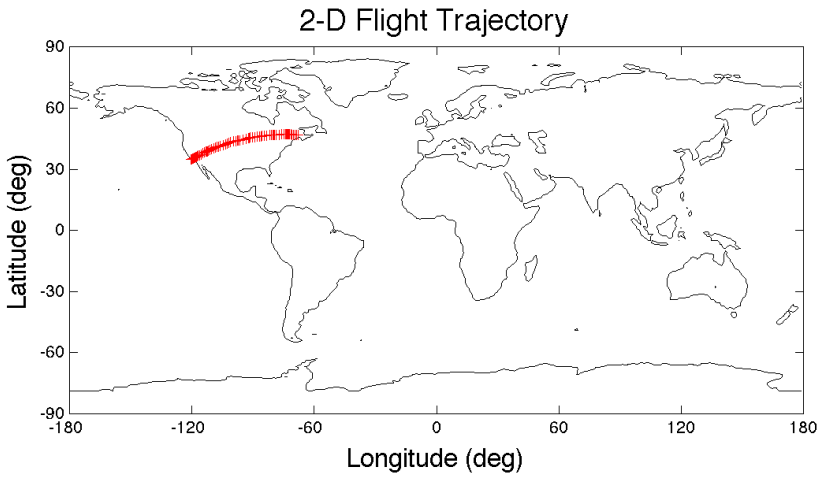

Mesh Interval History
(Interval Boundaries = Circles / Phase Boundaries = Squares)
Mesh Iteration
Time (Normalized for Each Mesh Iteration)

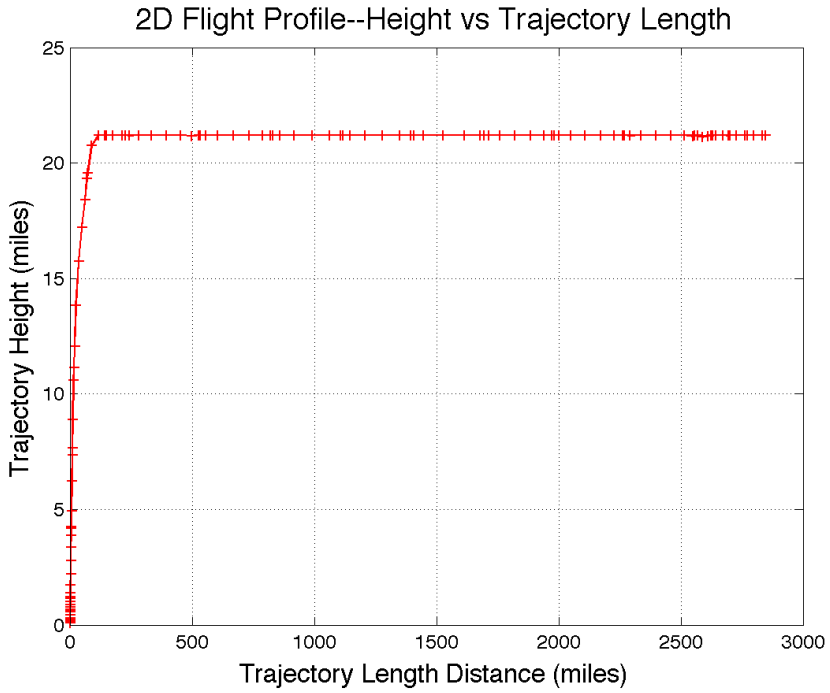


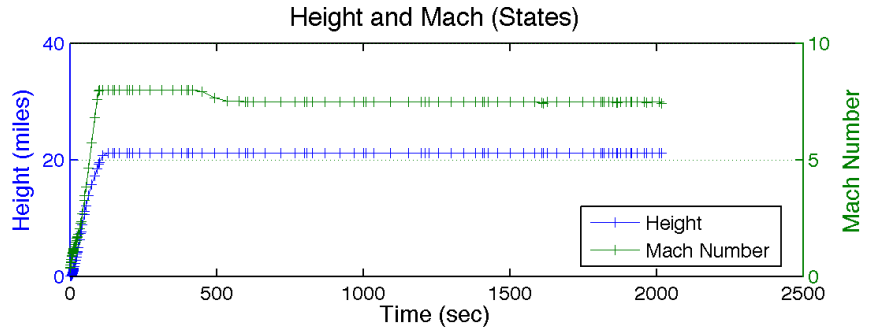
C.9 Minimum Time with Temperature Constraints; Climb and Cruise

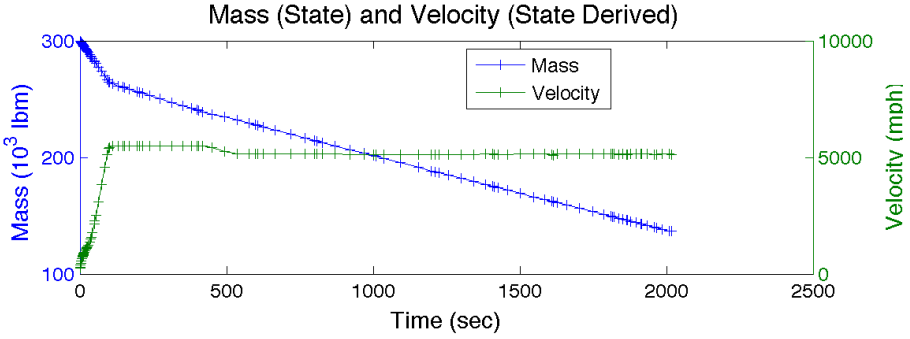


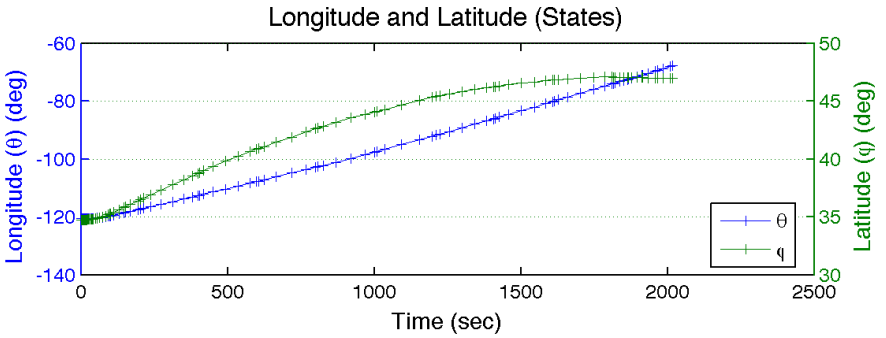

3D Flight Profile
Trajectory Height (miles)
20
15
10
5
0
46
44
42
40
38
36
Latitude (deg)
-70
-80
-90
-100
-110
-120
Longitude (deg)

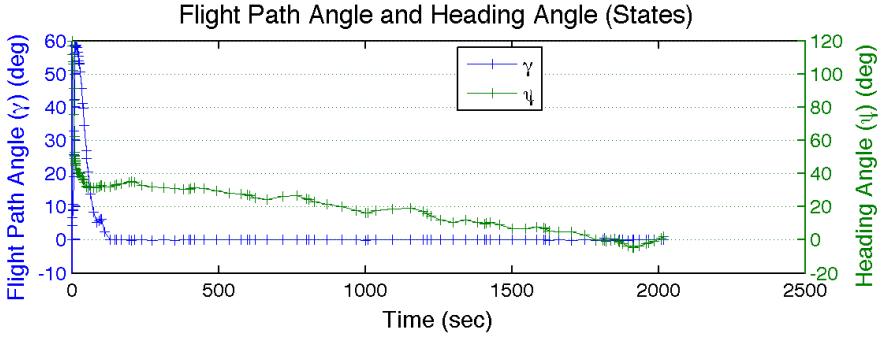

2-D Flight Trajectory
Latitude (deg)
90
60
30
0
-30
-60
-90
-180
-120
-60
0
60
120
180
Longitude (deg)

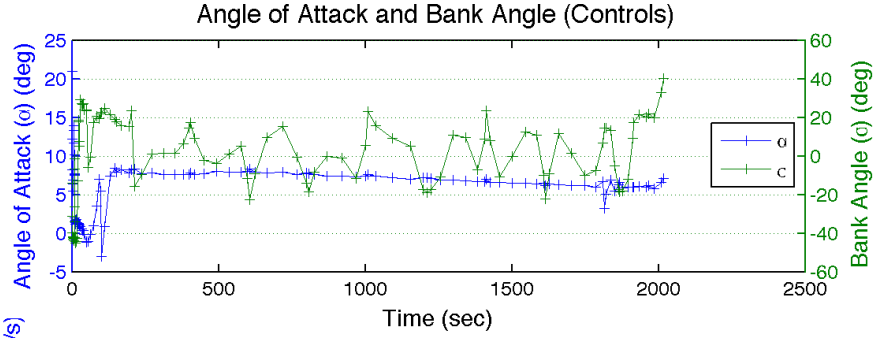

2D Flight Profile--Height vs Trajectory Length
Trajectory Height (miles)
Trajectory Length Distance (miles)
0
5
10
15
20
25
0
500
1000
1500
2000
2500
3000

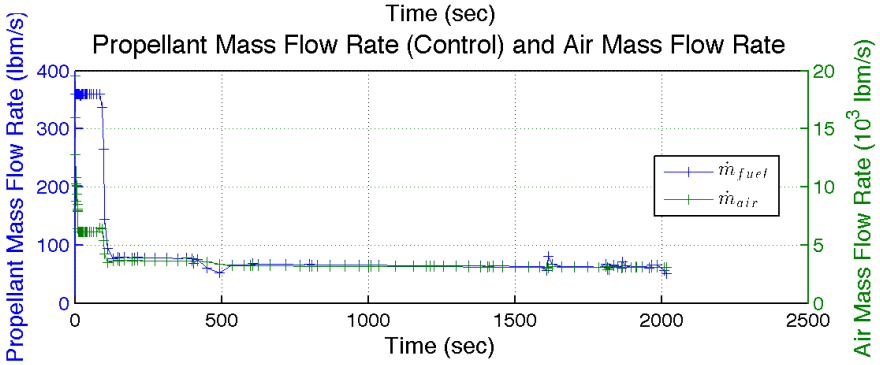

Height and Mach (States)
Height (miles)
Mach Number
Time (sec)
Height
Mach Number
0
20
40
0
5
10
0
500
1000
1500
2000
2500

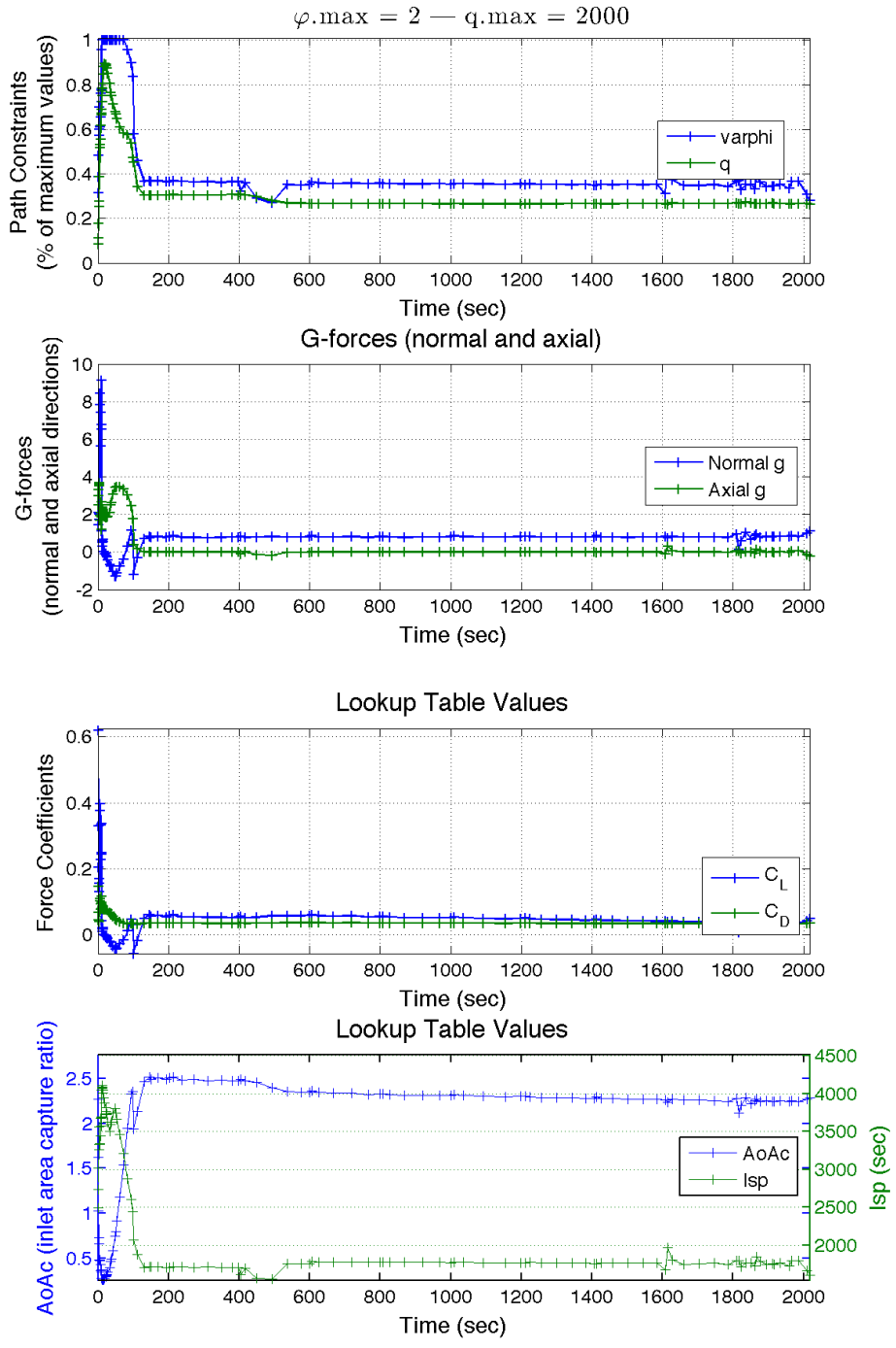

Mass (State) and Velocity (State Derived)
Mass (10^3 lbm)
Velocity (mph)
Time (sec)
Mass
Velocity
100
200
300
0
5000
10000
0
500
1000
1500
2000
2500

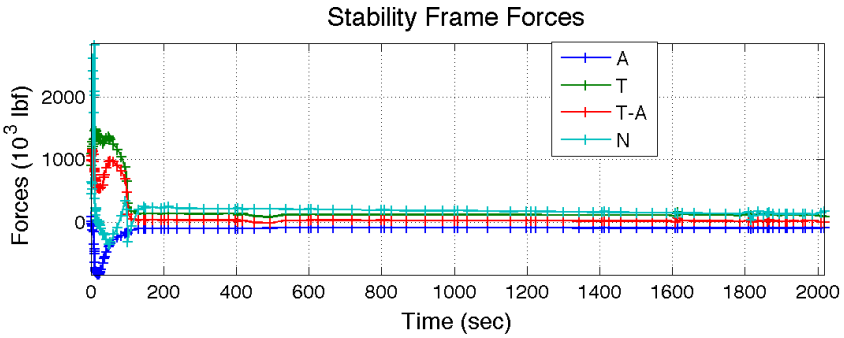

Longitude and Latitude (States)
Longitude (θ) (deg)
Latitude (φ) (deg)
Time (sec)
θ
φ

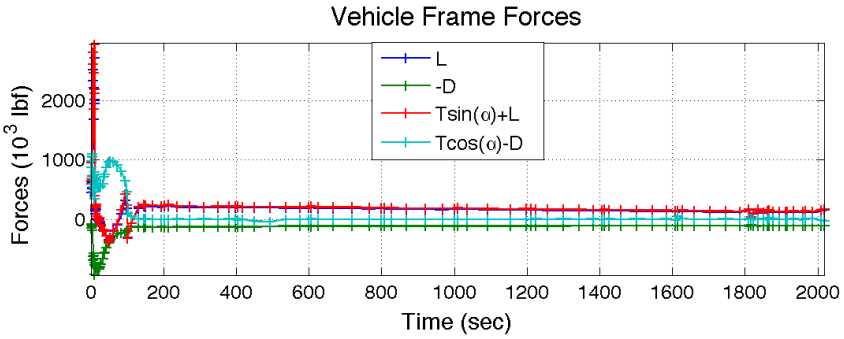

Flight Path Angle and Heading Angle (States)
Flight Path Angle (γ) (deg)
Heading Angle (ψ) (deg)
Time (sec)
γ
ψ

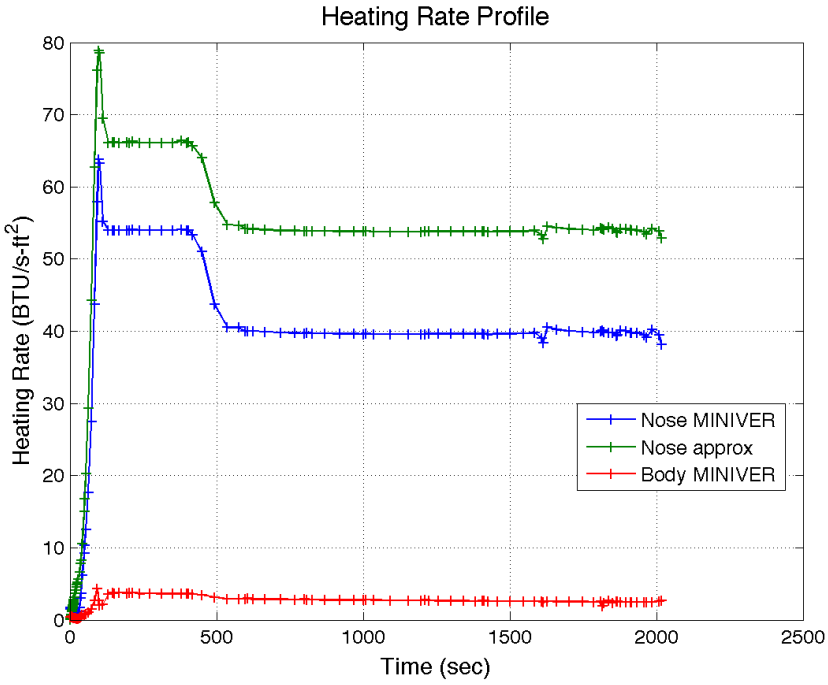

Angle of Attack and Bank Angle (Controls)
Angle of Attack (α) (deg)
Bank Angle (σ) (deg)
Time (sec)
α
σ

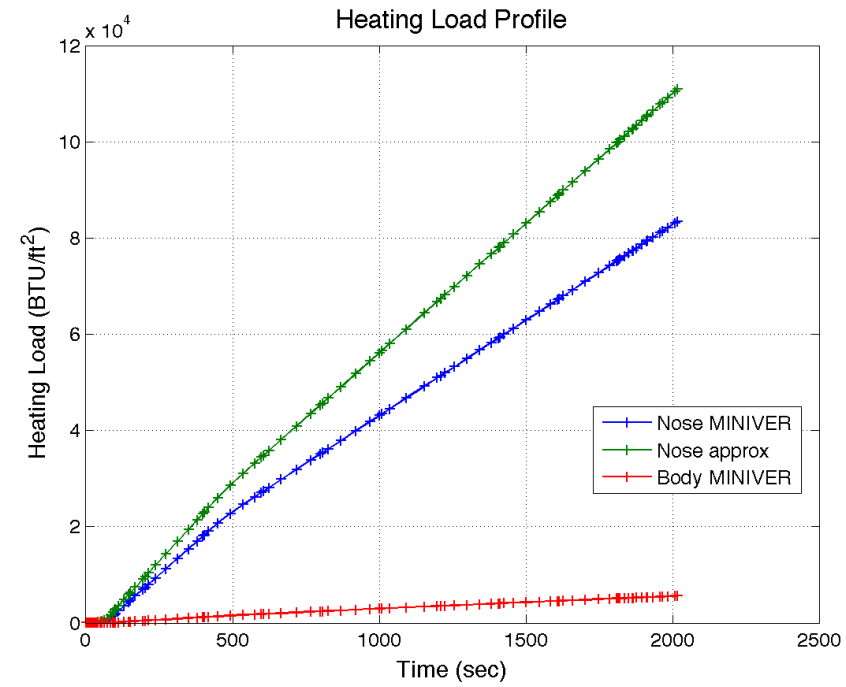

Propellant Mass Flow Rate (Control) and Air Mass Flow Rate
Propellant Mass Flow Rate (lbm/s)
Air Mass Flow Rate (10^3 lbm/s)
Time (sec)
$\dot{m}_{fuel}$
$\dot{m}_{air}$

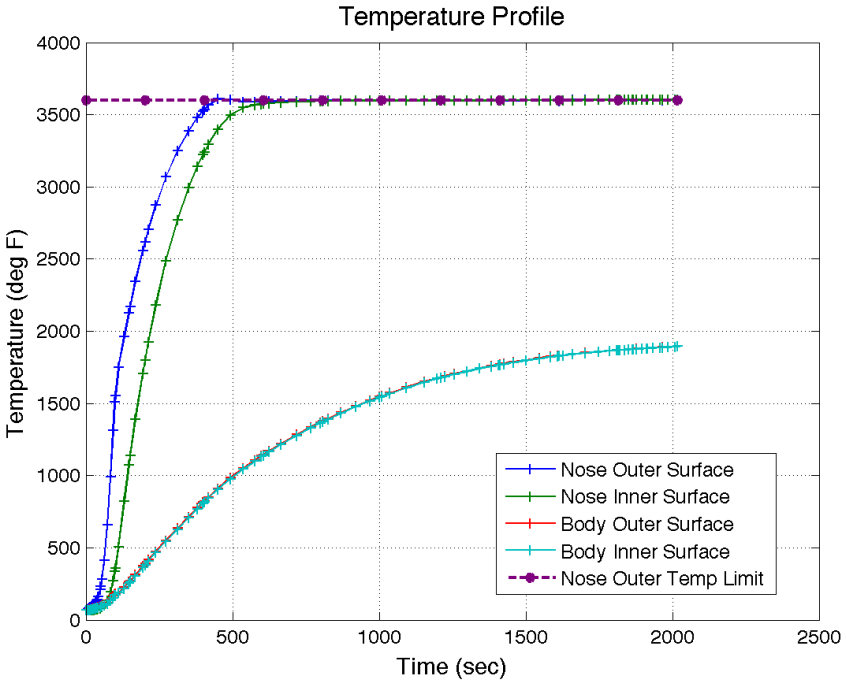

φ.max = 2 — q.max = 2000
Path Constraints
(% of maximum values)
varphi
q
Time (sec)
G-forces (normal and axial)
G-forces
(normal and axial directions)
Normal g
Axial g
Time (sec)
Lookup Table Values
Force Coefficients
C_L
C_D
Time (sec)
Lookup Table Values
AoAc (inlet area capture ratio)
Isp (sec)
AoAc
Isp
Time (sec)

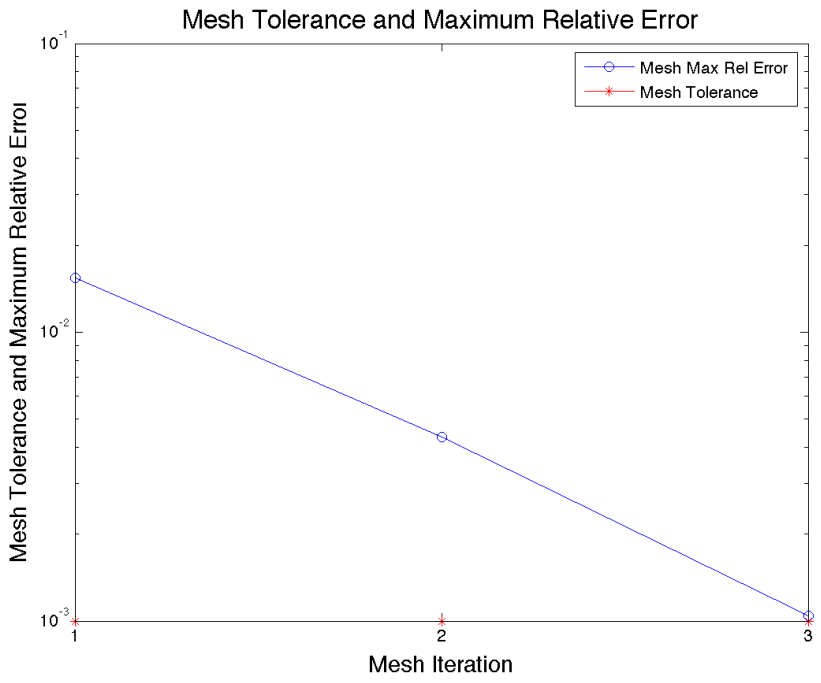

Stability Frame Forces
A
T
T-A
N
Forces (10^3 lbf)
2000
1000
0
0
200
400
600
800
1000
1200
1400
1600
1800
2000
Time (sec)

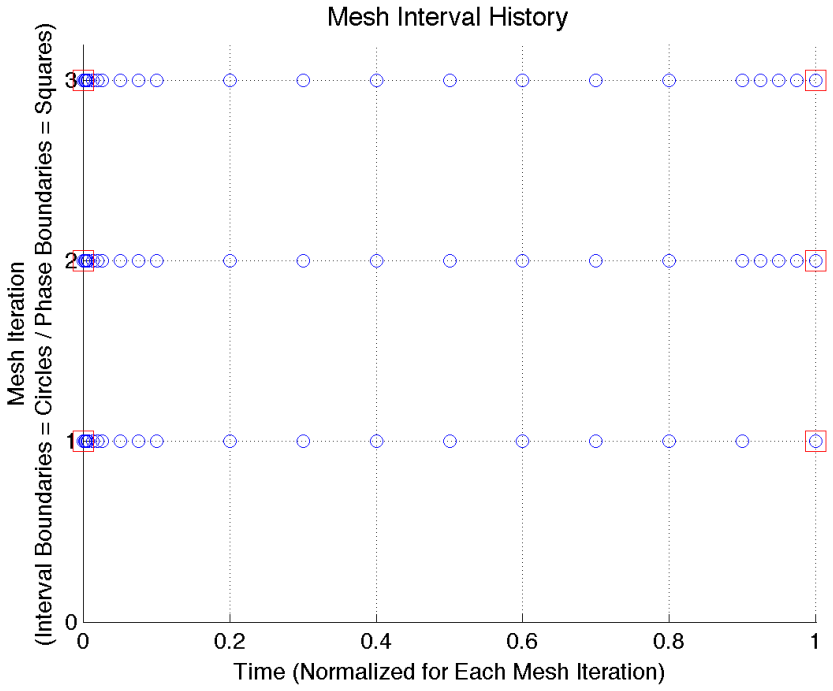

Vehicle Frame Forces
L
-D
Tsin(α)+L
Tcos(α)-D
Forces (10^3 lbf)
2000
1000
0
0
200
400
600
800
1000
1200
1400
1600
1800
2000
Time (sec)

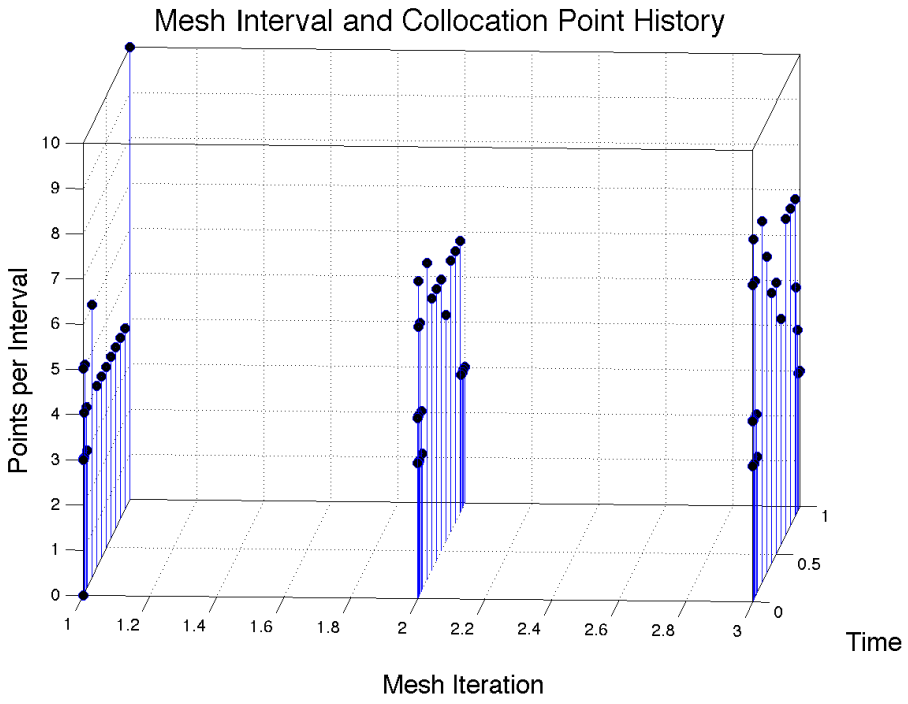

Heating Rate Profile
Nose MINIVER
Nose approx
Body MINIVER
Heating Rate (BTU/s-ft^2)
80
70
60
50
40
30
20
10
0
0
500
1000
1500
2000
2500
Time (sec)


Heating Load Profile
x 10^4
Heating Load (BTU/ft^2)
Time (sec)
Nose MINIVER
Nose approx
Body MINIVER


Temperature Profile
Temperature (deg F)
Time (sec)
Nose Outer Surface
Nose Inner Surface
Body Outer Surface
Body Inner Surface
Nose Outer Temp Limit

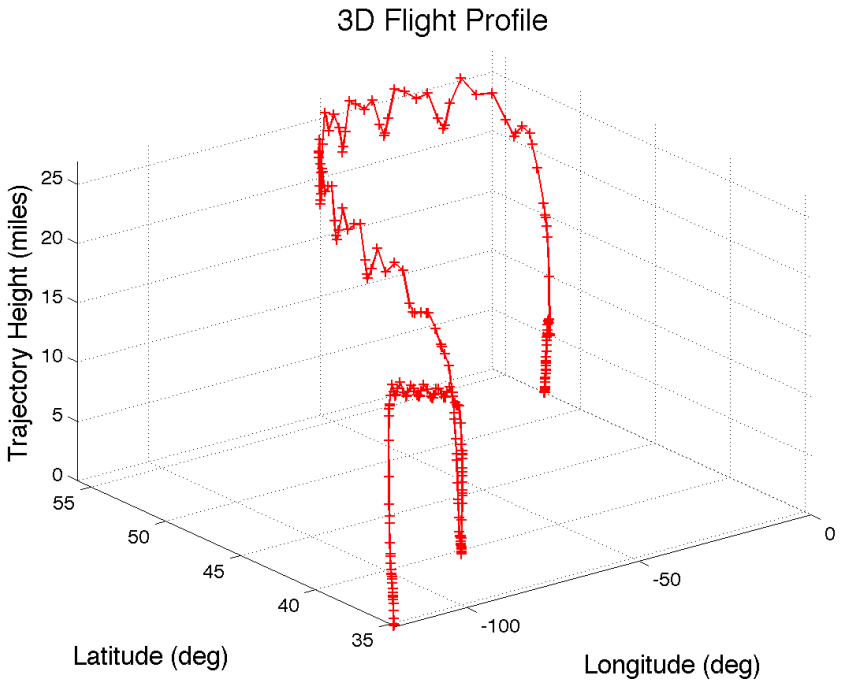

Mesh Tolerance and Maximum Relative Error
Mesh Max Rel Error
Mesh Tolerance
Mesh Tolerance and Maximum Relative Error
Mesh Iteration
10^-1
10^-2
10^-3
1
2
3

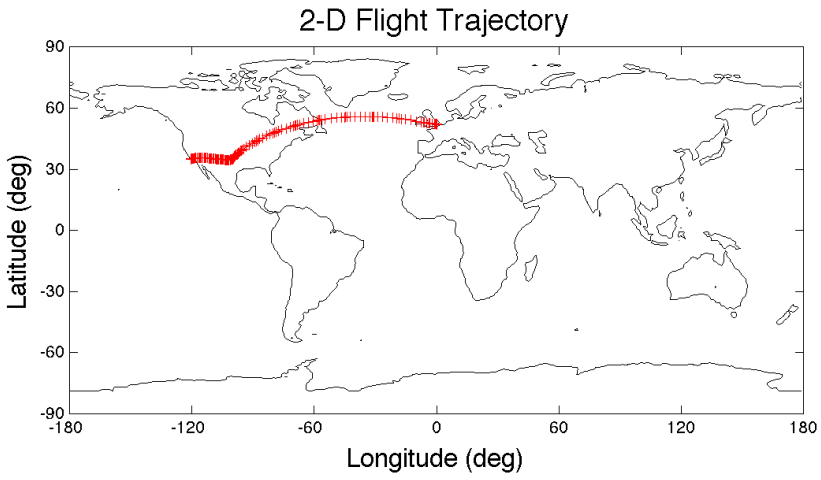

Mesh Interval History
(Interval Boundaries = Circles / Phase Boundaries = Squares)
Mesh Iteration
Time (Normalized for Each Mesh Iteration)
0
1
2
3
0
0.2
0.4
0.6
0.8
1

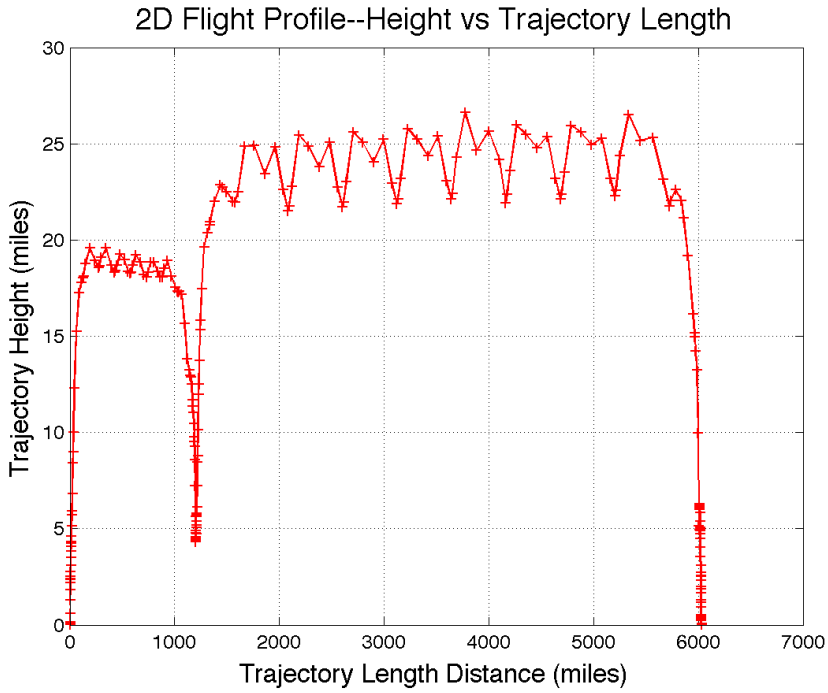


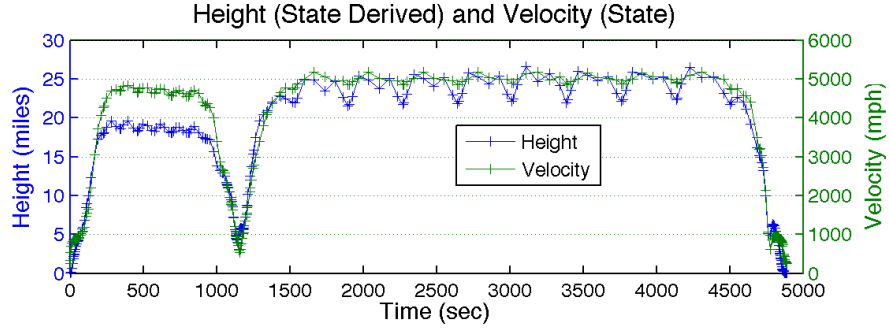
C.10 Minimum Time with Control Penalty and Temperature Constraints (Climb, Cruise, Refuel, Cruise, and Land)

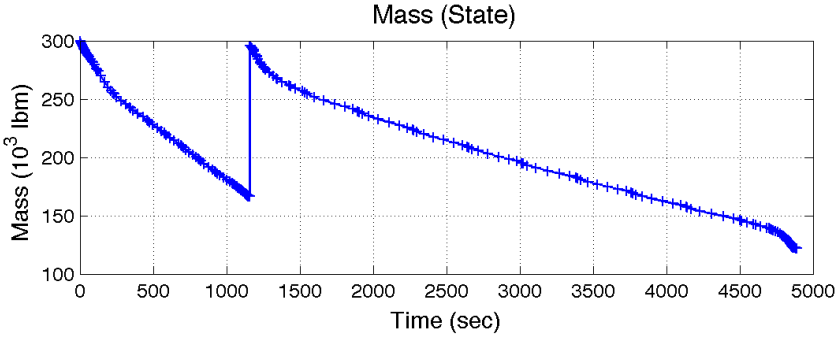


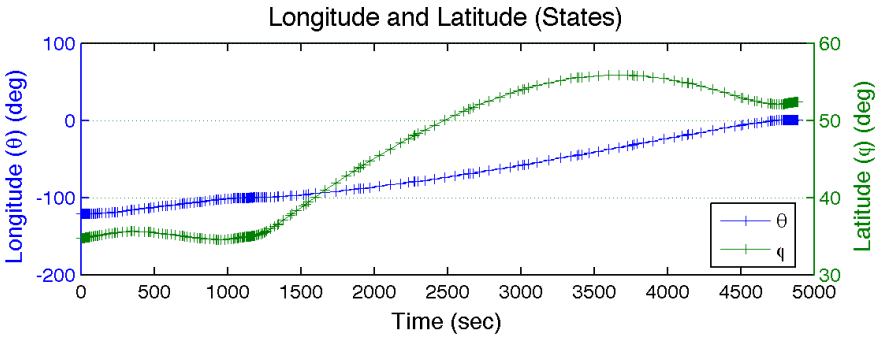

3D Flight Profile
Trajectory Height (miles)
25
20
15
10
5
0
55
50
45
40
35
Latitude (deg)
0
-50
-100
Longitude (deg)

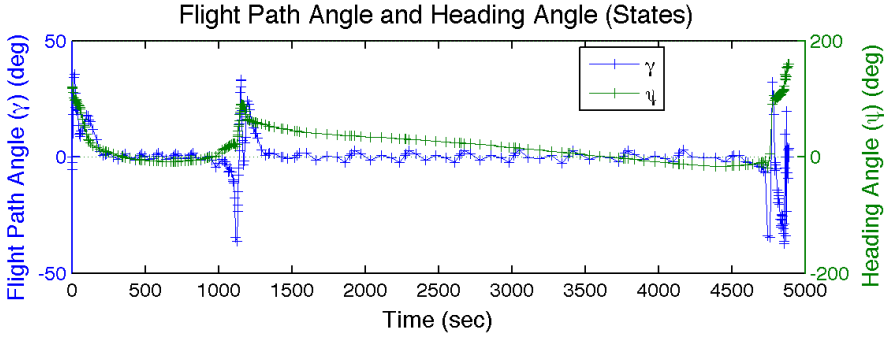

2-D Flight Trajectory
Latitude (deg)
90
60
30
0
-30
-60
-90
-180
-120
-60
0
60
120
180
Longitude (deg)

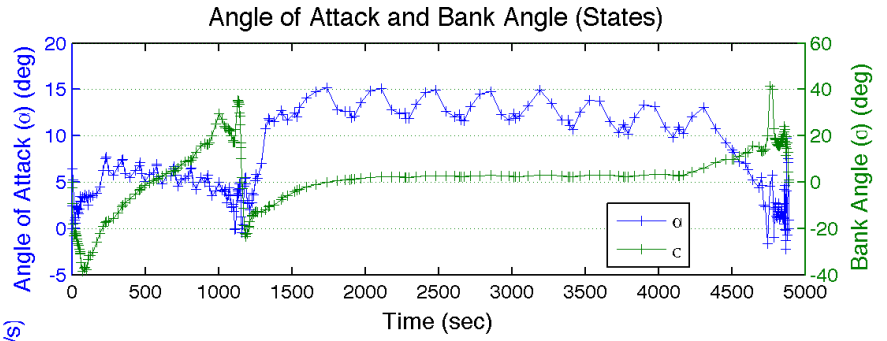

2D Flight Profile--Height vs Trajectory Length
Trajectory Height (miles)
Trajectory Length Distance (miles)
0
5
10
15
20
25
30
0
1000
2000
3000
4000
5000
6000
7000

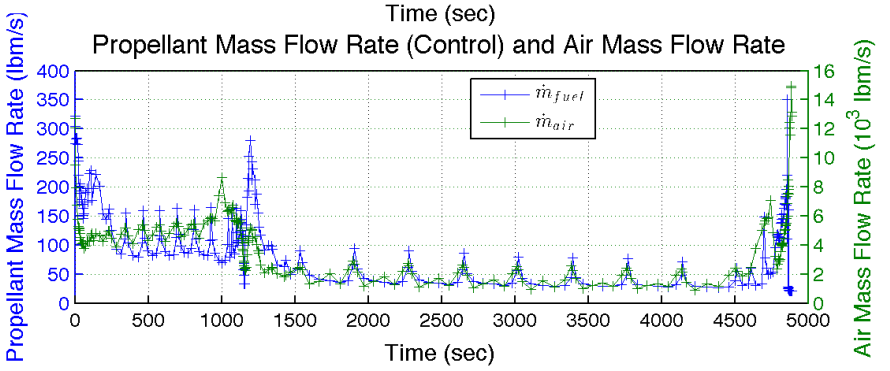

Height (State Derived) and Velocity (State)
Height (miles)
Velocity (mph)
Time (sec)
Height
Velocity
0
5
10
15
20
25
30
0
1000
2000
3000
4000
5000
6000
0
500
1000
1500
2000
2500
3000
3500
4000
4500
5000

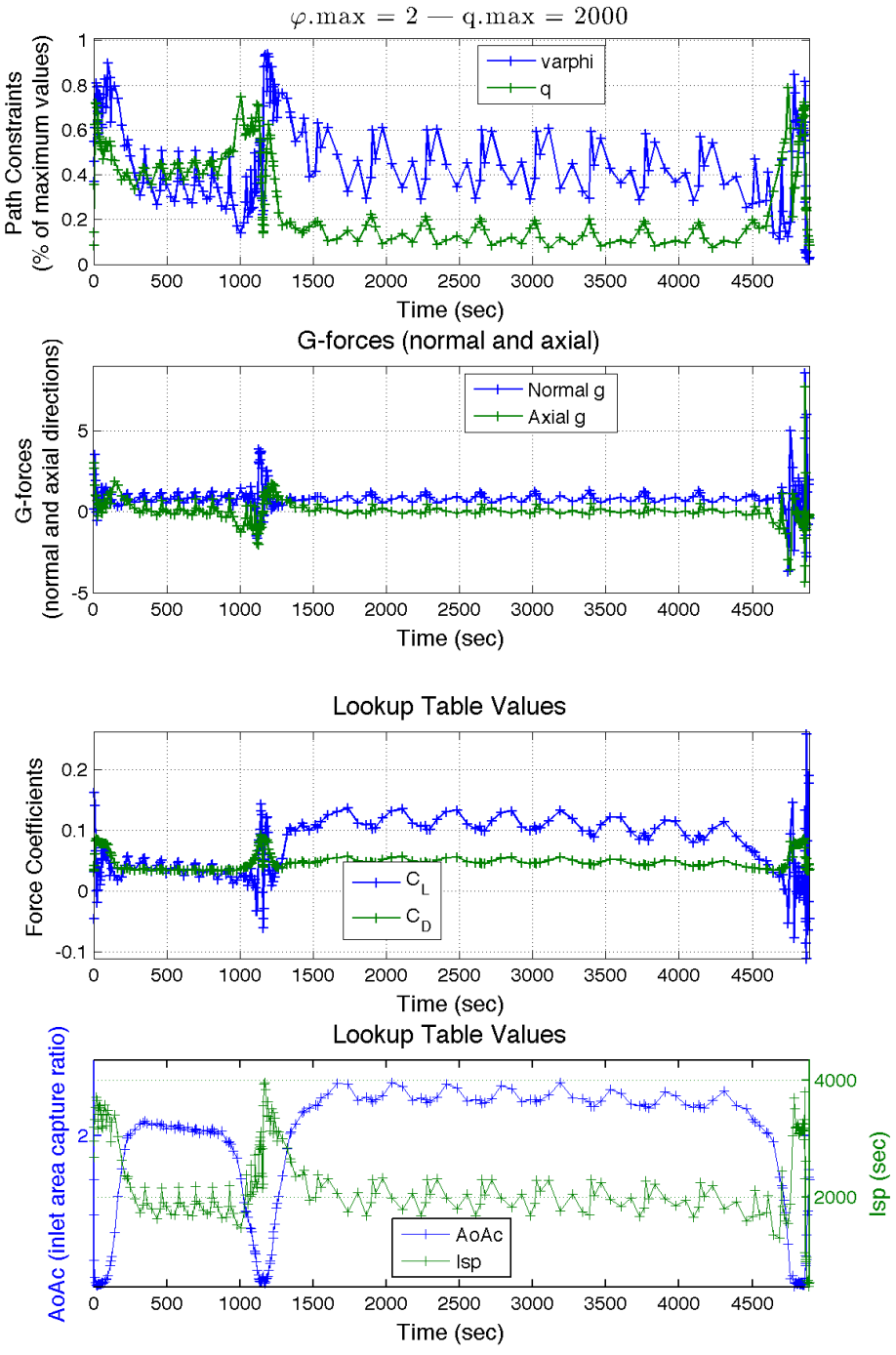

Mass (State)
Mass (10^3 lbm)
Time (sec)
100
150
200
250
300
0
500
1000
1500
2000
2500
3000
3500
4000
4500
5000

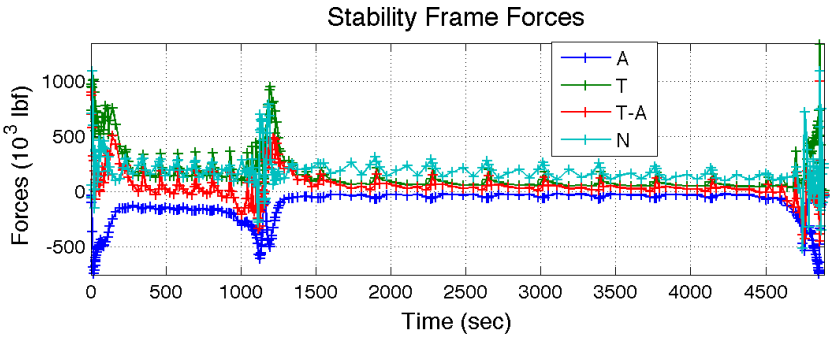

Longitude and Latitude (States)
Longitude (θ) (deg)
Latitude (φ) (deg)
Time (sec)
θ
φ

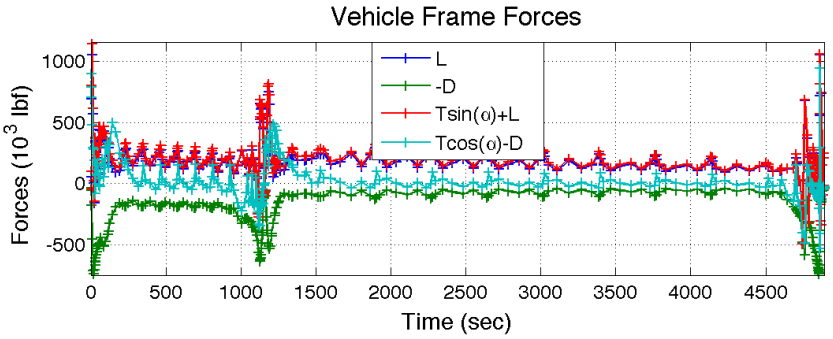

Flight Path Angle and Heading Angle (States)
Flight Path Angle (γ) (deg)
Heading Angle (ψ) (deg)
Time (sec)
γ
ψ

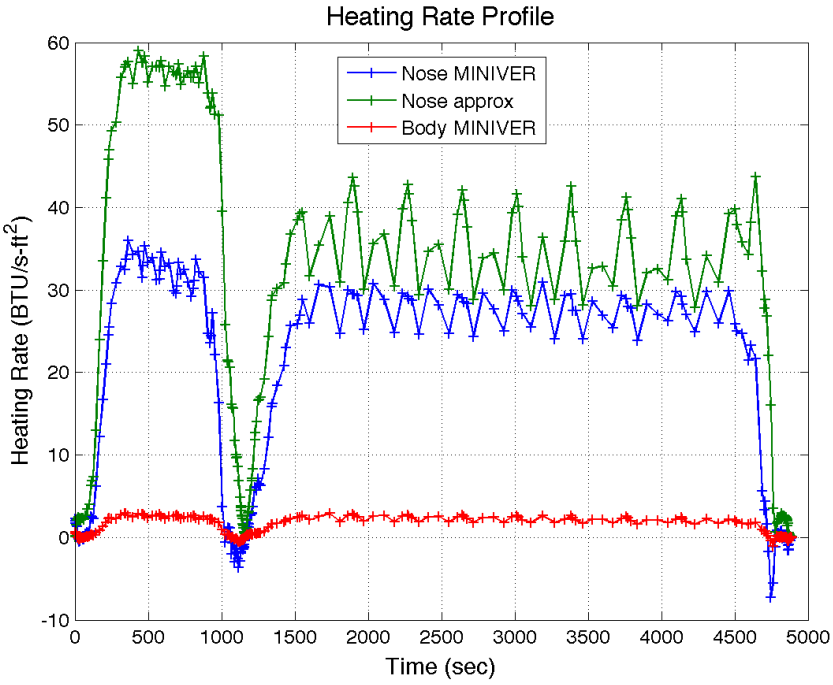

Angle of Attack and Bank Angle (States)
Angle of Attack (α) (deg)
Bank Angle (σ) (deg)
Time (sec)
α
σ

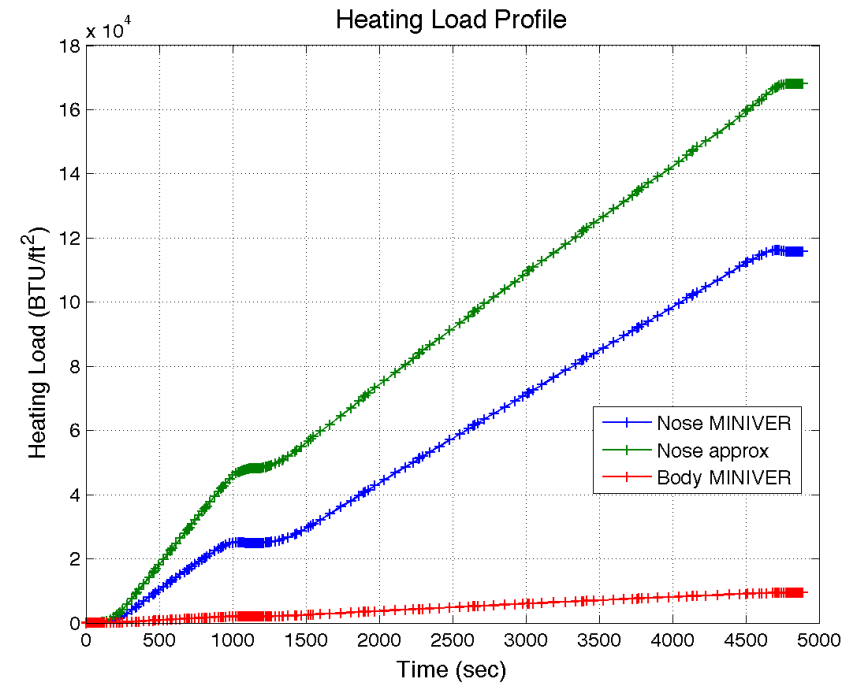

Propellant Mass Flow Rate (Control) and Air Mass Flow Rate
Propellant Mass Flow Rate (lbm/s)
Air Mass Flow Rate (10^3 lbm/s)
Time (sec)
$\dot{m}_{fuel}$
$\dot{m}_{air}$

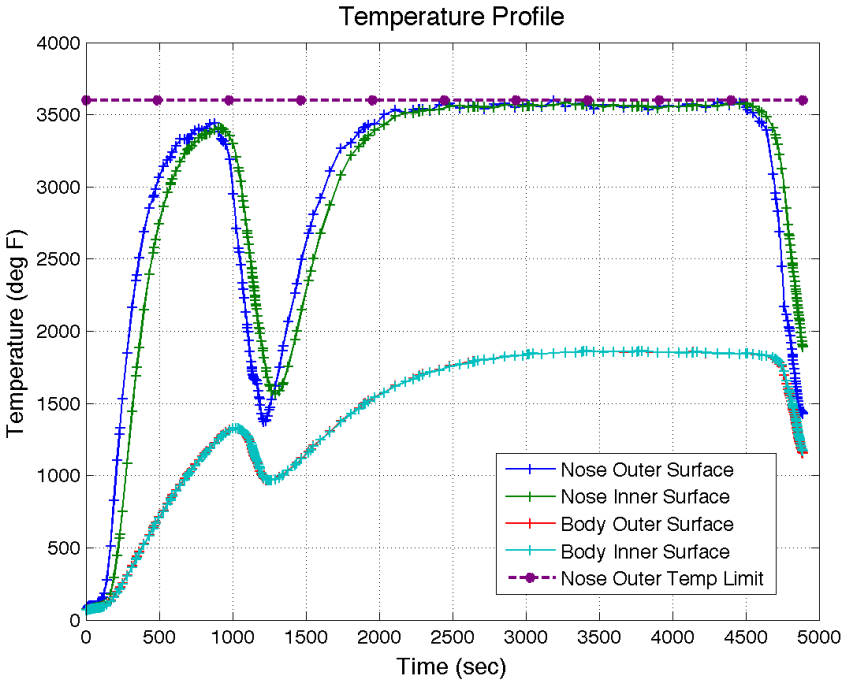

φ.max = 2 — q.max = 2000
Path Constraints
(% of maximum values)
varphi
q
Time (sec)
G-forces (normal and axial)
G-forces
(normal and axial directions)
Normal g
Axial g
Time (sec)
Lookup Table Values
Force Coefficients
C_L
C_D
Time (sec)
Lookup Table Values
AoAc (inlet area capture ratio)
Isp (sec)
AoAc
Isp
Time (sec)

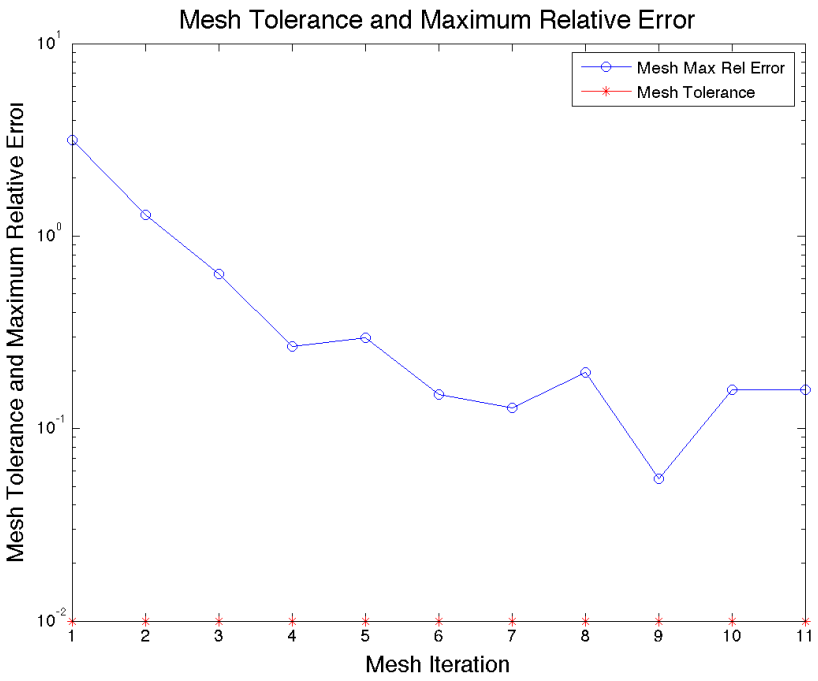

Stability Frame Forces
A
T
T-A
N
Forces (10^3 lbf)
1000
500
0
-500
0
500
1000
1500
2000
2500
3000
3500
4000
4500
Time (sec)

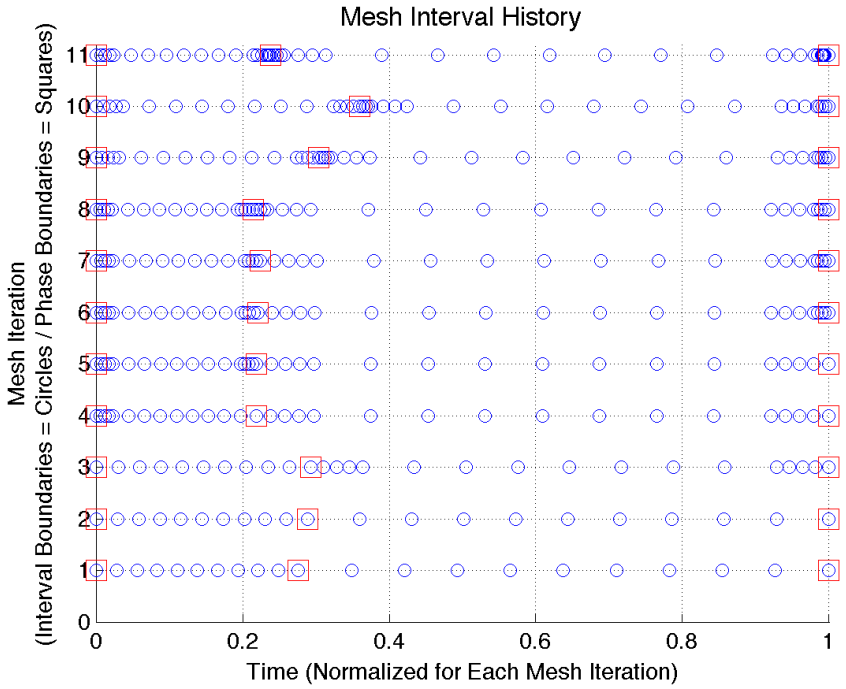

Vehicle Frame Forces
L
-D
Tsin(α)+L
Tcos(α)-D
Forces (10^3 lbf)
1000
500
0
-500
0
500
1000
1500
2000
2500
3000
3500
4000
4500
Time (sec)

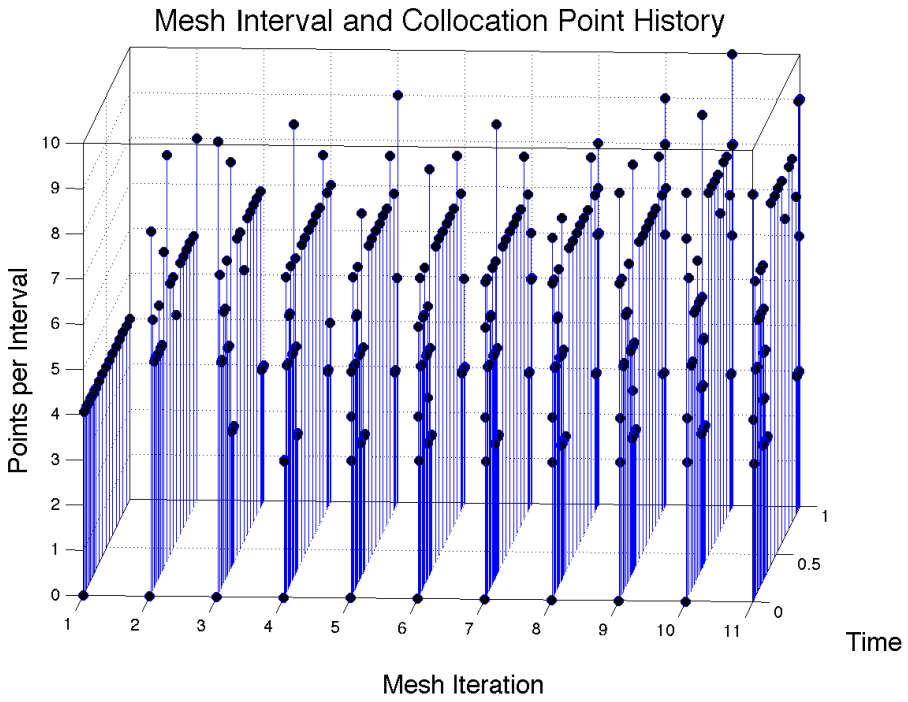

Heating Rate Profile
Nose MINIVER
Nose approx
Body MINIVER
Heating Rate (BTU/s-ft^2)
60
50
40
30
20
10
0
-10
0
500
1000
1500
2000
2500
3000
3500
4000
4500
5000
Time (sec)


Heating Load Profile
x 10^4
Heating Load (BTU/ft^2)
Time (sec)
Nose MINIVER
Nose approx
Body MINIVER


Temperature Profile
Temperature (deg F)
Time (sec)
Nose Outer Surface
Nose Inner Surface
Body Outer Surface
Body Inner Surface
Nose Outer Temp Limit


Mesh Tolerance and Maximum Relative Error
Mesh Max Rel Error
Mesh Tolerance
Mesh Tolerance and Maximum Relative Error
Mesh Iteration


Mesh Interval History
(Interval Boundaries = Circles / Phase Boundaries = Squares)
Mesh Iteration
Time (Normalized for Each Mesh Iteration)

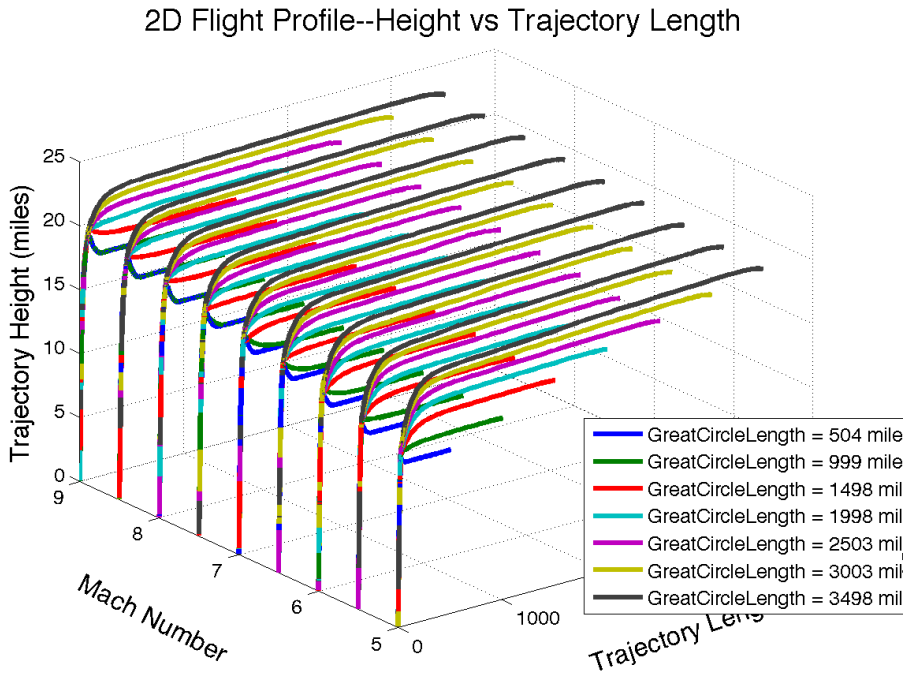

Mesh Interval and Collocation Point History
Points per Interval
Mesh Iteration
Time
0
1
2
3
4
5
6
7
8
9
10
11
0.5

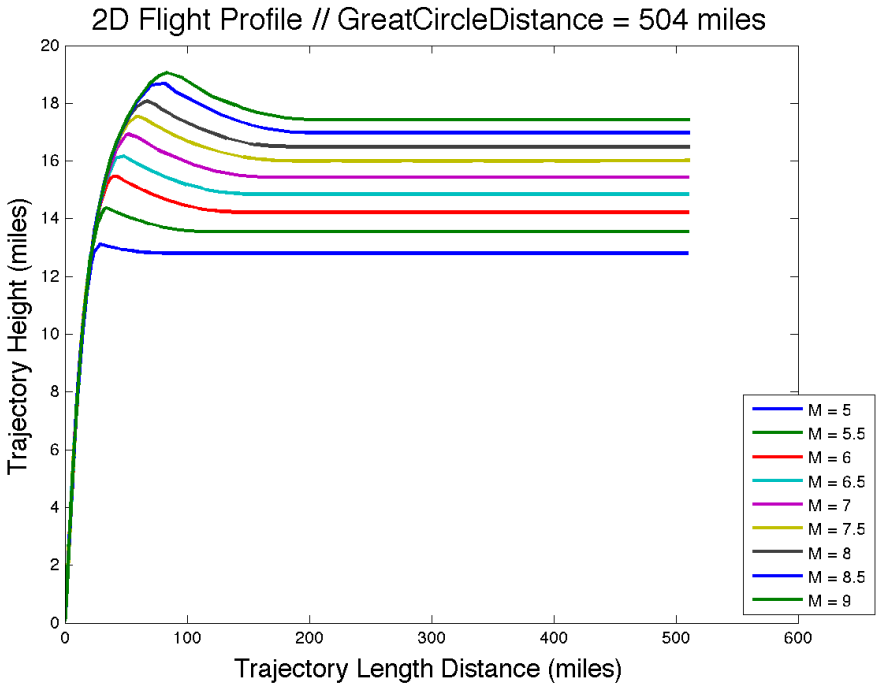
Appendix D. Parametric Analysis Detailed Charts

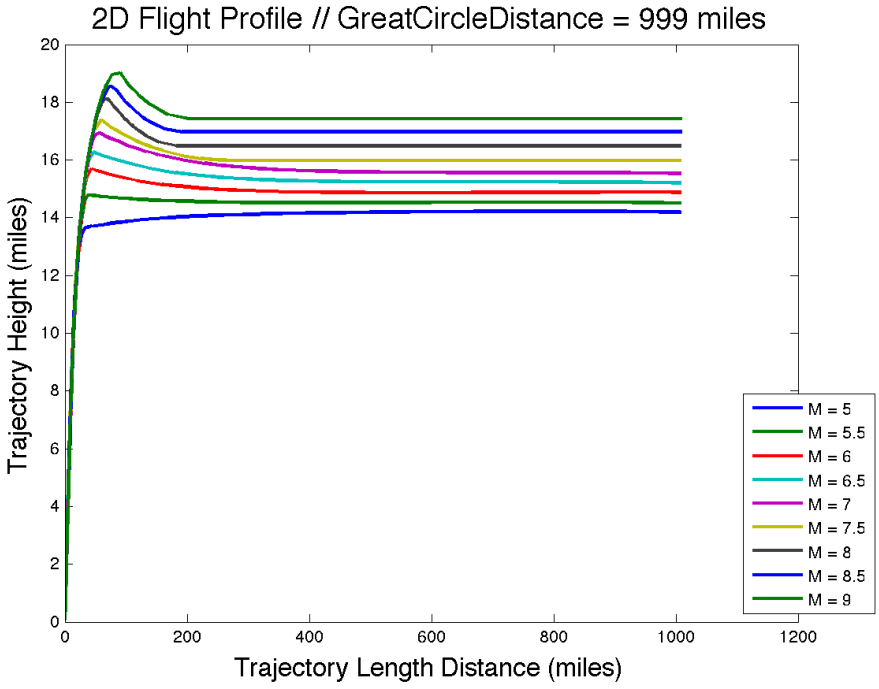
This appendix contains the detailed results used to generate the 3-D charts presented and discussed in Chapter VI.

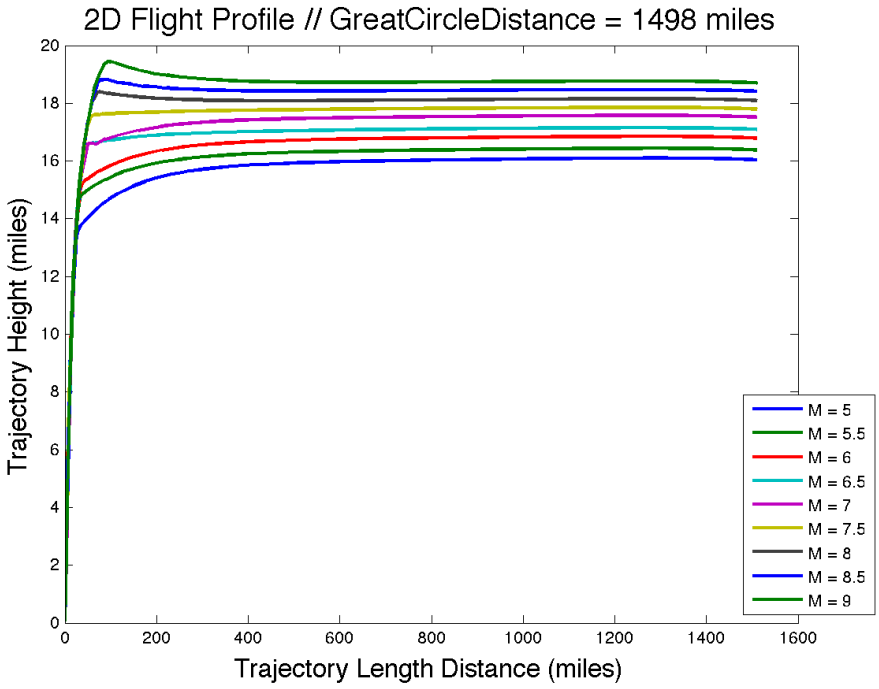
In this appendix, the section name describes the data presented in that section. For example, in Appendix D Section D.1, Trajectory Height is the z-axis and Path (Trajectory) Length is the x-axis. The first chart in the section is a 3-D summary of the subsequent charts, which are presented in two different perspectives. The first perspective varies the maximum Mach number from 5.0 to 9.0 for each increment of Great Circle Distance. The second perspective varies the Great Circle Distance from (approximately) 500 to 3500 miles for each increment of maximum Mach number. Both perspectives show the results from common scenario runs, just from different perspectives.

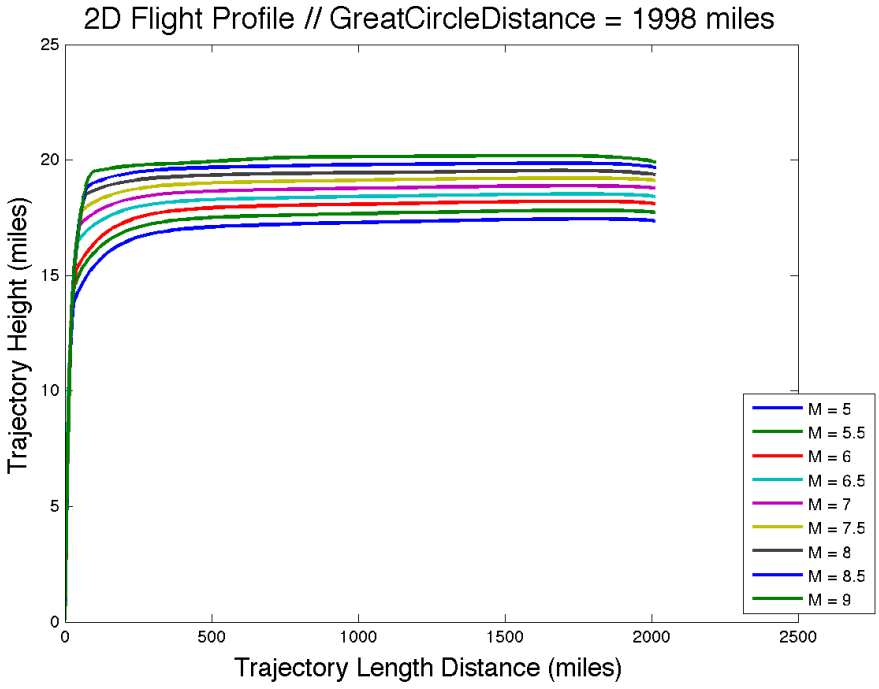
D.1 Trajectory Height vs. Path Length

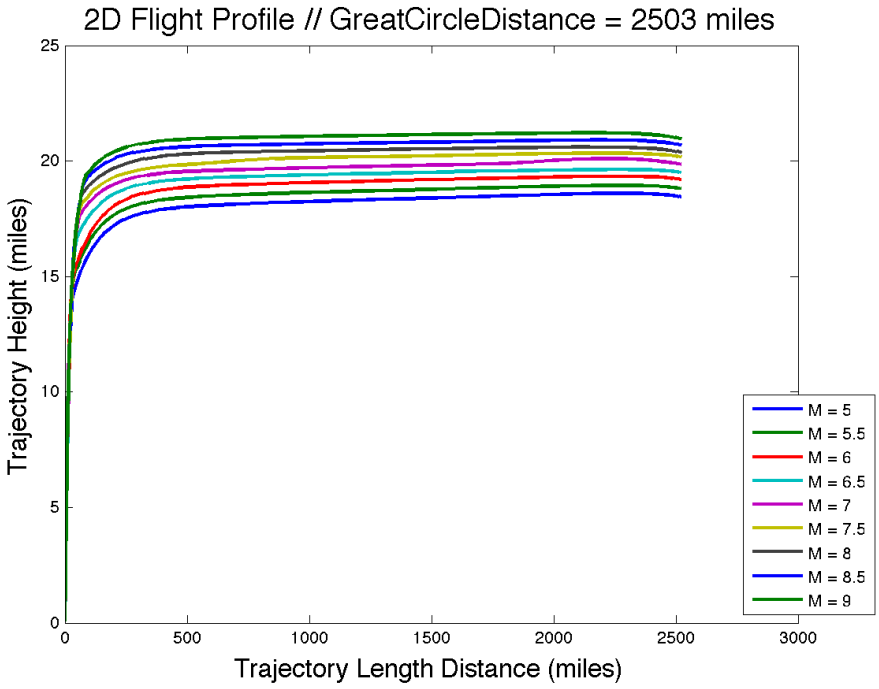


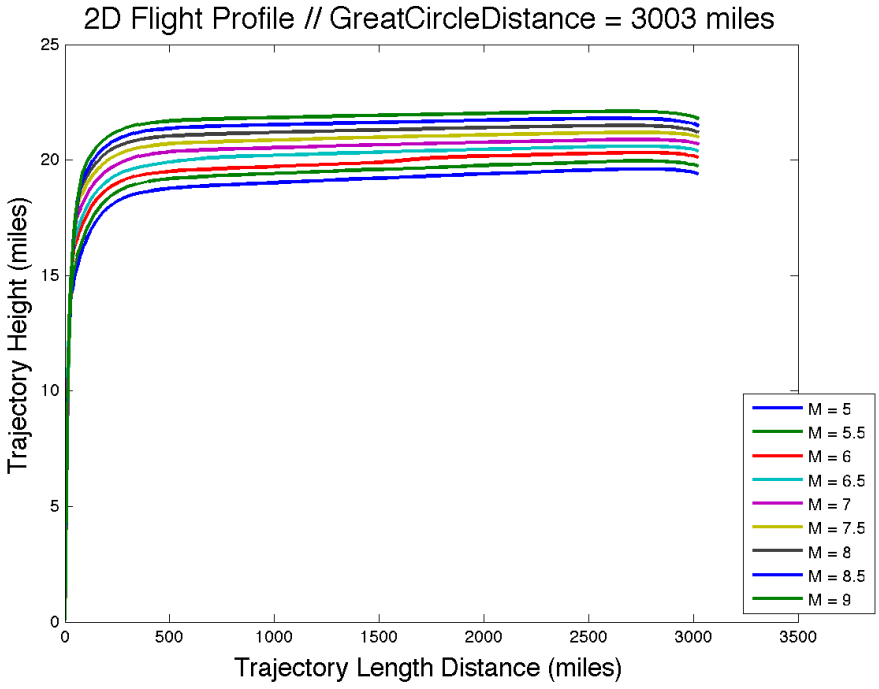

2D Flight Profile // GreatCircleDistance = 504 miles
Trajectory Height (miles)
Trajectory Length Distance (miles)
0
2
4
6
8
10
12
14
16
18
20
0
100
200
300
400
500
600
M = 5
M = 5.5
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9

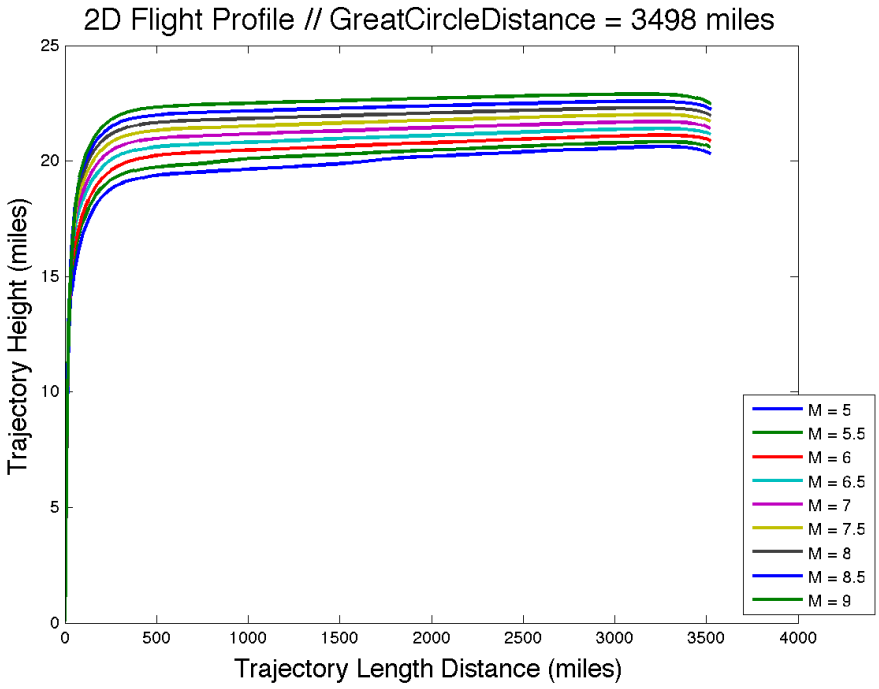

2D Flight Profile // GreatCircleDistance = 999 miles
Trajectory Height (miles)
Trajectory Length Distance (miles)
0
2
4
6
8
10
12
14
16
18
20
0
200
400
600
800
1000
1200
M = 5
M = 5.5
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9

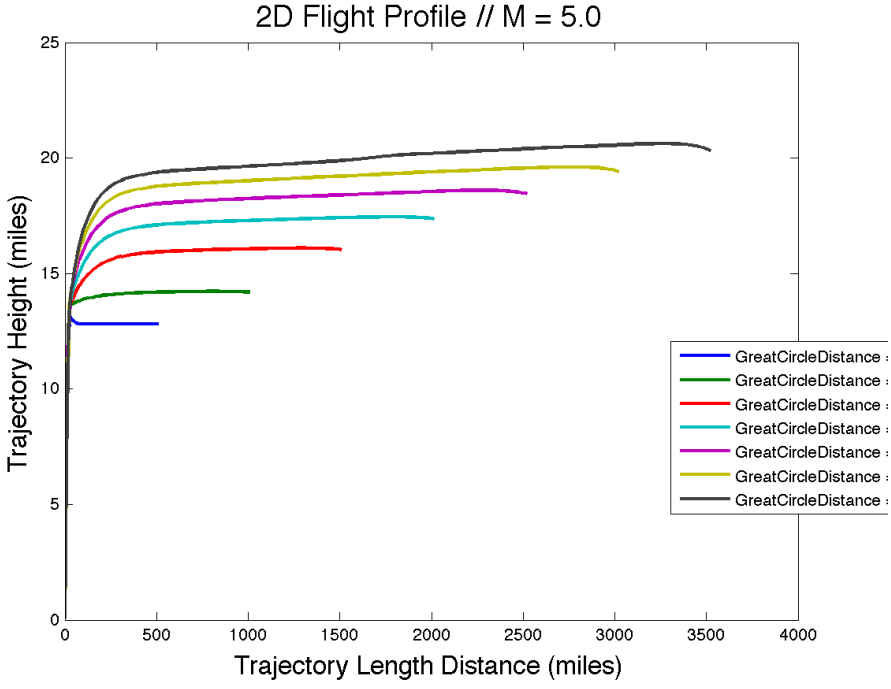

2D Flight Profile // GreatCircleDistance = 1498 miles
Trajectory Height (miles)
Trajectory Length Distance (miles)
0
2
4
6
8
10
12
14
16
18
20
0
200
400
600
800
1000
1200
1400
1600
M = 5
M = 5.5
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9

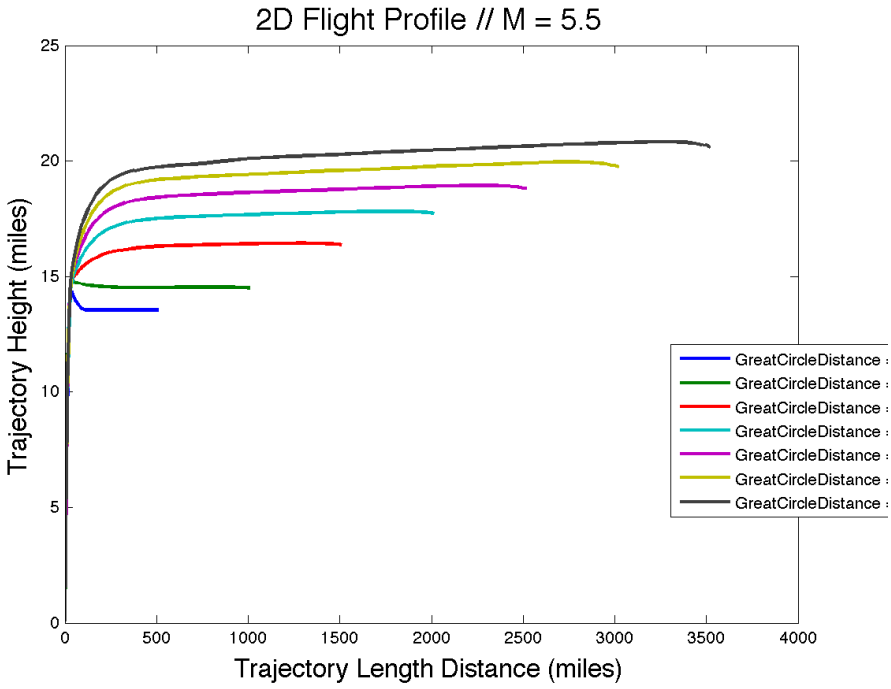

2D Flight Profile // GreatCircleDistance = 1998 miles
Trajectory Height (miles)
Trajectory Length Distance (miles)
0
5
10
15
20
25
0
500
1000
1500
2000
2500
M = 5
M = 5.5
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9

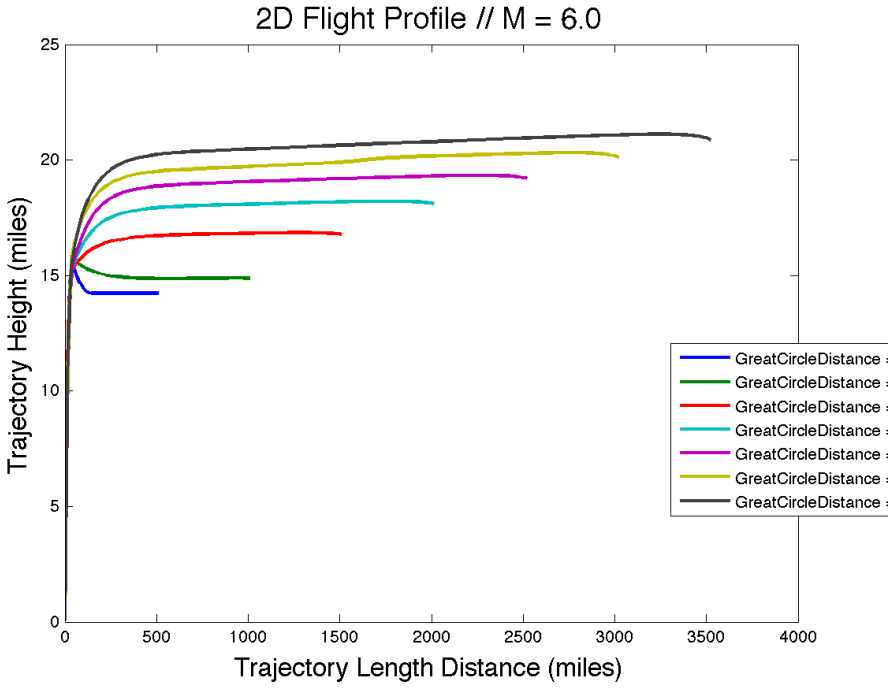

2D Flight Profile // GreatCircleDistance = 2503 miles
Trajectory Height (miles)
Trajectory Length Distance (miles)
0
5
10
15
20
25
0
500
1000
1500
2000
2500
3000
M = 5
M = 5.5
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9

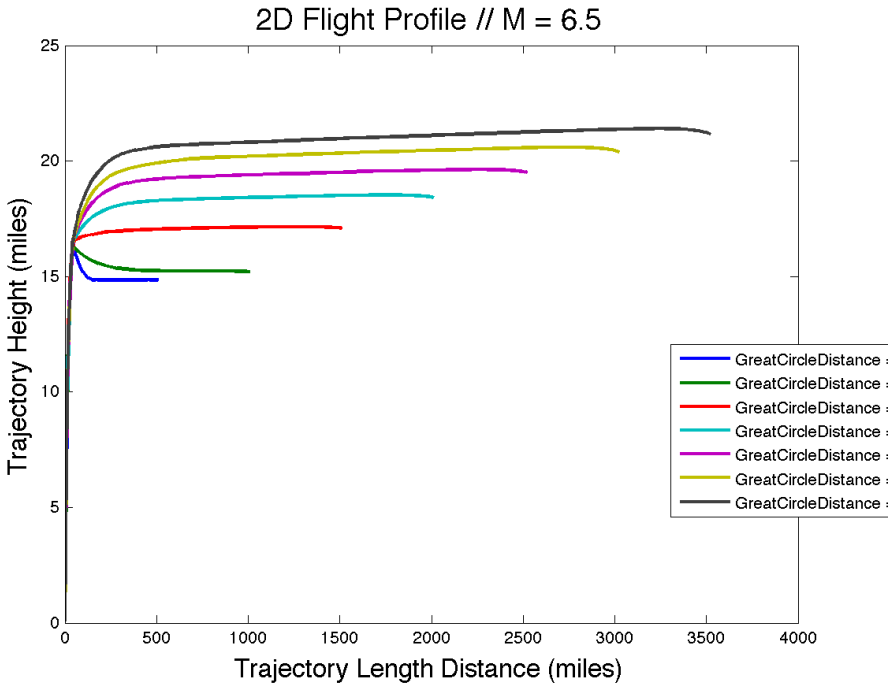

2D Flight Profile // GreatCircleDistance = 3003 miles
Trajectory Height (miles)
Trajectory Length Distance (miles)
0
5
10
15
20
25
0
500
1000
1500
2000
2500
3000
3500
M = 5
M = 5.5
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9

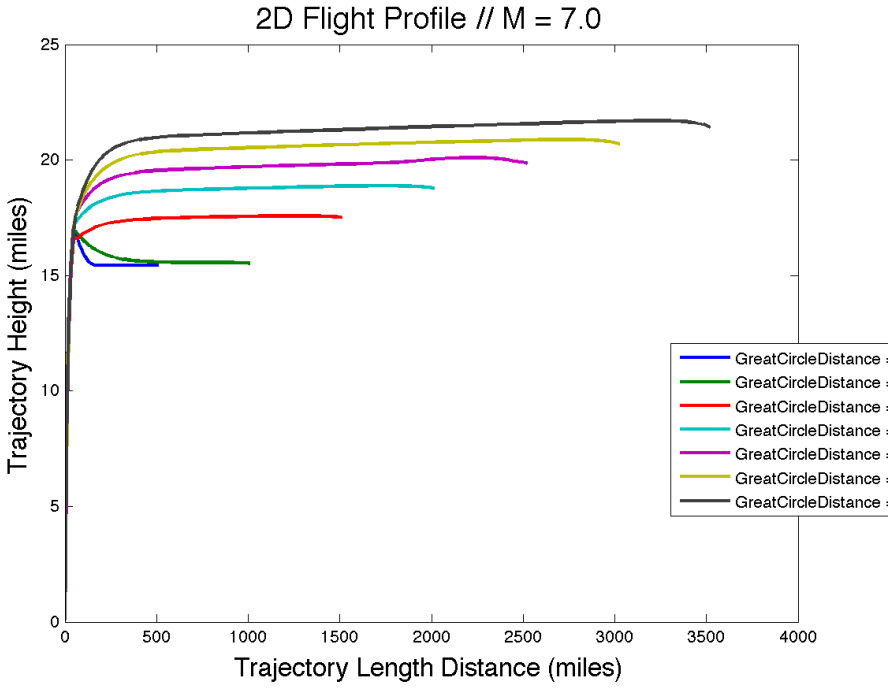

2D Flight Profile // GreatCircleDistance = 3498 miles
Trajectory Height (miles)
Trajectory Length Distance (miles)
0
5
10
15
20
25
0
500
1000
1500
2000
2500
3000
3500
4000
M = 5
M = 5.5
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9

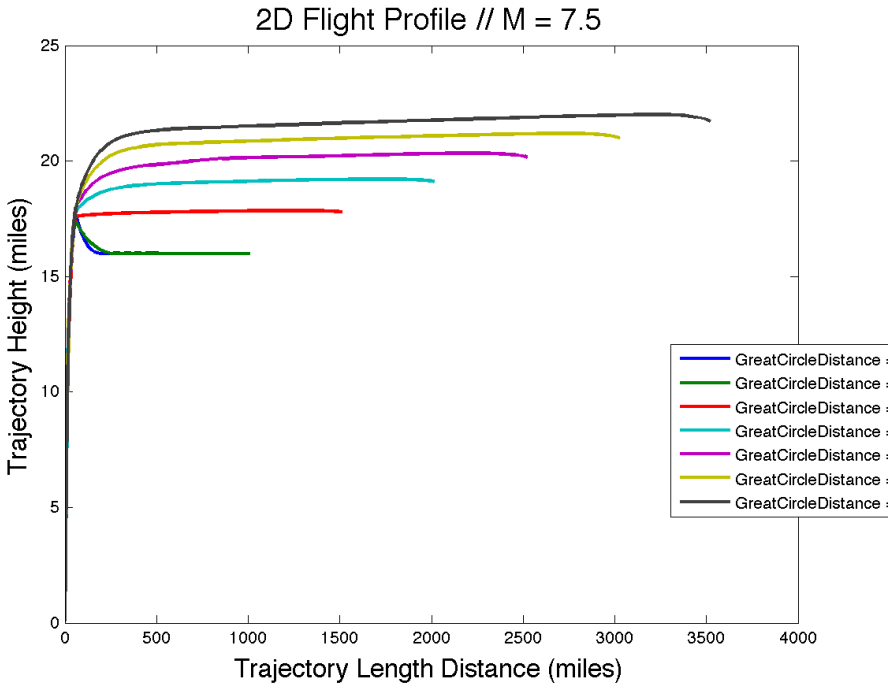

2D Flight Profile // M = 5.0
Trajectory Height (miles)
Trajectory Length Distance (miles)
0
5
10
15
20
25
0
500
1000
1500
2000
2500
3000
3500
4000
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance

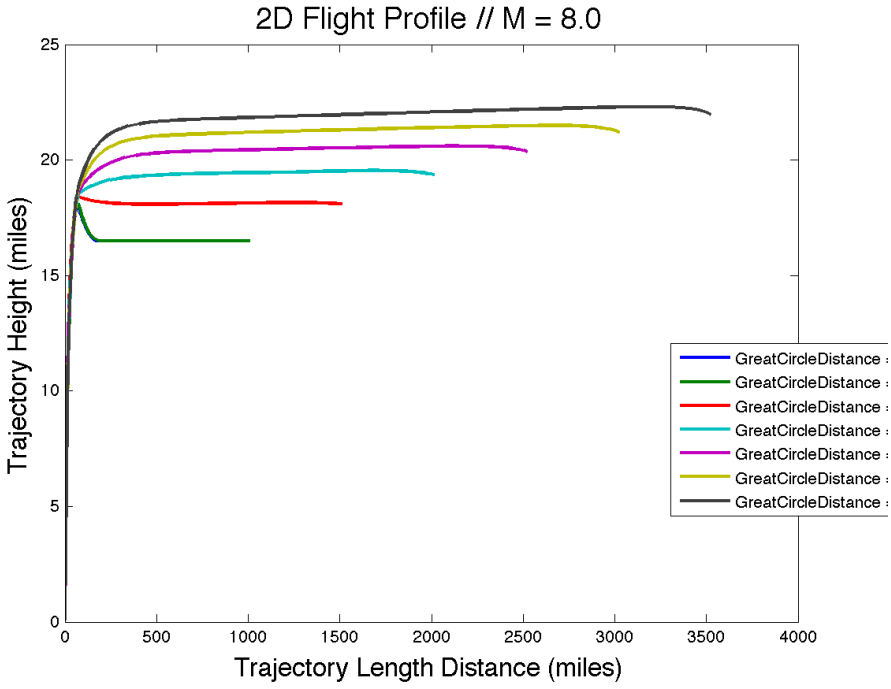

2D Flight Profile // M = 5.5
Trajectory Height (miles)
Trajectory Length Distance (miles)
0
5
10
15
20
25
0
500
1000
1500
2000
2500
3000
3500
4000
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :

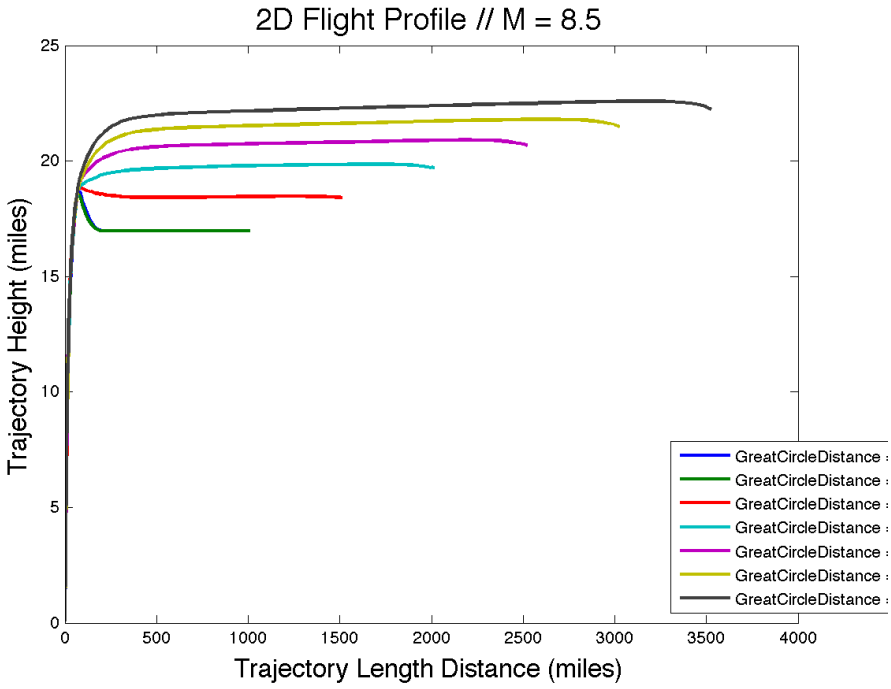

2D Flight Profile // M = 6.0
Trajectory Height (miles)
Trajectory Length Distance (miles)
0
5
10
15
20
25
0
500
1000
1500
2000
2500
3000
3500
4000
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :

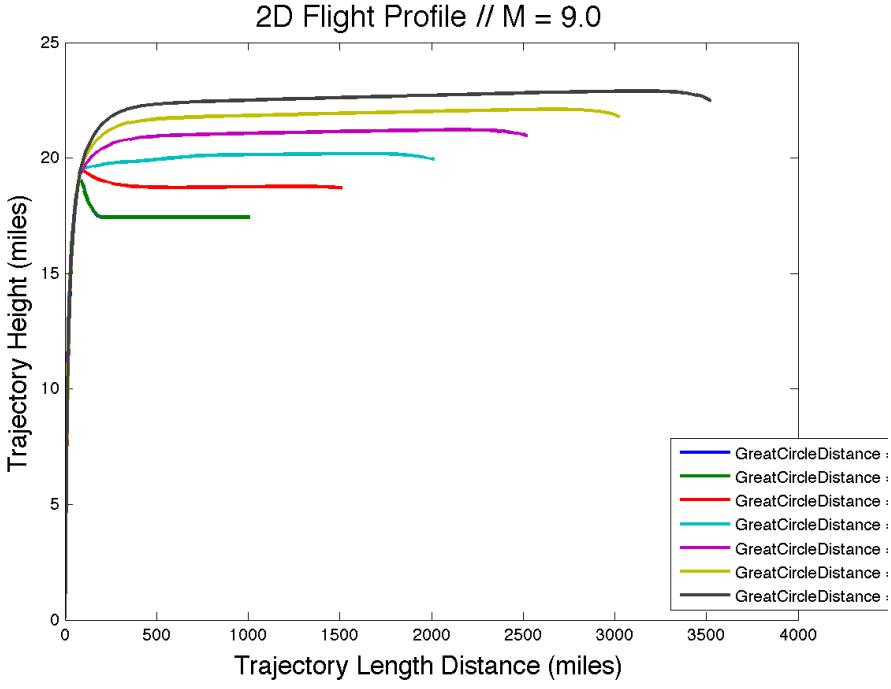

2D Flight Profile // M = 6.5
Trajectory Height (miles)
Trajectory Length Distance (miles)
0
5
10
15
20
25
0
500
1000
1500
2000
2500
3000
3500
4000
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :


2D Flight Profile // M = 7.0
Trajectory Height (miles)
Trajectory Length Distance (miles)
0
5
10
15
20
25
0
500
1000
1500
2000
2500
3000
3500
4000
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :

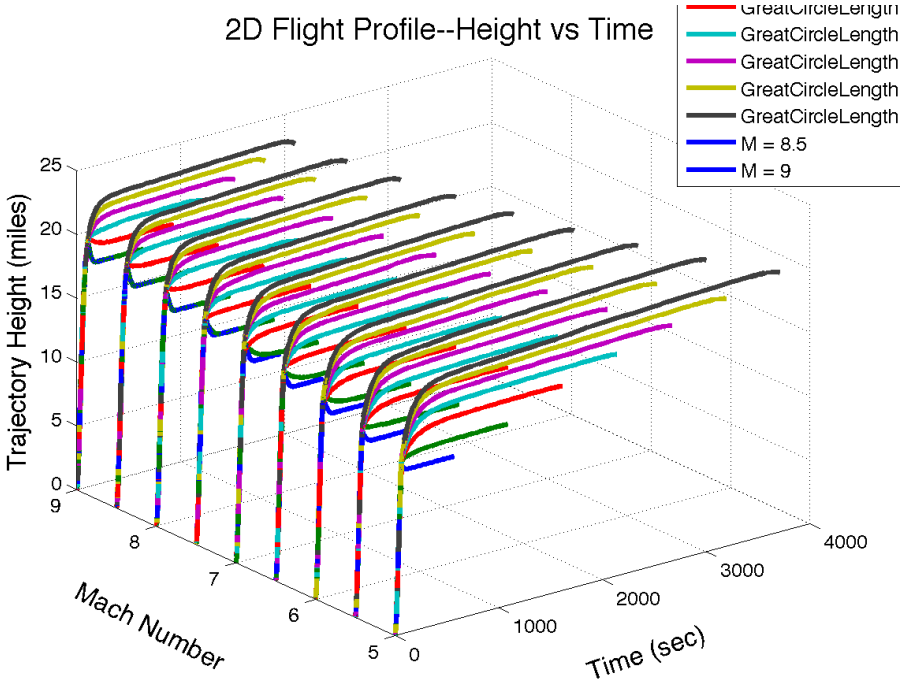

2D Flight Profile // M = 7.5
Trajectory Height (miles)
Trajectory Length Distance (miles)
0
5
10
15
20
25
0
500
1000
1500
2000
2500
3000
3500
4000
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :

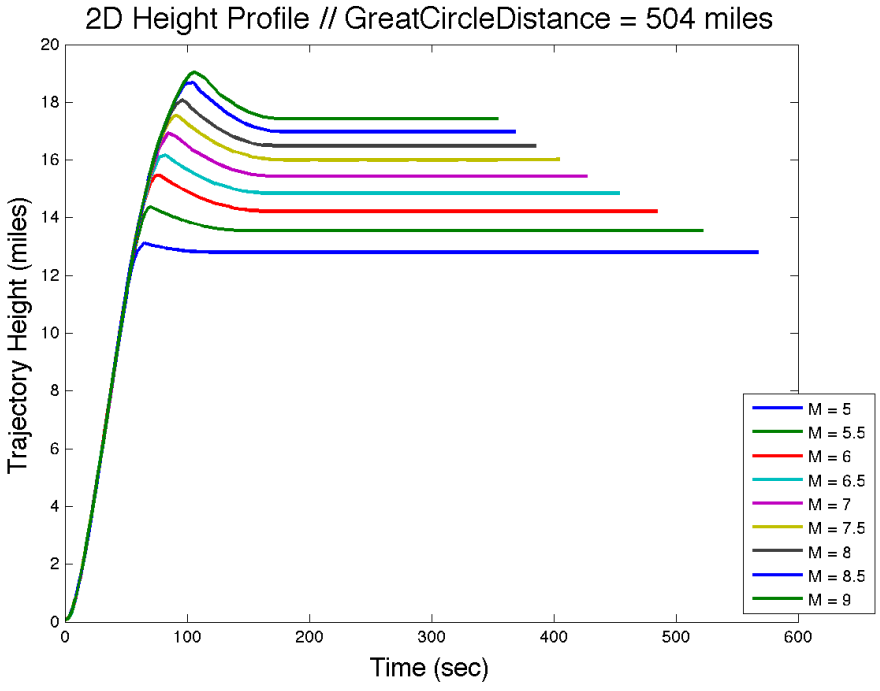

2D Flight Profile // M = 8.0
Trajectory Height (miles)
Trajectory Length Distance (miles)
0
5
10
15
20
25
0
500
1000
1500
2000
2500
3000
3500
4000
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :

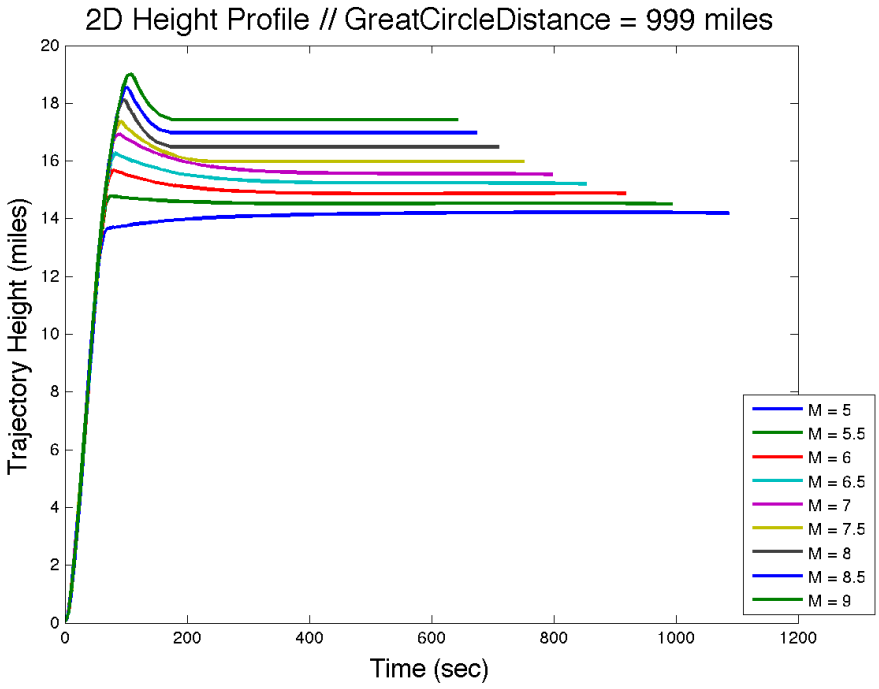

2D Flight Profile // M = 8.5
Trajectory Height (miles)
Trajectory Length Distance (miles)
0
5
10
15
20
25
0
500
1000
1500
2000
2500
3000
3500
4000
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance

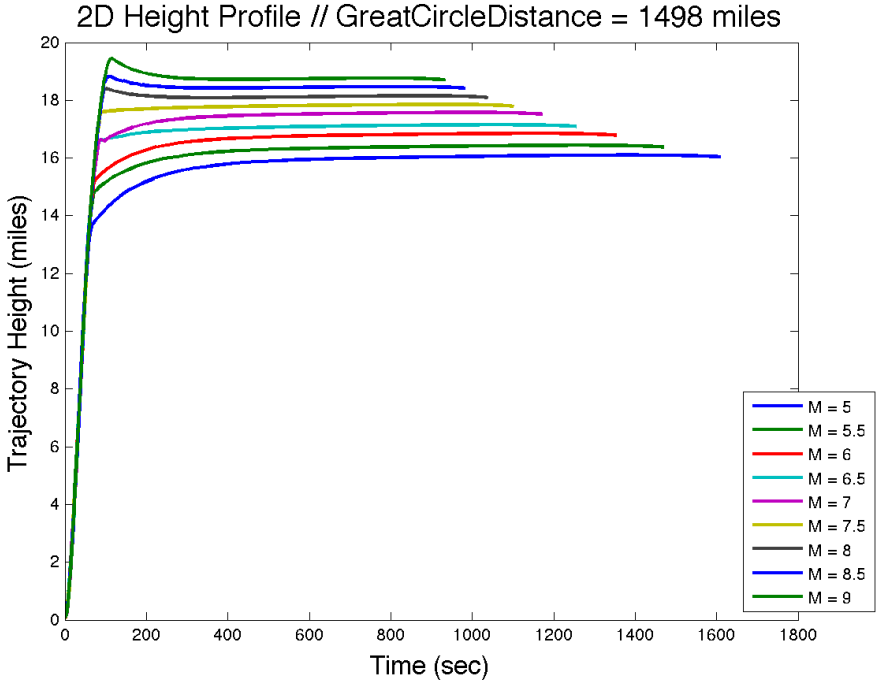

2D Flight Profile // M = 9.0
Trajectory Height (miles)
Trajectory Length Distance (miles)
0
5
10
15
20
25
0
500
1000
1500
2000
2500
3000
3500
4000
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance

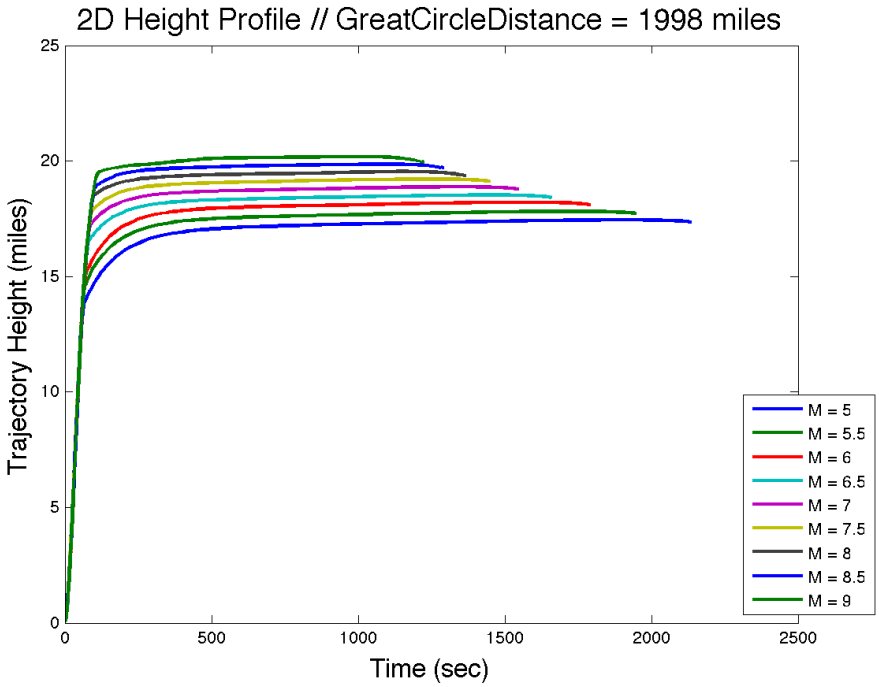
D.2 Trajectory Height vs. Time

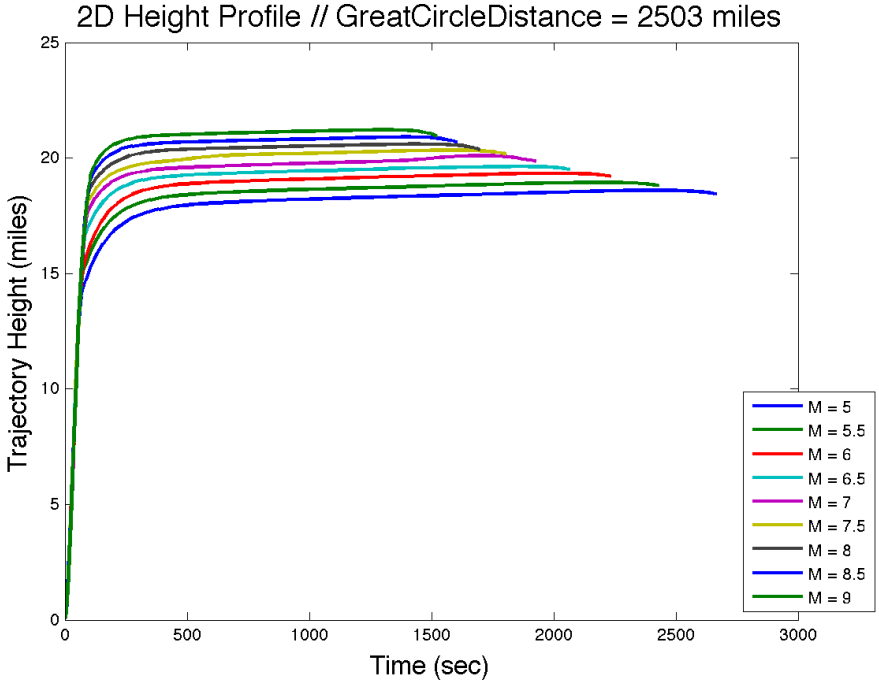


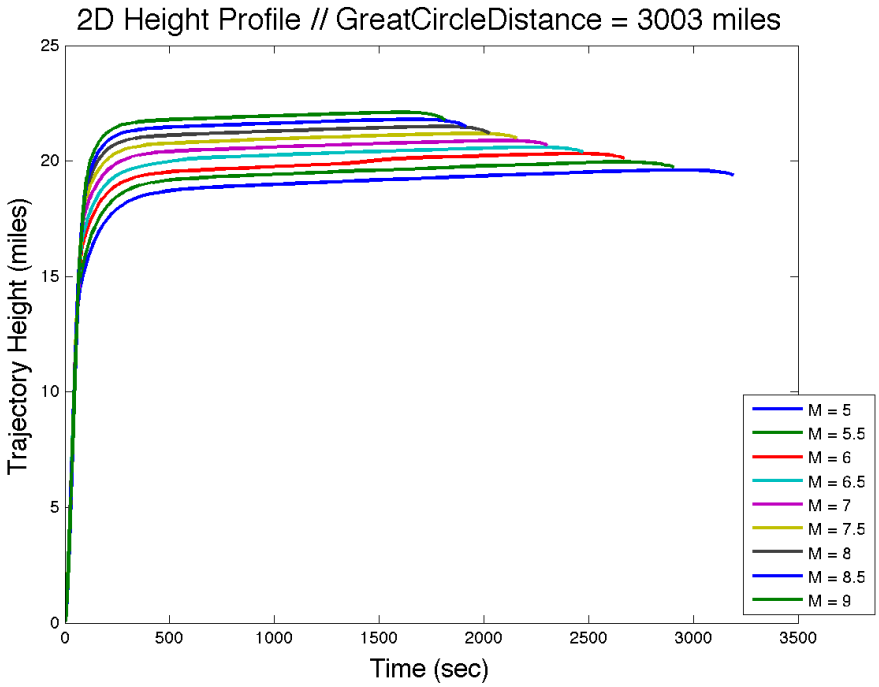


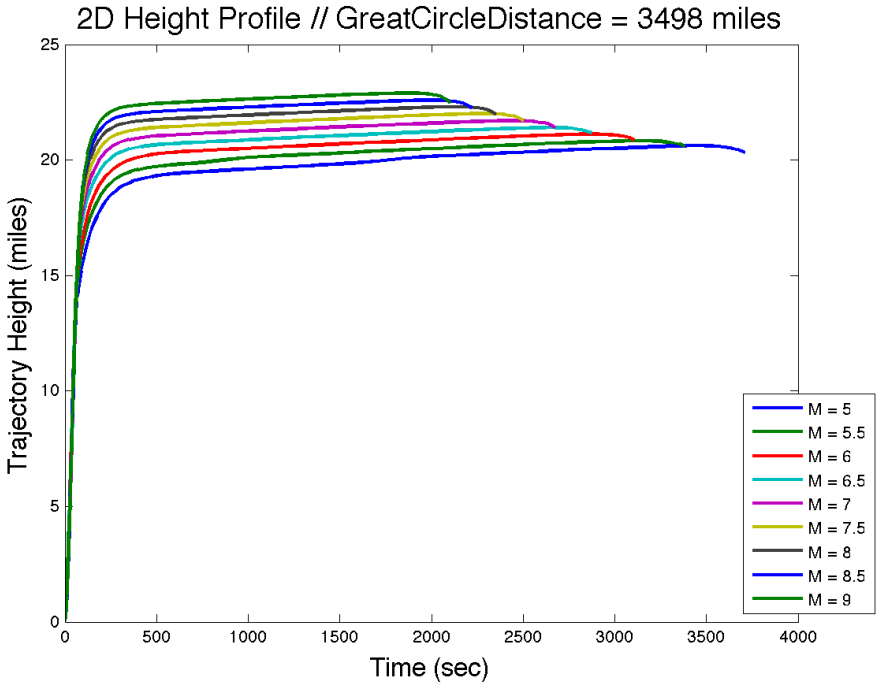

2D Height Profile // GreatCircleDistance = 999 miles
Trajectory Height (miles)
Time (sec)
0
2
4
6
8
10
12
14
16
18
20
0
200
400
600
800
1000
1200
M = 5
M = 5.5
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9

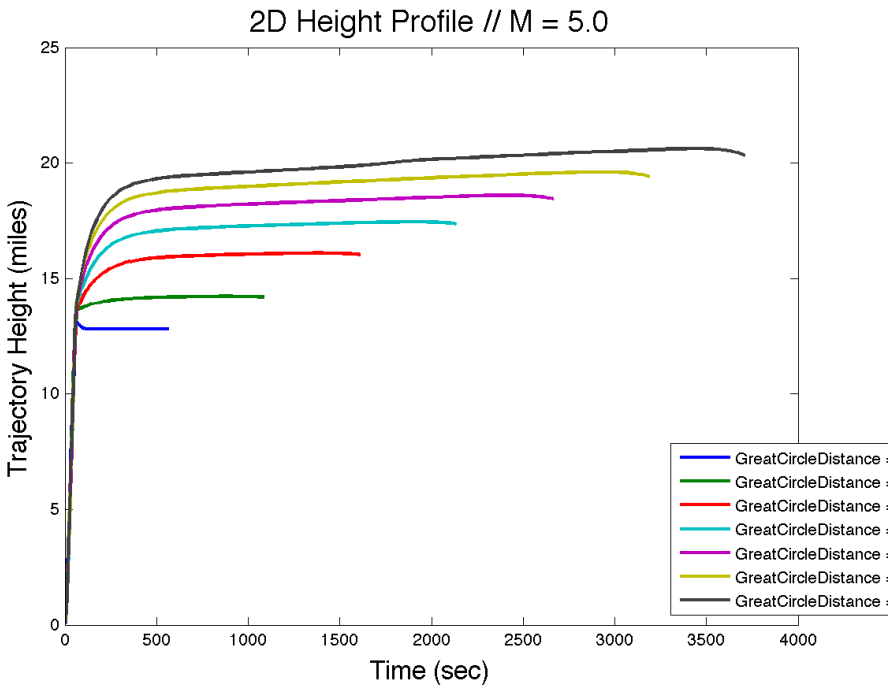

2D Height Profile // GreatCircleDistance = 1498 miles
Trajectory Height (miles)
Time (sec)
0
2
4
6
8
10
12
14
16
18
20
0
200
400
600
800
1000
1200
1400
1600
1800
M = 5
M = 5.5
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9

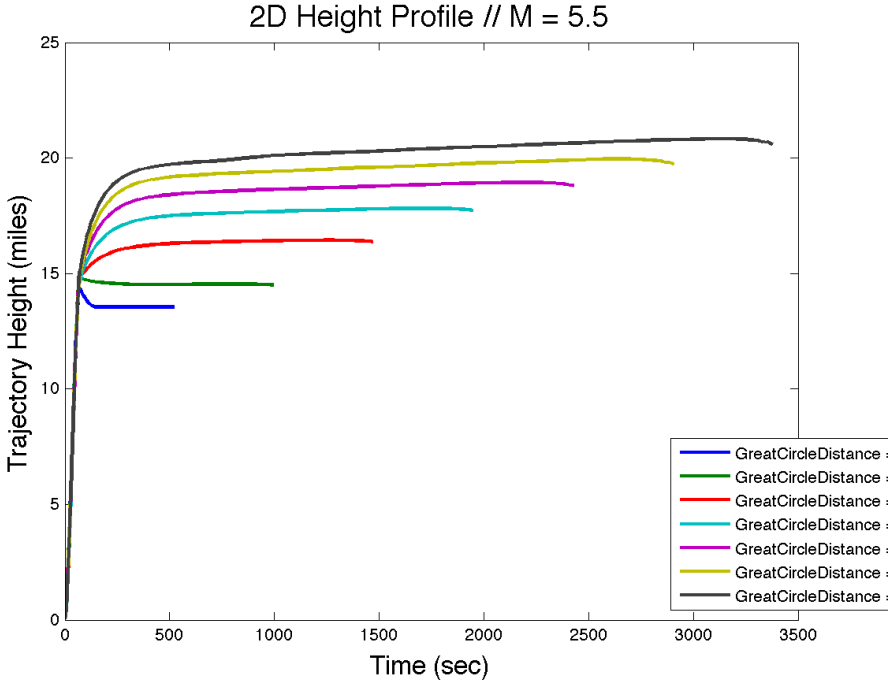

2D Height Profile // GreatCircleDistance = 1998 miles
Trajectory Height (miles)
Time (sec)
0
5
10
15
20
25
0
500
1000
1500
2000
2500
M = 5
M = 5.5
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9

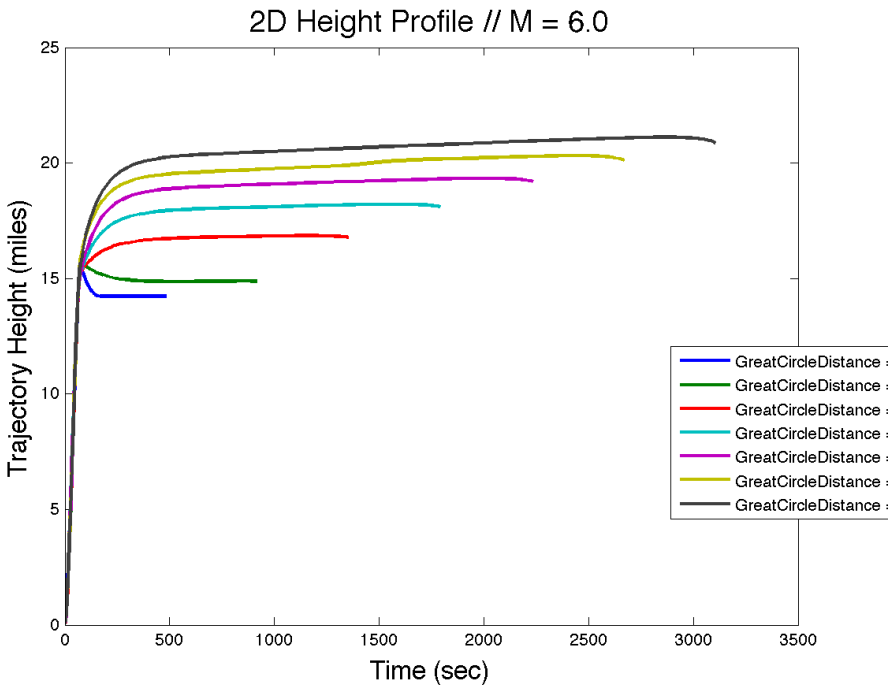

2D Height Profile // GreatCircleDistance = 2503 miles
Trajectory Height (miles)
Time (sec)
0
5
10
15
20
25
0
500
1000
1500
2000
2500
3000
M = 5
M = 5.5
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9

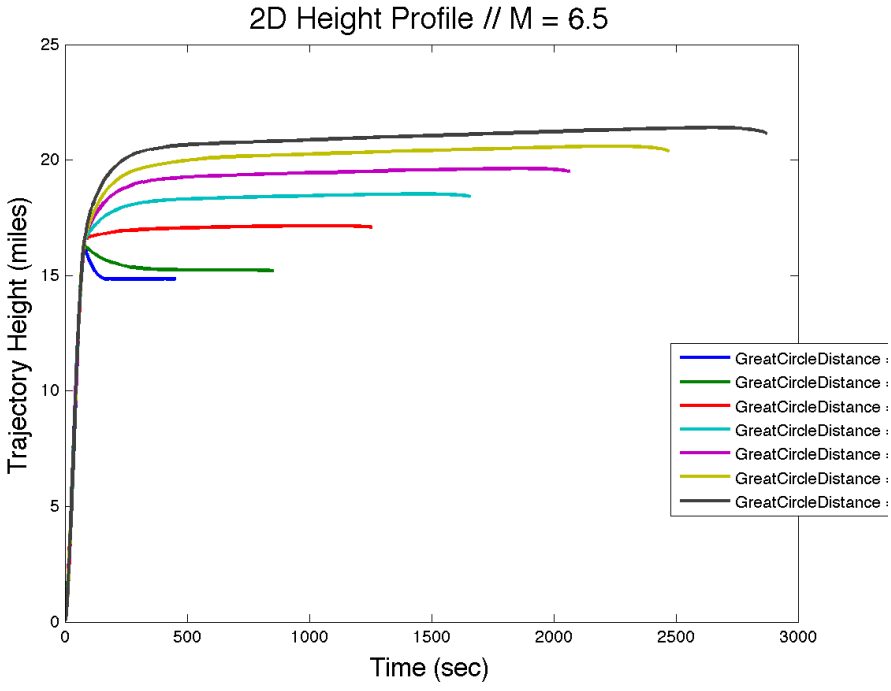

2D Height Profile // GreatCircleDistance = 3003 miles
Trajectory Height (miles)
Time (sec)
0
5
10
15
20
25
0
500
1000
1500
2000
2500
3000
3500
M = 5
M = 5.5
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9

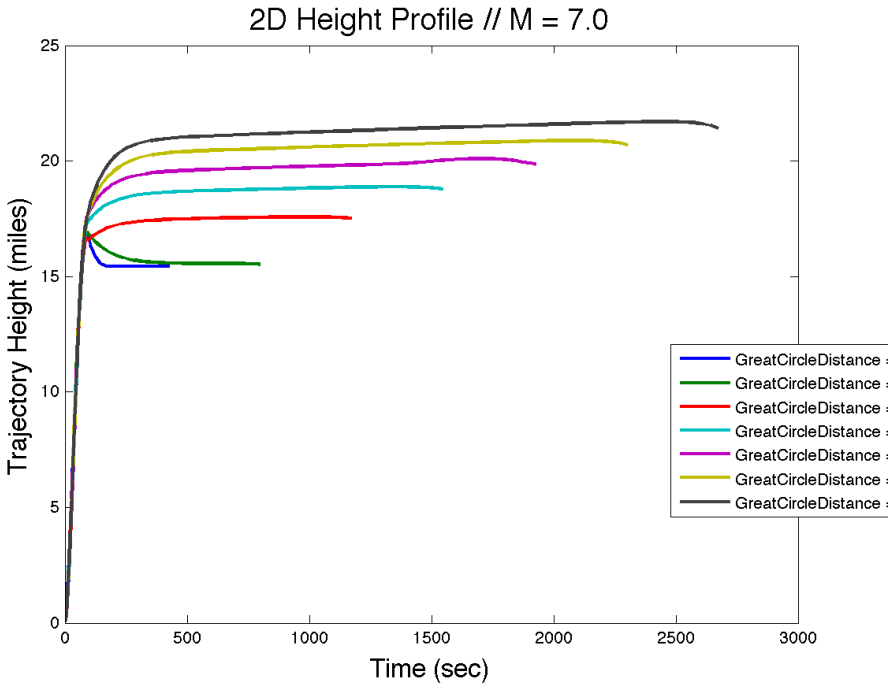

2D Height Profile // GreatCircleDistance = 3498 miles
Trajectory Height (miles)
Time (sec)
0
5
10
15
20
25
0
500
1000
1500
2000
2500
3000
3500
4000
M = 5
M = 5.5
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9

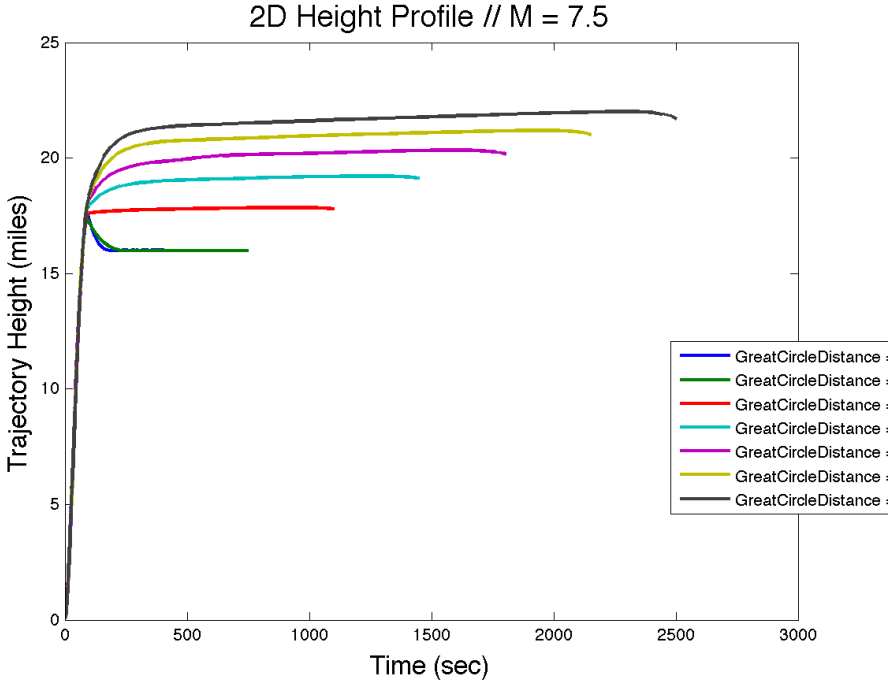

2D Height Profile // M = 5.0
Trajectory Height (miles)
Time (sec)
0
5
10
15
20
25
0
500
1000
1500
2000
2500
3000
3500
4000
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance

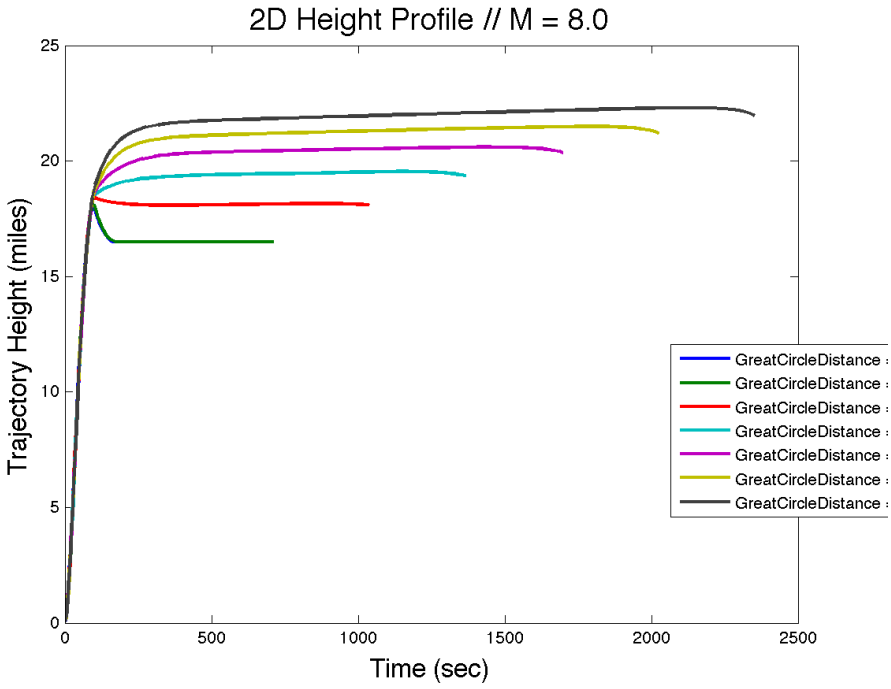

2D Height Profile // M = 5.5
Trajectory Height (miles)
Time (sec)
0
5
10
15
20
25
0
500
1000
1500
2000
2500
3000
3500
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance

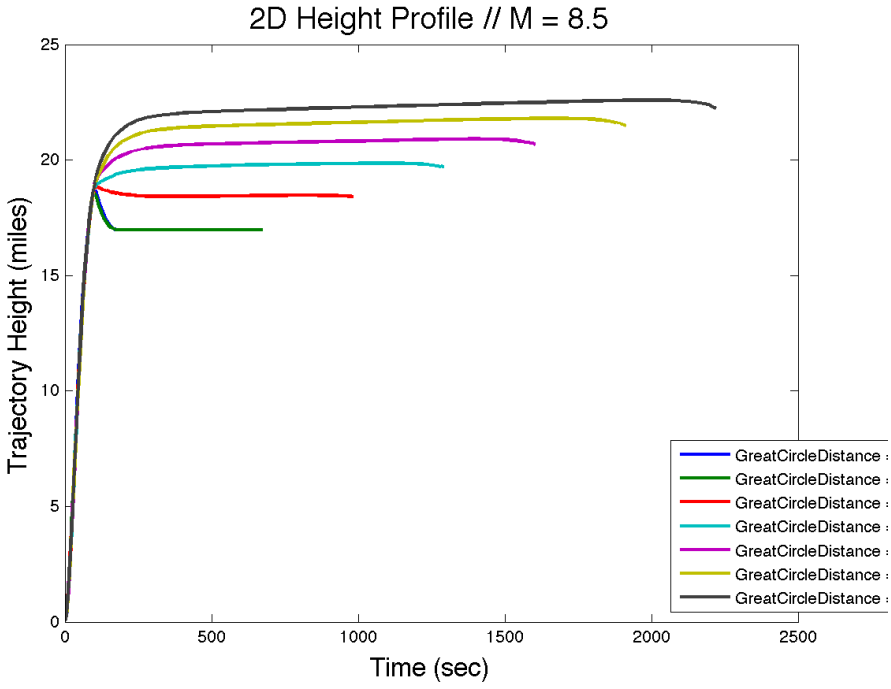

2D Height Profile // M = 6.0
Trajectory Height (miles)
Time (sec)
0
5
10
15
20
25
0
500
1000
1500
2000
2500
3000
3500
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :

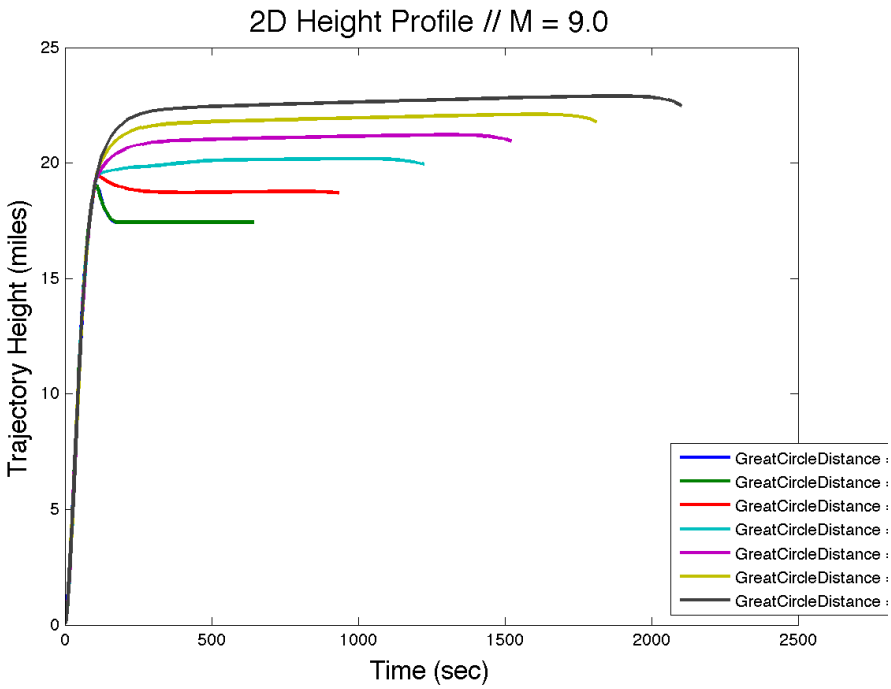

2D Height Profile // M = 6.5
Trajectory Height (miles)
Time (sec)
0
5
10
15
20
25
0
500
1000
1500
2000
2500
3000
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :


2D Height Profile // M = 7.0
Trajectory Height (miles)
Time (sec)
0
5
10
15
20
25
0
500
1000
1500
2000
2500
3000
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :

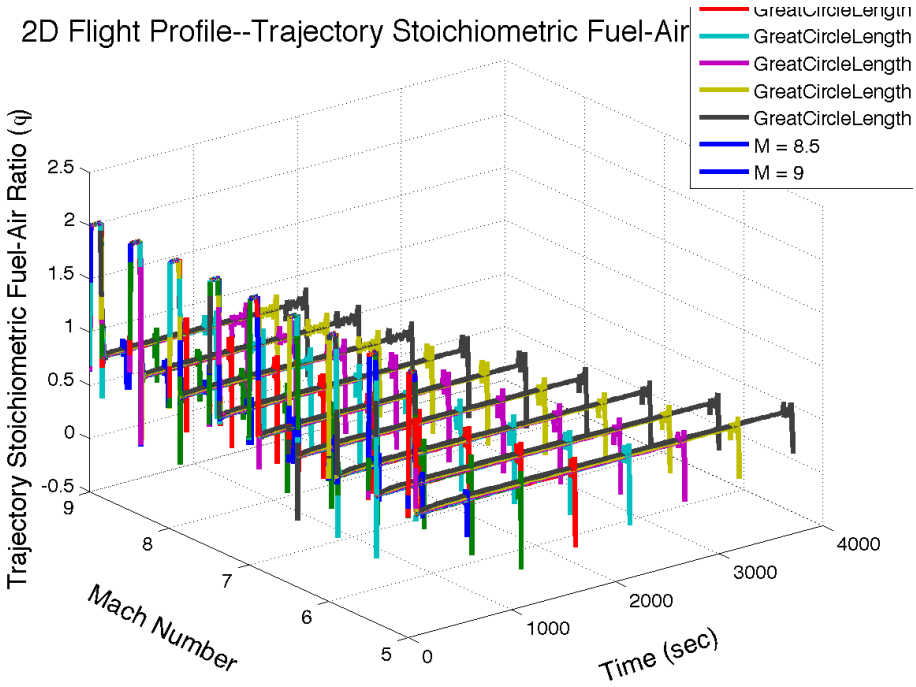

2D Height Profile // M = 7.5
Trajectory Height (miles)
Time (sec)
0
5
10
15
20
25
0
500
1000
1500
2000
2500
3000
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :

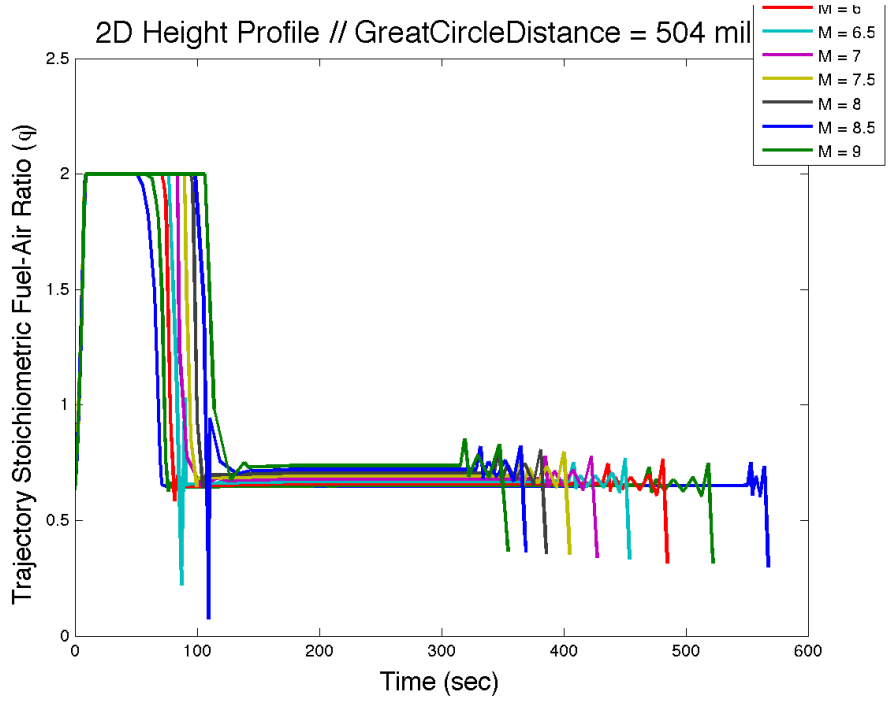

2D Height Profile // M = 8.0
Trajectory Height (miles)
Time (sec)
0
5
10
15
20
25
0
500
1000
1500
2000
2500
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :

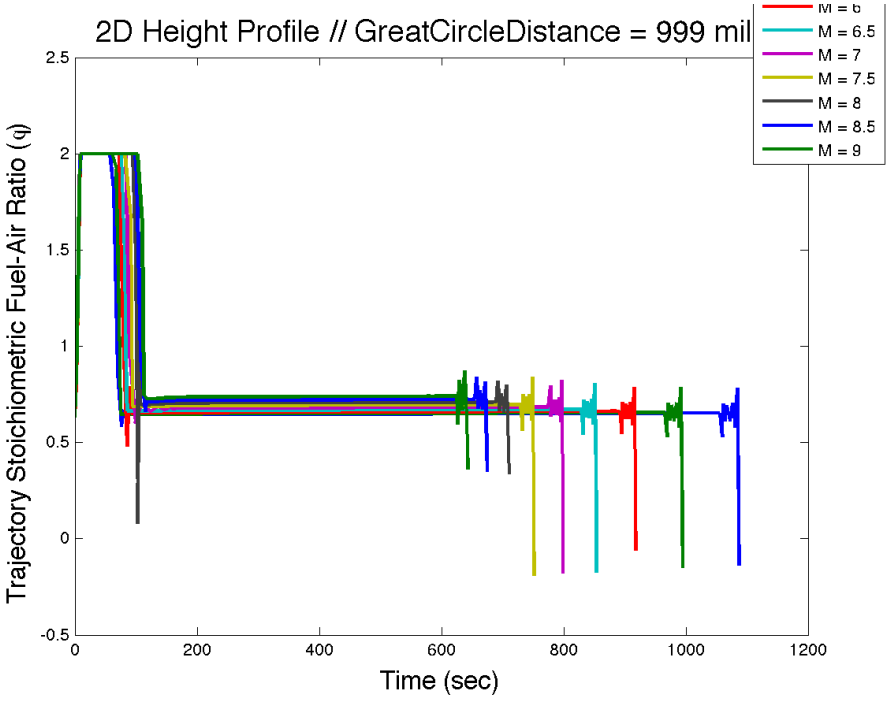

2D Height Profile // M = 8.5
Trajectory Height (miles)
Time (sec)
0
5
10
15
20
25
0
500
1000
1500
2000
2500
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :

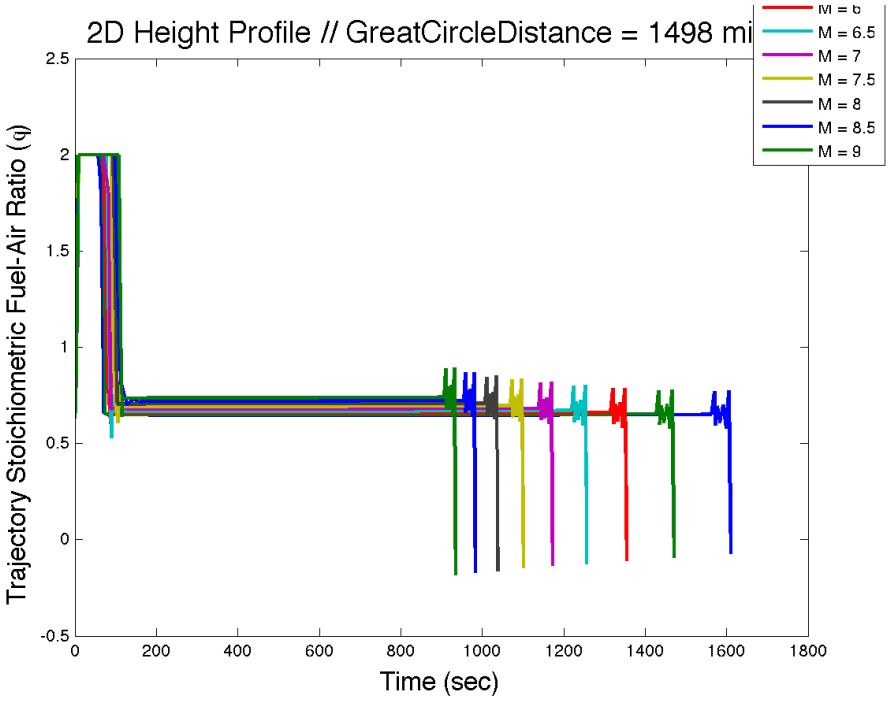


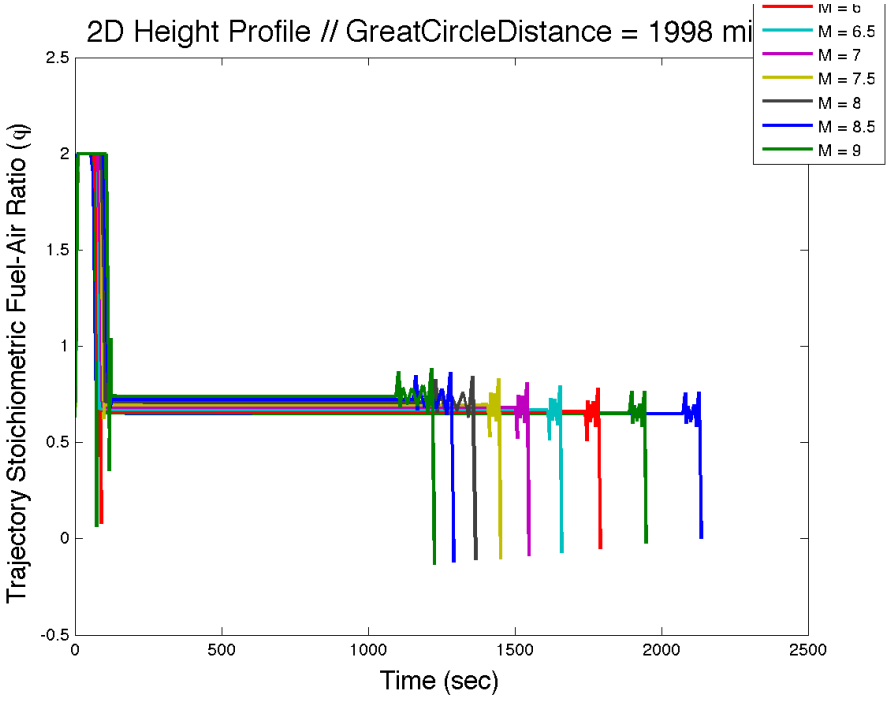
D.3 Stoichiometric Fuel-Air Ratio vs. Time

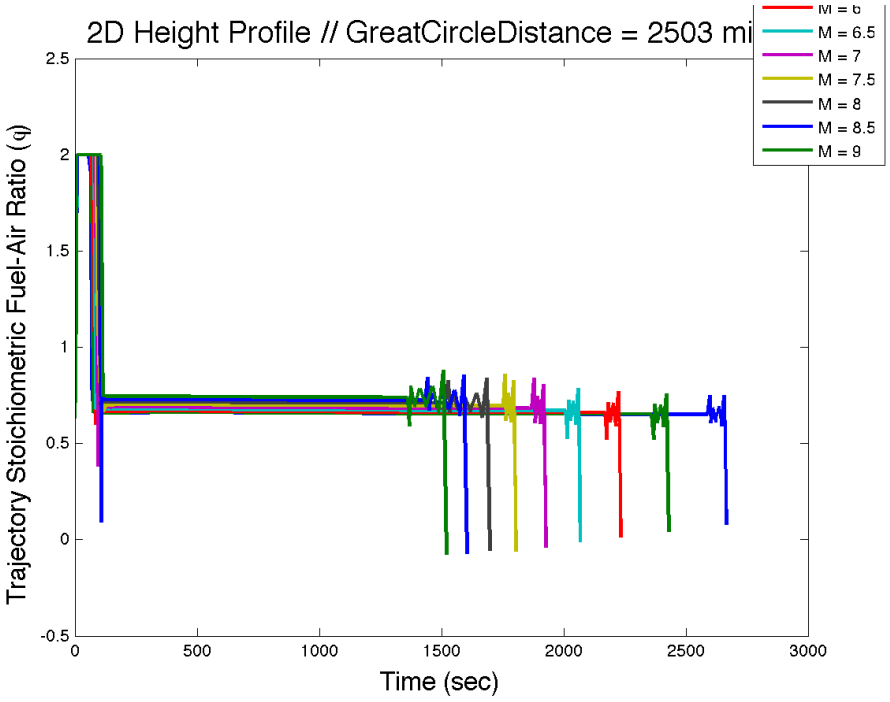


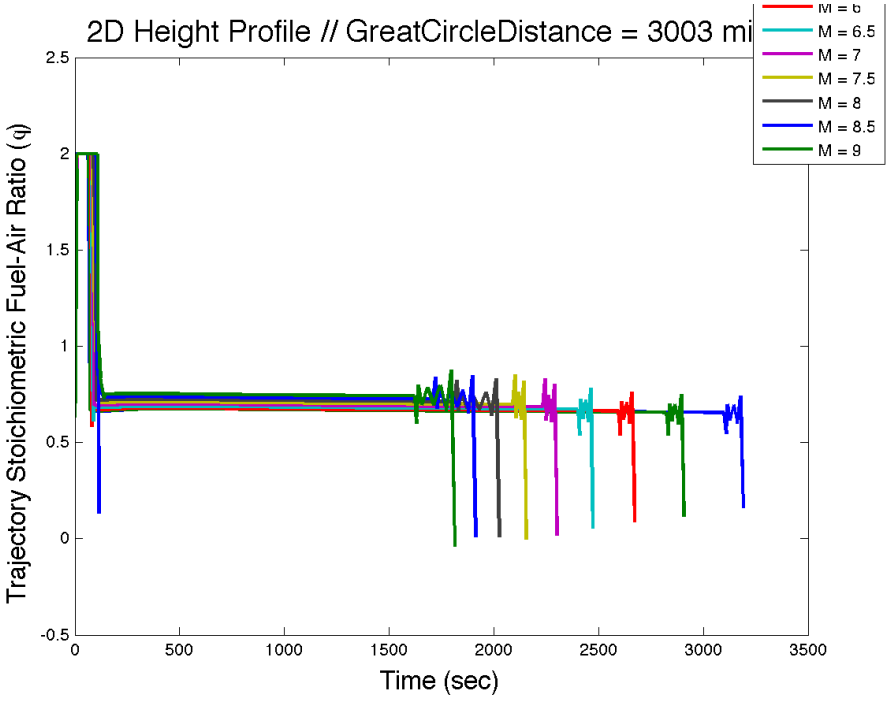

2D Height Profile // GreatCircleDistance = 504 mil
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9
Trajectory Stoichiometric Fuel-Air Ratio (ϕ)
Time (sec)
0
0.5
1
1.5
2
2.5
0
100
200
300
400
500
600

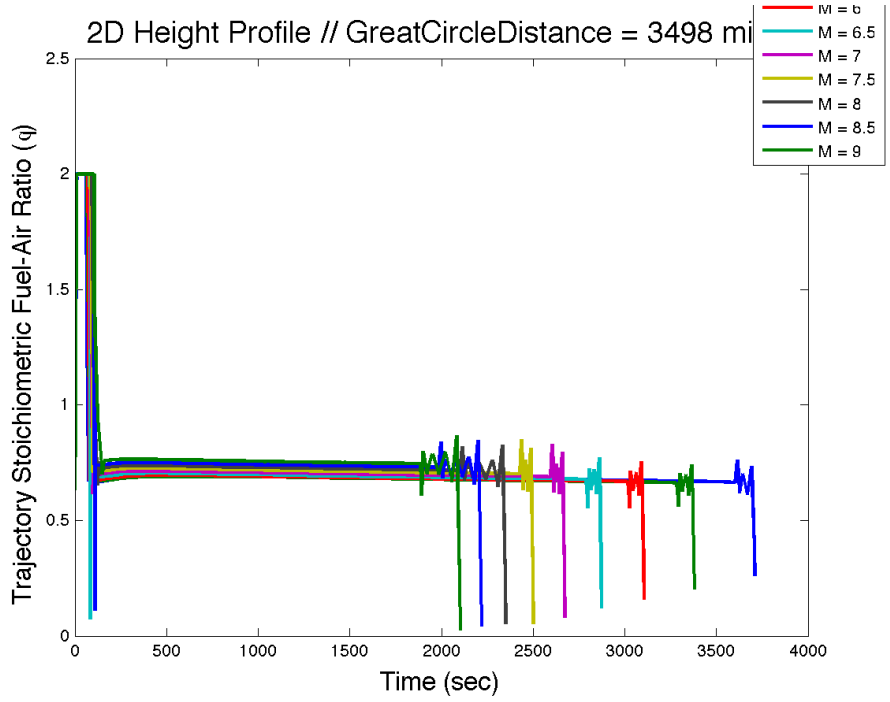

2D Height Profile // GreatCircleDistance = 999 mil
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9
Trajectory Stoichiometric Fuel-Air Ratio (ϕ)
Time (sec)
-0.5
0
0.5
1
1.5
2
2.5
0
200
400
600
800
1000
1200

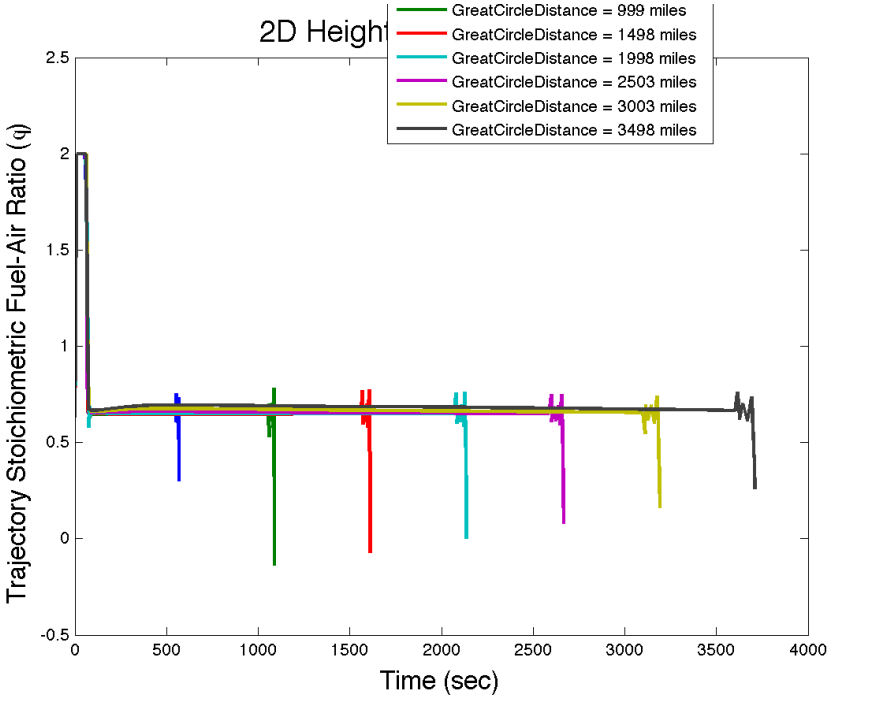

2D Height Profile // GreatCircleDistance = 1498 mi
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9
Trajectory Stoichiometric Fuel-Air Ratio (q)
Time (sec)
2.5
2
1.5
1
0.5
0
-0.5
0
200
400
600
800
1000
1200
1400
1600
1800

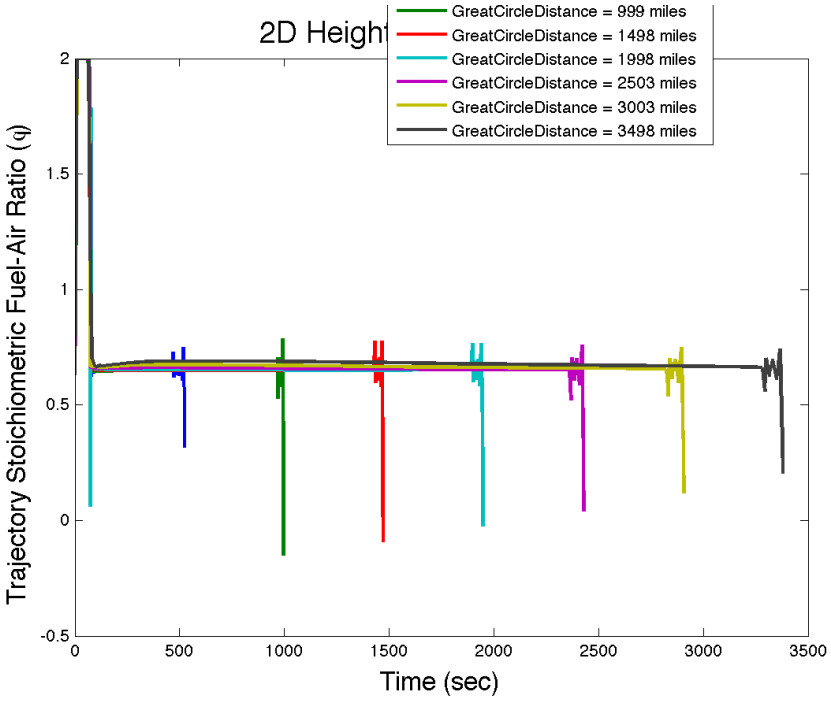

2D Height Profile // GreatCircleDistance = 1998 mi
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9
Trajectory Stoichiometric Fuel-Air Ratio (q)
Time (sec)
2.5
2
1.5
1
0.5
0
-0.5
0
500
1000
1500
2000
2500

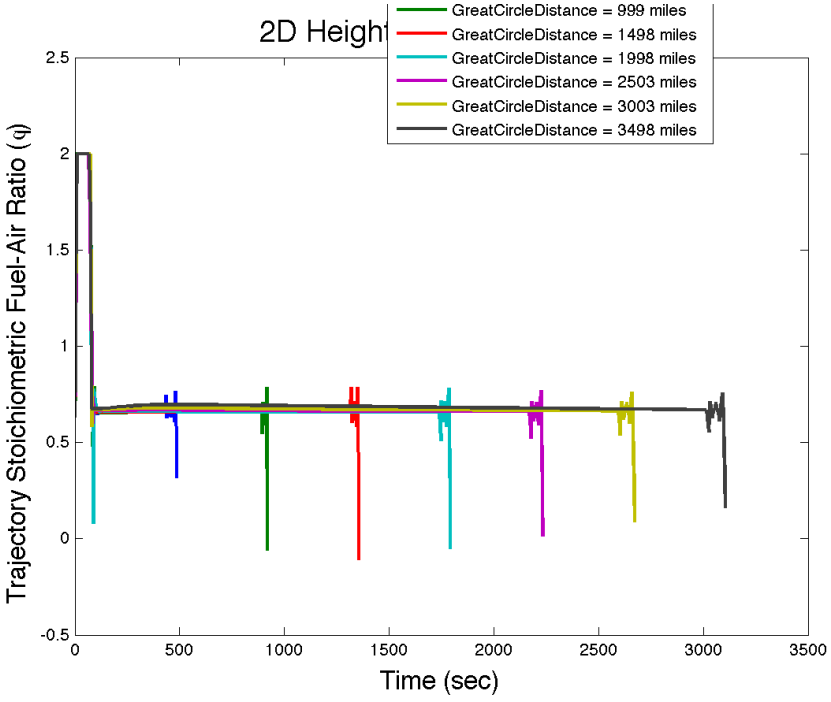

2D Height Profile // GreatCircleDistance = 2503 mi
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9
Trajectory Stoichiometric Fuel-Air Ratio (φ)
Time (sec)
2.5
2
1.5
1
0.5
0
-0.5
0
500
1000
1500
2000
2500
3000

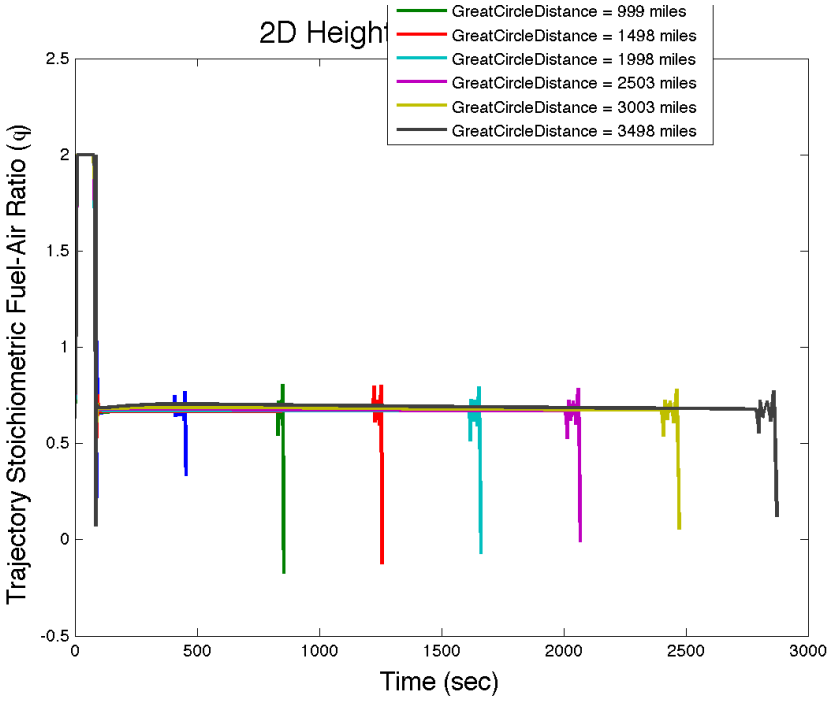

2D Height Profile // GreatCircleDistance = 3003 mi
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9
Trajectory Stoichiometric Fuel-Air Ratio (φ)
Time (sec)
2.5
2
1.5
1
0.5
0
-0.5
0
500
1000
1500
2000
2500
3000
3500

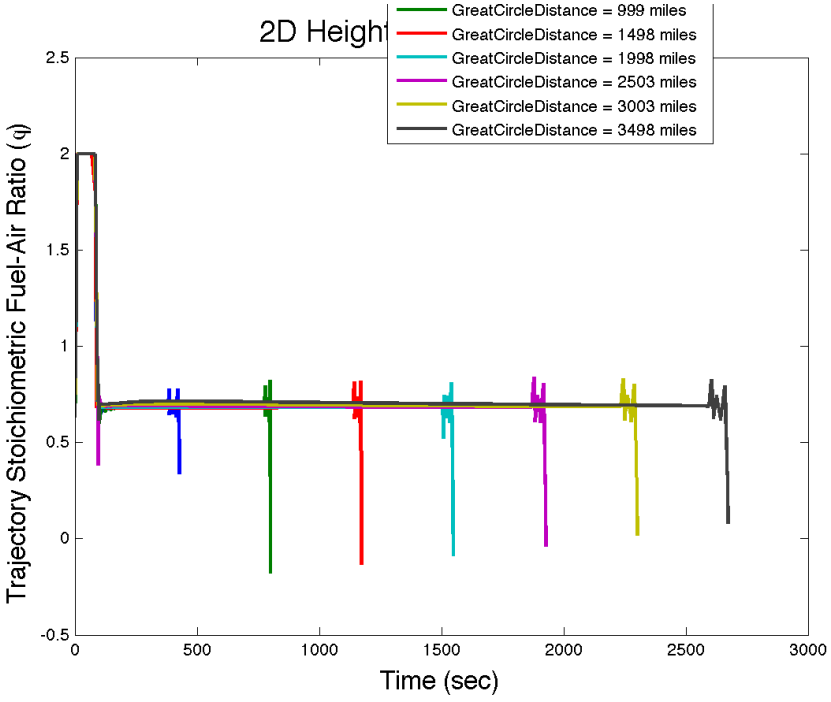

2D Height Profile // GreatCircleDistance = 3498 mi
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9
Trajectory Stoichiometric Fuel-Air Ratio (q)
Time (sec)
0
0.5
1
1.5
2
2.5
0
500
1000
1500
2000
2500
3000
3500
4000

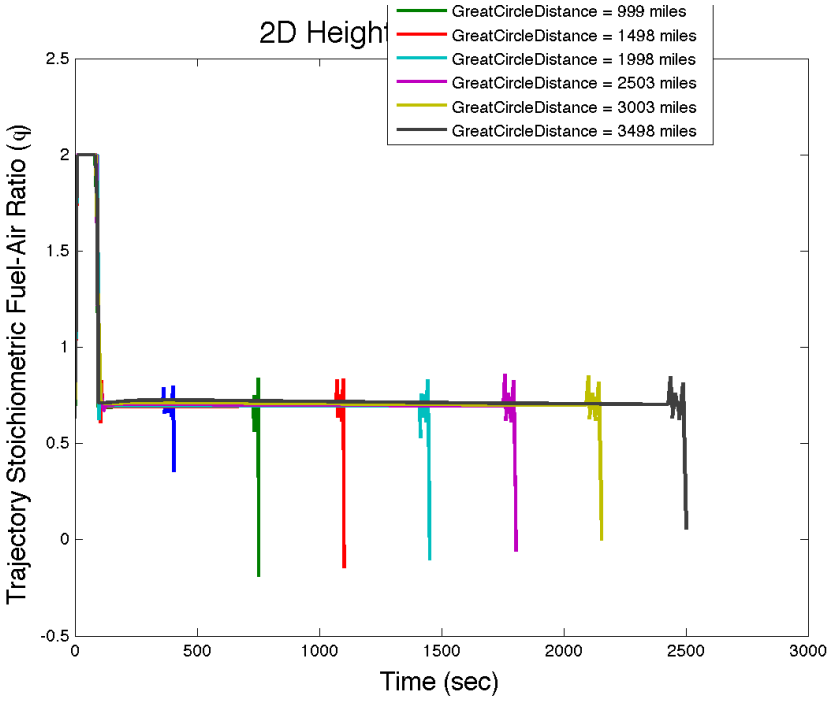

2D Height
GreatCircleDistance = 999 miles
GreatCircleDistance = 1498 miles
GreatCircleDistance = 1998 miles
GreatCircleDistance = 2503 miles
GreatCircleDistance = 3003 miles
GreatCircleDistance = 3498 miles
Trajectory Stoichiometric Fuel-Air Ratio (q)
Time (sec)
-0.5
0
0.5
1
1.5
2
2.5
0
500
1000
1500
2000
2500
3000
3500
4000

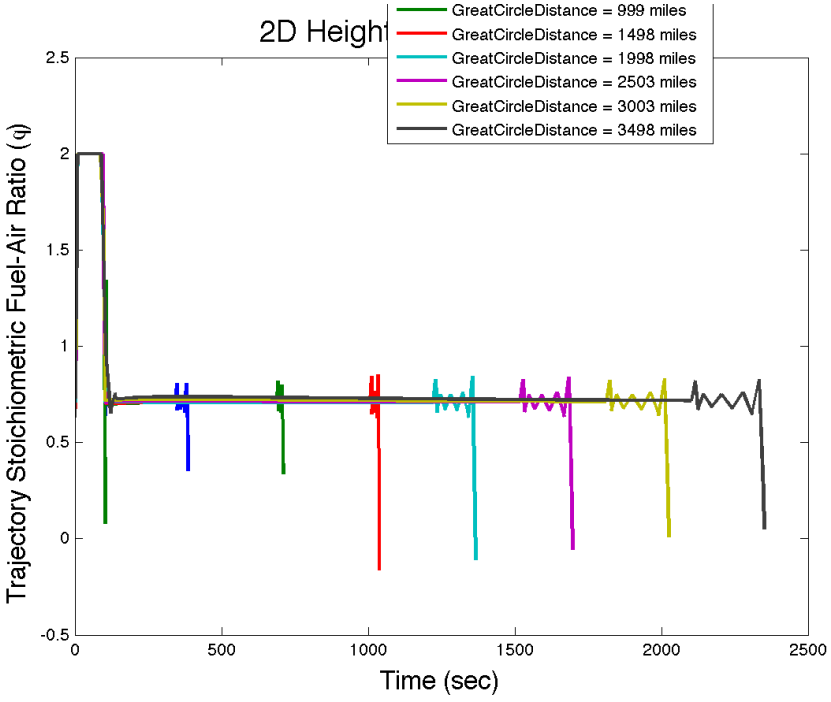

2D Height
GreatCircleDistance = 999 miles
GreatCircleDistance = 1498 miles
GreatCircleDistance = 1998 miles
GreatCircleDistance = 2503 miles
GreatCircleDistance = 3003 miles
GreatCircleDistance = 3498 miles
Trajectory Stoichiometric Fuel-Air Ratio (q)
Time (sec)
2
1.5
1
0.5
0
-0.5
0
500
1000
1500
2000
2500
3000
3500

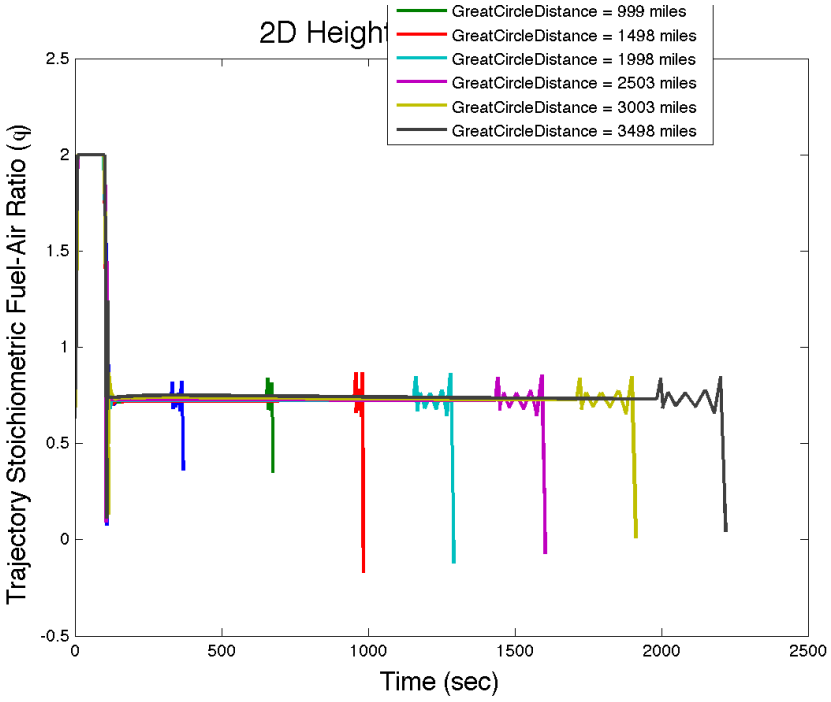

2D Height
GreatCircleDistance = 999 miles
GreatCircleDistance = 1498 miles
GreatCircleDistance = 1998 miles
GreatCircleDistance = 2503 miles
GreatCircleDistance = 3003 miles
GreatCircleDistance = 3498 miles
Trajectory Stoichiometric Fuel-Air Ratio (q)
Time (sec)
2.5
2
1.5
1
0.5
0
-0.5
0
500
1000
1500
2000
2500
3000
3500

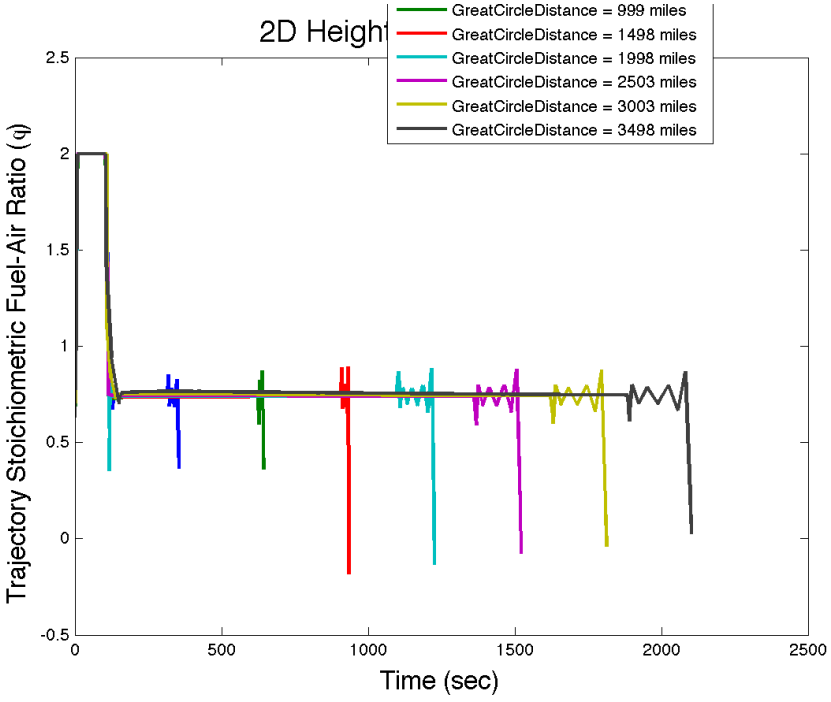

2D Height
GreatCircleDistance = 999 miles
GreatCircleDistance = 1498 miles
GreatCircleDistance = 1998 miles
GreatCircleDistance = 2503 miles
GreatCircleDistance = 3003 miles
GreatCircleDistance = 3498 miles
Trajectory Stoichiometric Fuel-Air Ratio (q)
Time (sec)
2.5
2
1.5
1
0.5
0
-0.5
0
500
1000
1500
2000
2500
3000


2D Height
GreatCircleDistance = 999 miles
GreatCircleDistance = 1498 miles
GreatCircleDistance = 1998 miles
GreatCircleDistance = 2503 miles
GreatCircleDistance = 3003 miles
GreatCircleDistance = 3498 miles
Trajectory Stoichiometric Fuel-Air Ratio (q)
Time (sec)
2.5
2
1.5
1
0.5
0
-0.5
0
500
1000
1500
2000
2500
3000

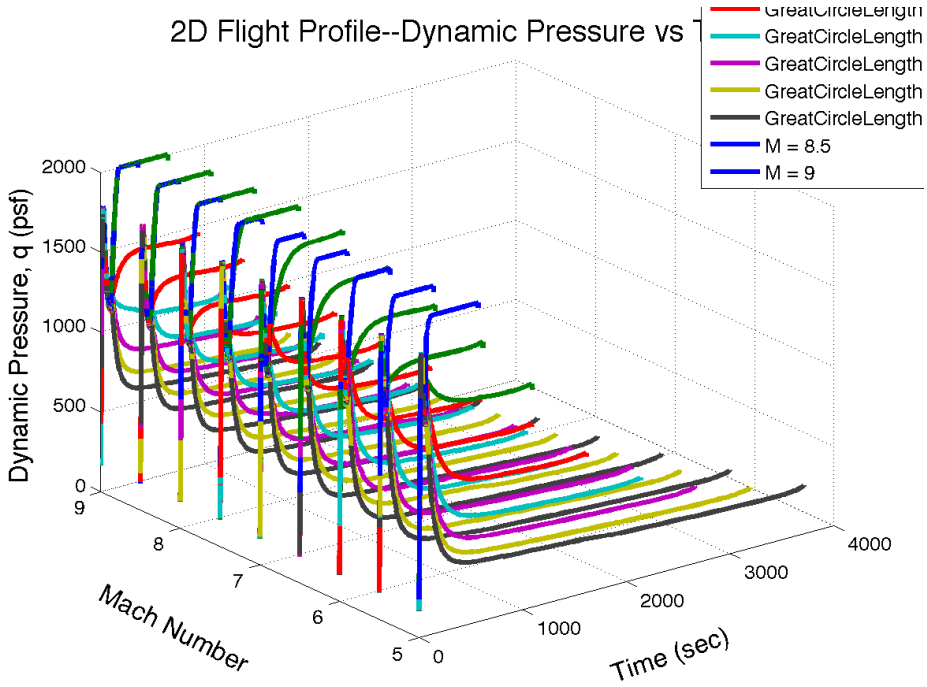

2D Height
GreatCircleDistance = 999 miles
GreatCircleDistance = 1498 miles
GreatCircleDistance = 1998 miles
GreatCircleDistance = 2503 miles
GreatCircleDistance = 3003 miles
GreatCircleDistance = 3498 miles
Trajectory Stoichiometric Fuel-Air Ratio (φ)
Time (sec)
2.5
2
1.5
1
0.5
0
-0.5
0
500
1000
1500
2000
2500
3000

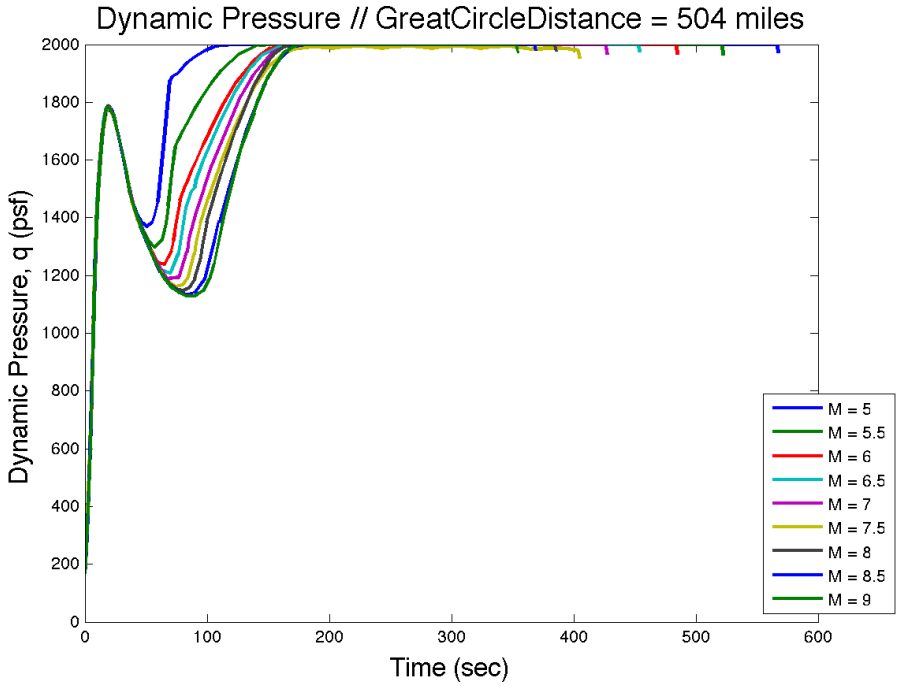

2D Height
GreatCircleDistance = 999 miles
GreatCircleDistance = 1498 miles
GreatCircleDistance = 1998 miles
GreatCircleDistance = 2503 miles
GreatCircleDistance = 3003 miles
GreatCircleDistance = 3498 miles
Trajectory Stoichiometric Fuel-Air Ratio (φ)
Time (sec)
2.5
2
1.5
1
0.5
0
-0.5
0
500
1000
1500
2000
2500

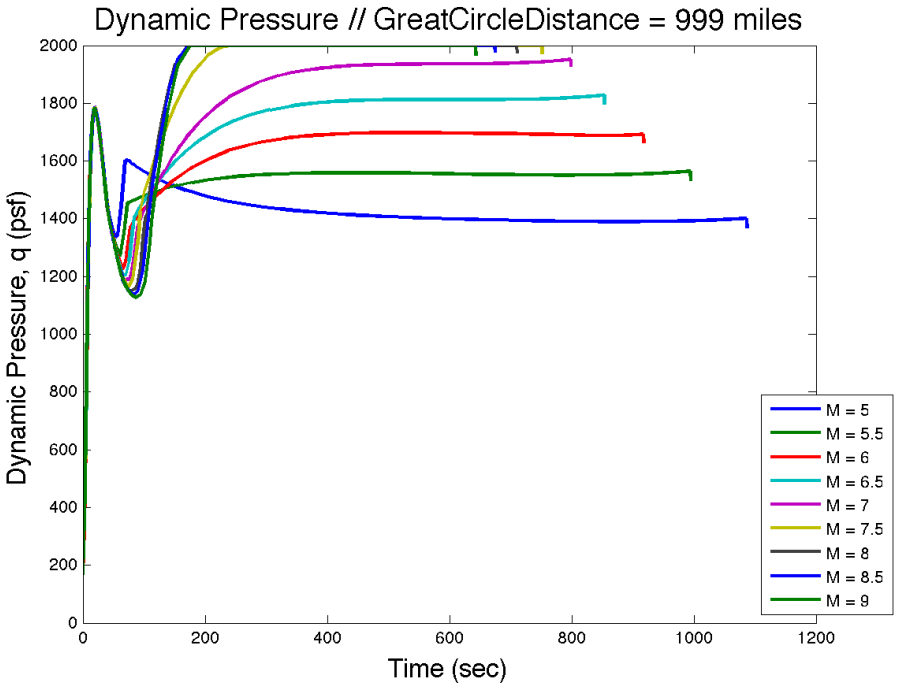

2D Height
GreatCircleDistance = 999 miles
GreatCircleDistance = 1498 miles
GreatCircleDistance = 1998 miles
GreatCircleDistance = 2503 miles
GreatCircleDistance = 3003 miles
GreatCircleDistance = 3498 miles
Trajectory Stoichiometric Fuel-Air Ratio (q)
Time (sec)
2.5
2
1.5
1
0.5
0
-0.5
0
500
1000
1500
2000
2500

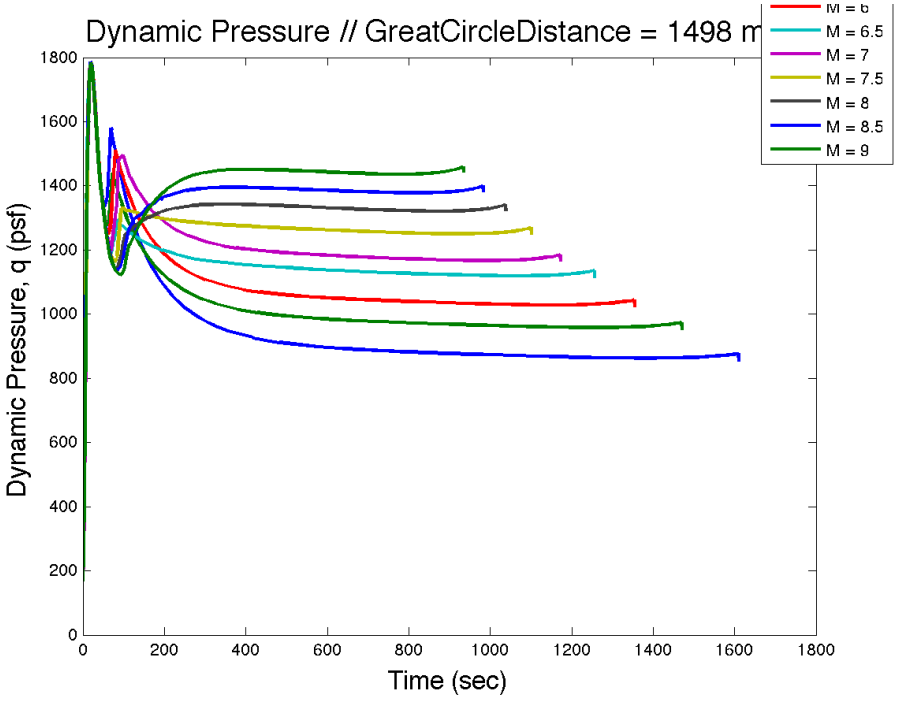

2D Height
GreatCircleDistance = 999 miles
GreatCircleDistance = 1498 miles
GreatCircleDistance = 1998 miles
GreatCircleDistance = 2503 miles
GreatCircleDistance = 3003 miles
GreatCircleDistance = 3498 miles
Trajectory Stoichiometric Fuel-Air Ratio (q)
Time (sec)
2.5
2
1.5
1
0.5
0
-0.5
0
500
1000
1500
2000
2500

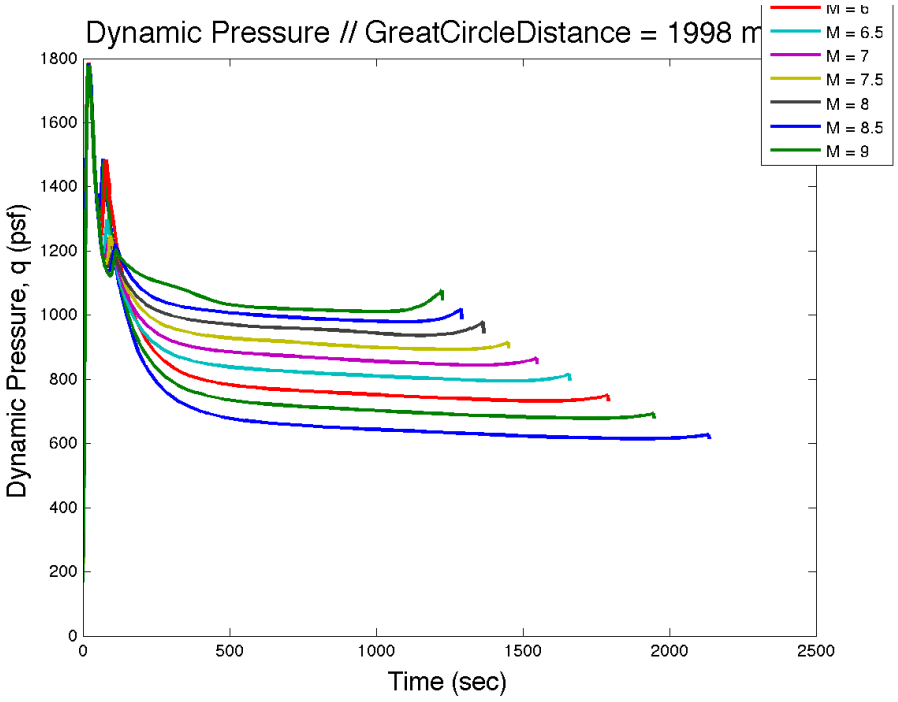
D.4 Dynamic Pressure vs. Time

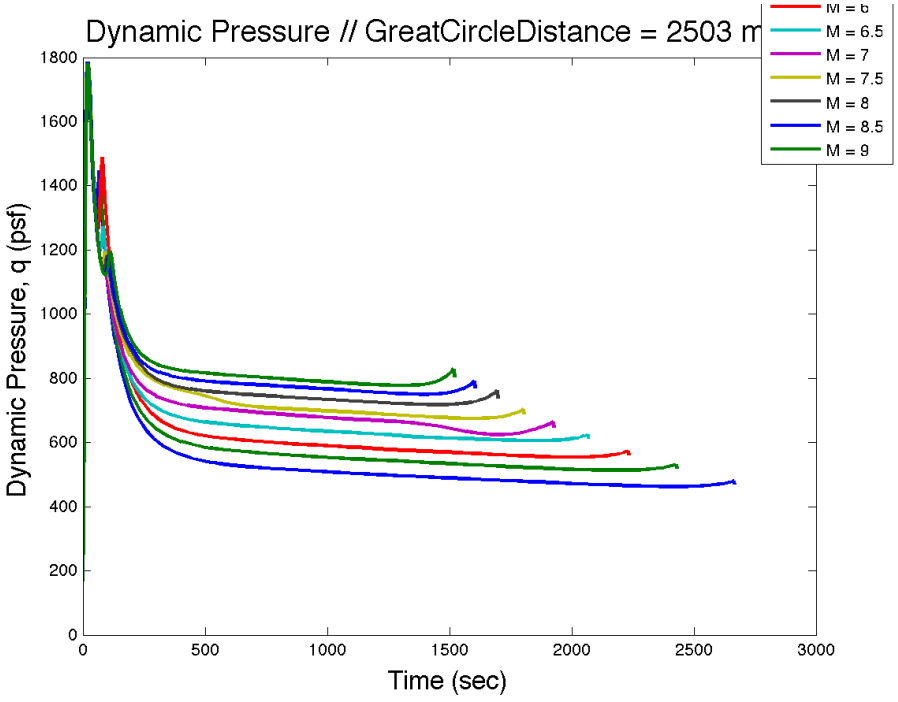


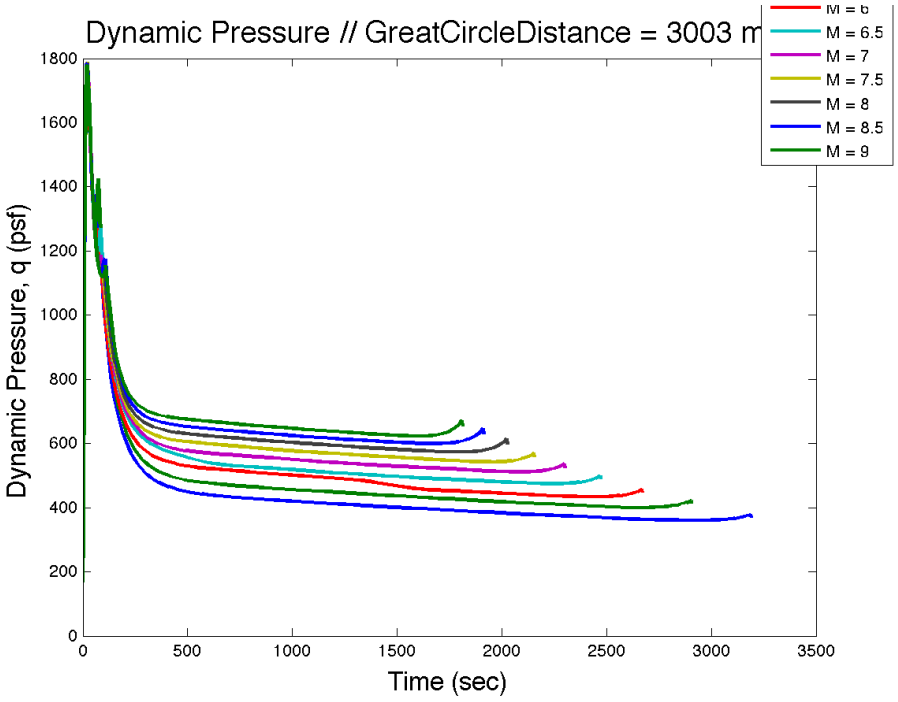


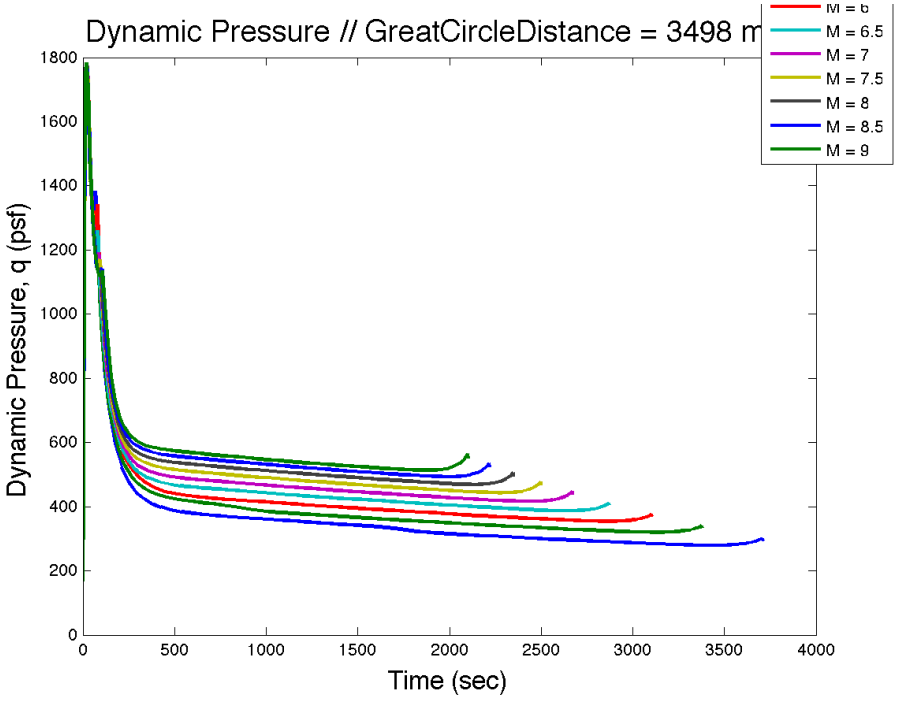

Dynamic Pressure // GreatCircleDistance = 999 miles
Dynamic Pressure, q (psf)
Time (sec)
0
200
400
600
800
1000
1200
1400
1600
1800
2000
M = 5
M = 5.5
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9

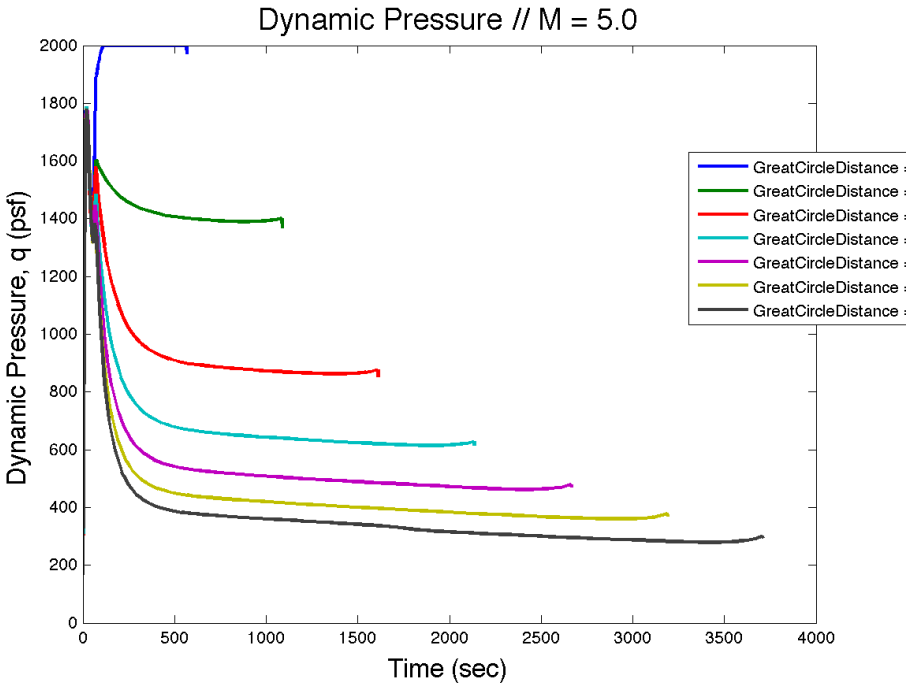

Dynamic Pressure // GreatCircleDistance = 1498 m
Dynamic Pressure, q (psf)
Time (sec)
0
200
400
600
800
1000
1200
1400
1600
1800
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9

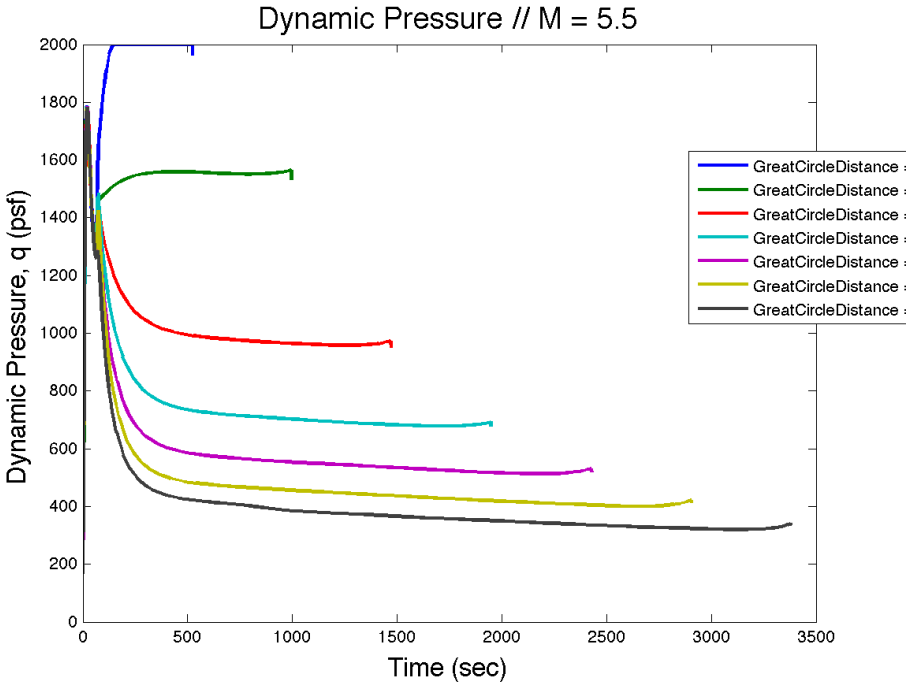

Dynamic Pressure // GreatCircleDistance = 1998 m
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9
Dynamic Pressure, q (psf)
Time (sec)
0
200
400
600
800
1000
1200
1400
1600
1800
0
500
1000
1500
2000
2500

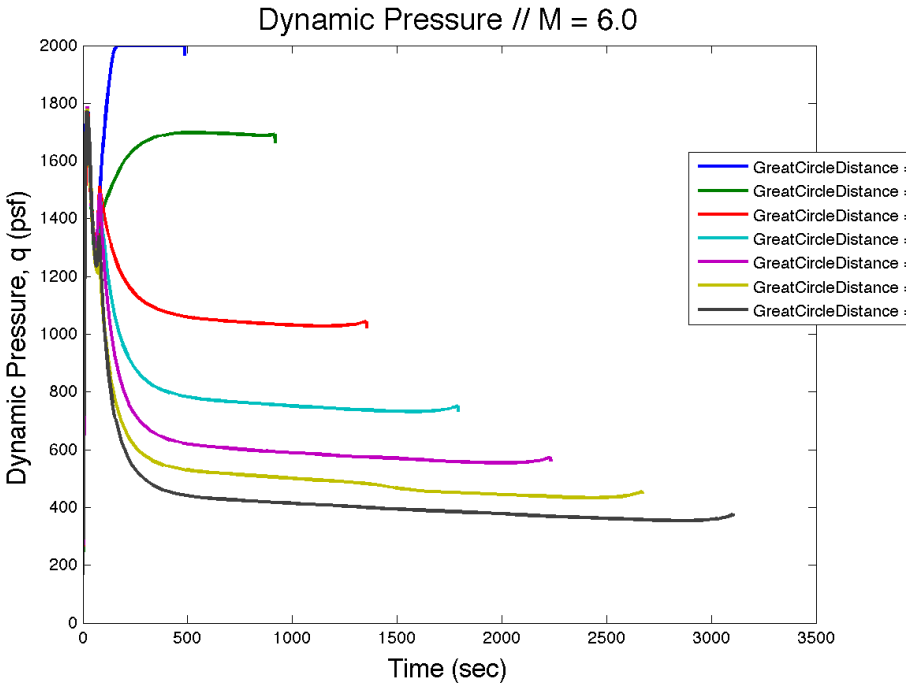

Dynamic Pressure // GreatCircleDistance = 2503 m
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9
Dynamic Pressure, q (psf)
Time (sec)
0
200
400
600
800
1000
1200
1400
1600
1800
0
500
1000
1500
2000
2500
3000

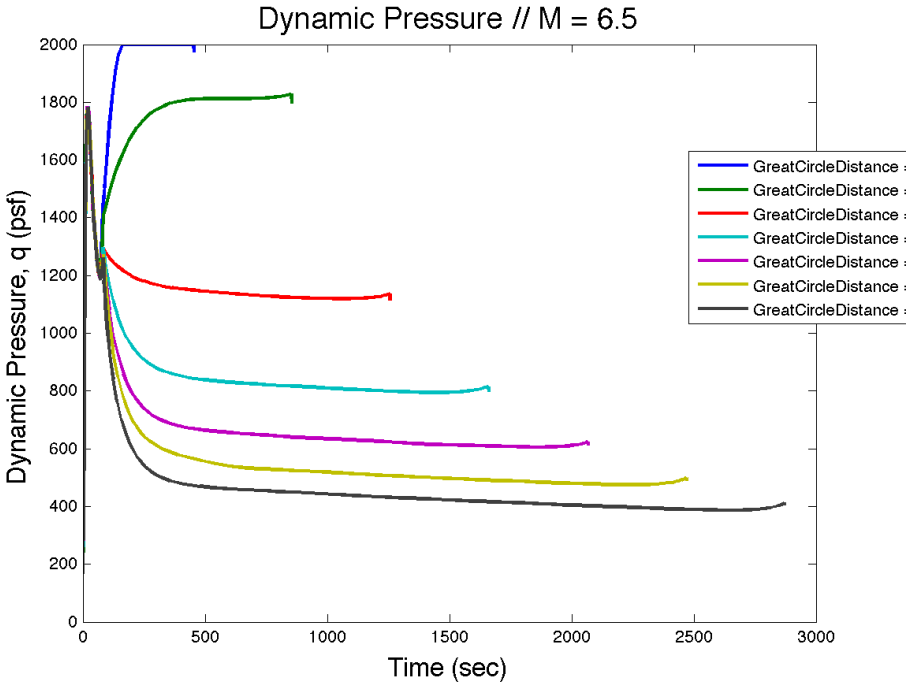

Dynamic Pressure // GreatCircleDistance = 3003 m
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9
Dynamic Pressure, q (psf)
Time (sec)
0
200
400
600
800
1000
1200
1400
1600
1800
0
500
1000
1500
2000
2500
3000
3500

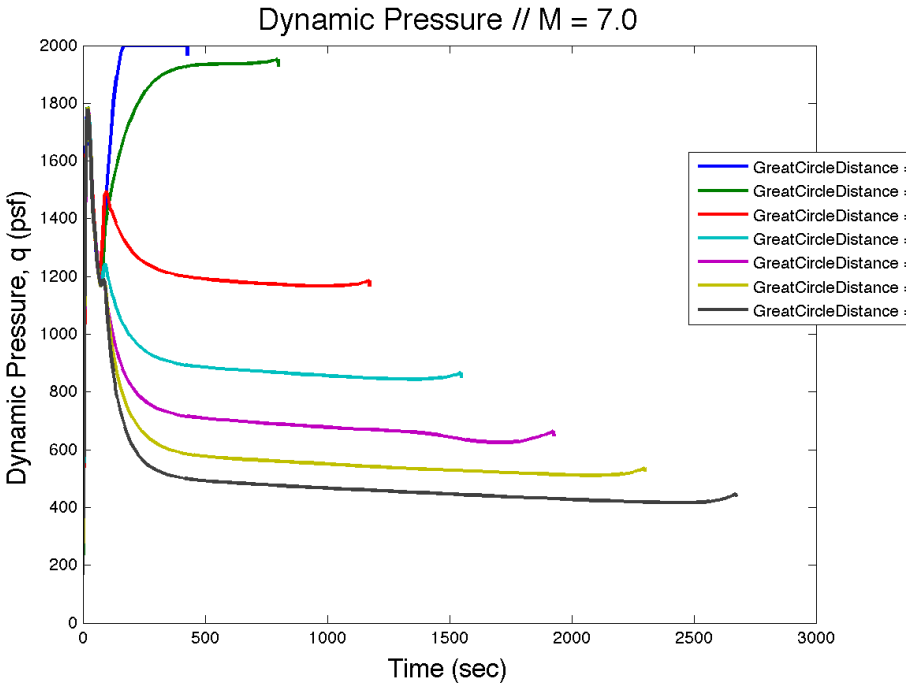

Dynamic Pressure // GreatCircleDistance = 3498 m
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9
Dynamic Pressure, q (psf)
Time (sec)
0
200
400
600
800
1000
1200
1400
1600
1800
0
500
1000
1500
2000
2500
3000
3500
4000

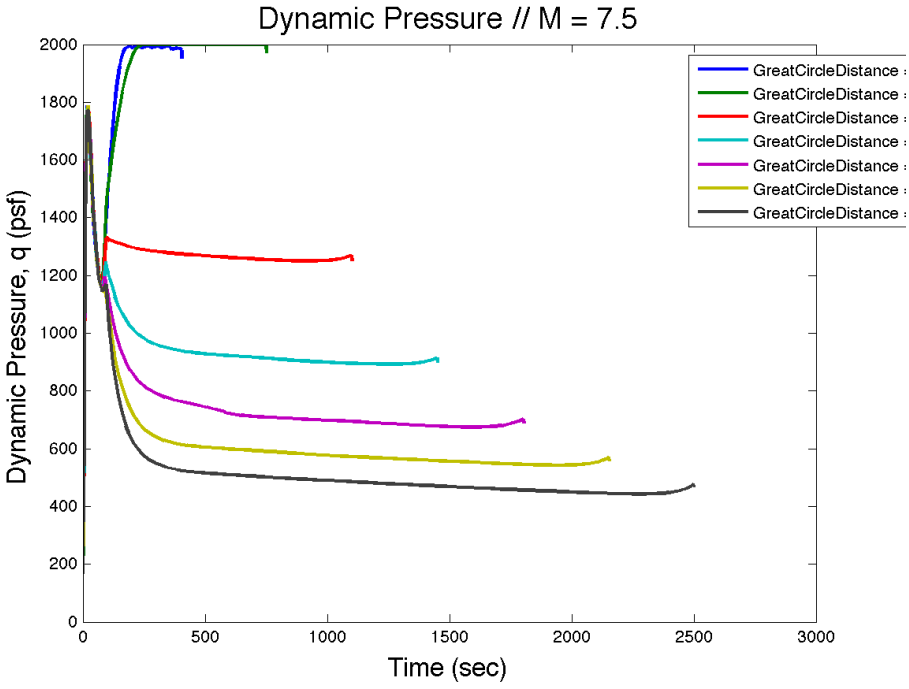

Dynamic Pressure // M = 5.0
Dynamic Pressure, q (psf)
Time (sec)
0
200
400
600
800
1000
1200
1400
1600
1800
2000
0
500
1000
1500
2000
2500
3000
3500
4000
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance

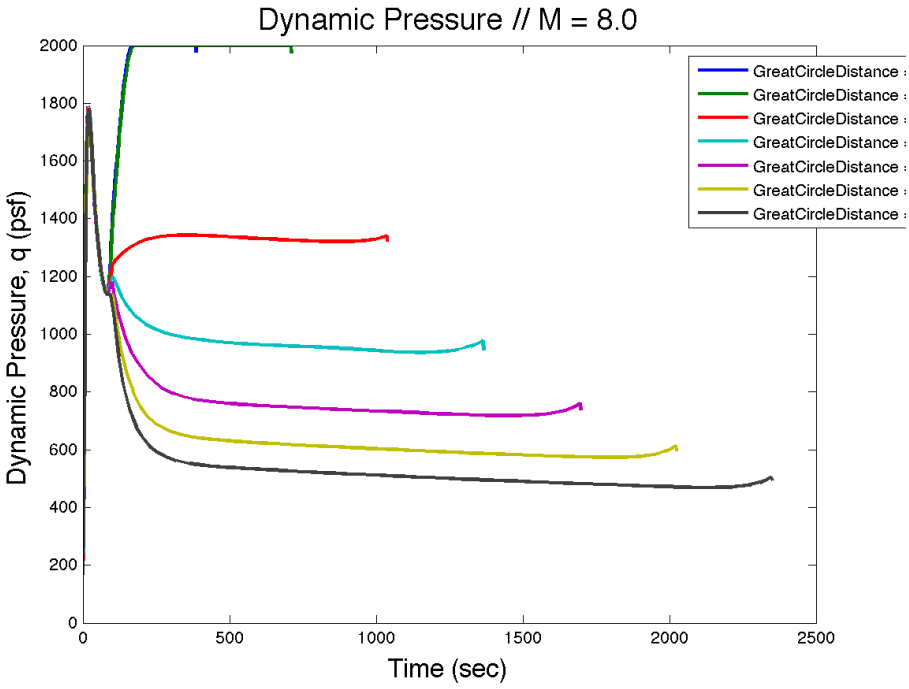

Dynamic Pressure // M = 5.5
Dynamic Pressure, q (psf)
Time (sec)
0
200
400
600
800
1000
1200
1400
1600
1800
2000
0
500
1000
1500
2000
2500
3000
3500
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance

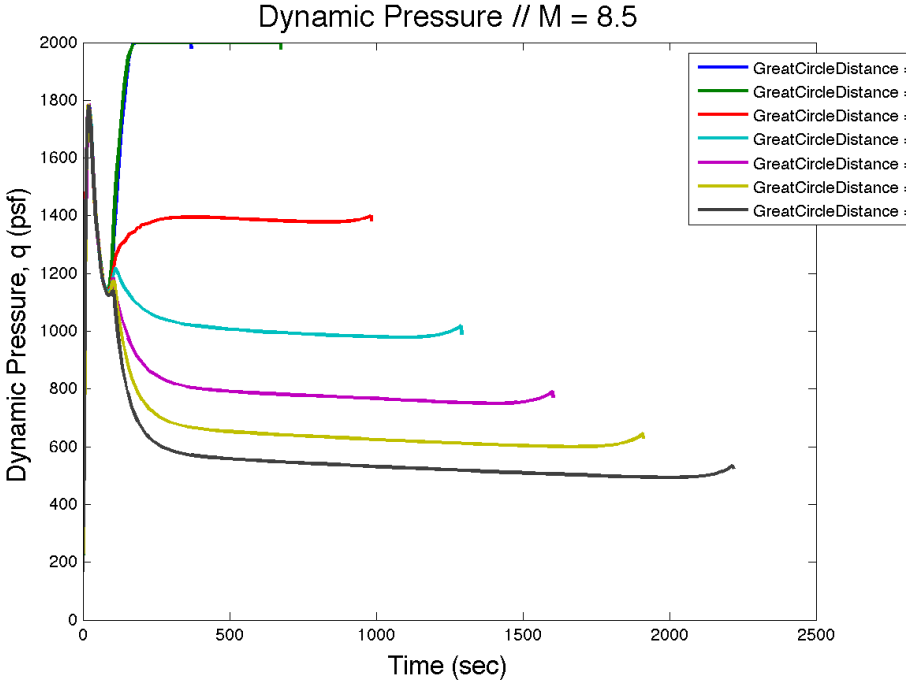

Dynamic Pressure // M = 6.0
Dynamic Pressure, q (psf)
Time (sec)
0
200
400
600
800
1000
1200
1400
1600
1800
2000
0
500
1000
1500
2000
2500
3000
3500
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance

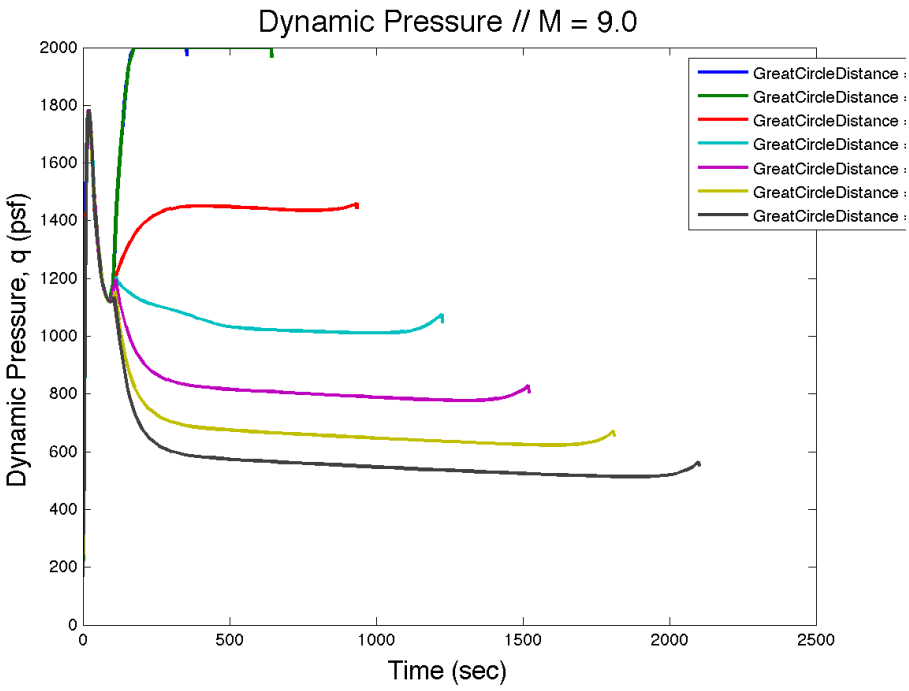

Dynamic Pressure // M = 6.5
Dynamic Pressure, q (psf)
Time (sec)
0
200
400
600
800
1000
1200
1400
1600
1800
2000
0
500
1000
1500
2000
2500
3000
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance


Dynamic Pressure // M = 7.0
Dynamic Pressure, q (psf)
Time (sec)
0
200
400
600
800
1000
1200
1400
1600
1800
2000
0
500
1000
1500
2000
2500
3000
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance

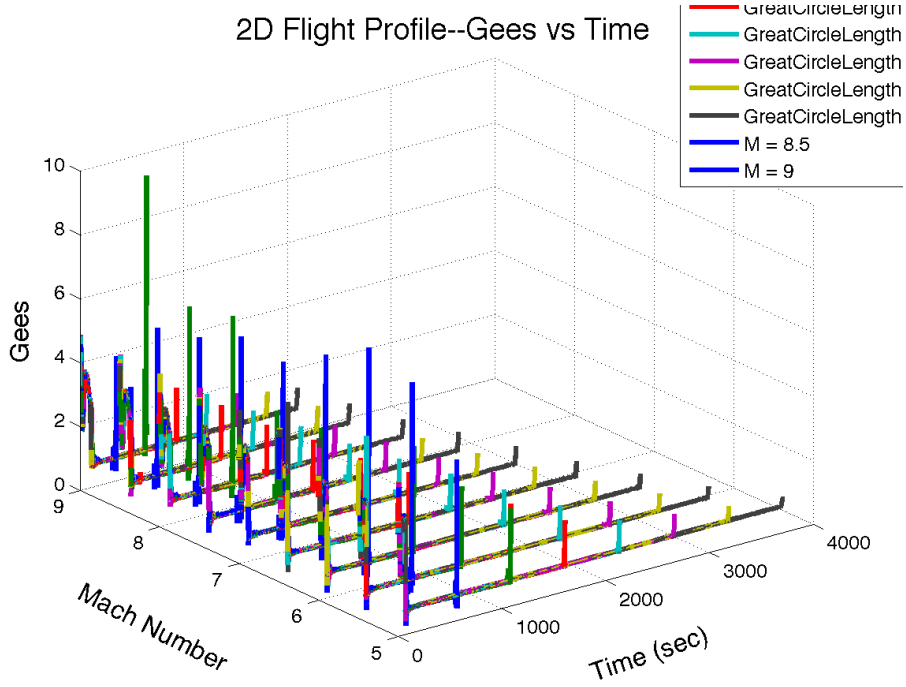

Dynamic Pressure // M = 7.5
Dynamic Pressure, q (psf)
Time (sec)
0
200
400
600
800
1000
1200
1400
1600
1800
2000
0
500
1000
1500
2000
2500
3000
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance

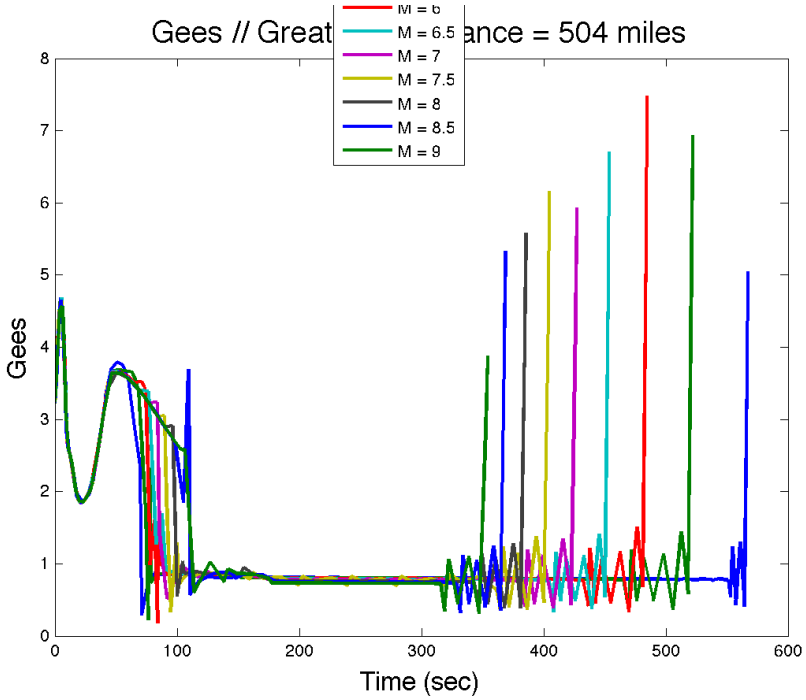

Dynamic Pressure // M = 8.0
Dynamic Pressure, q (psf)
Time (sec)
0
200
400
600
800
1000
1200
1400
1600
1800
2000
0
500
1000
1500
2000
2500
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance

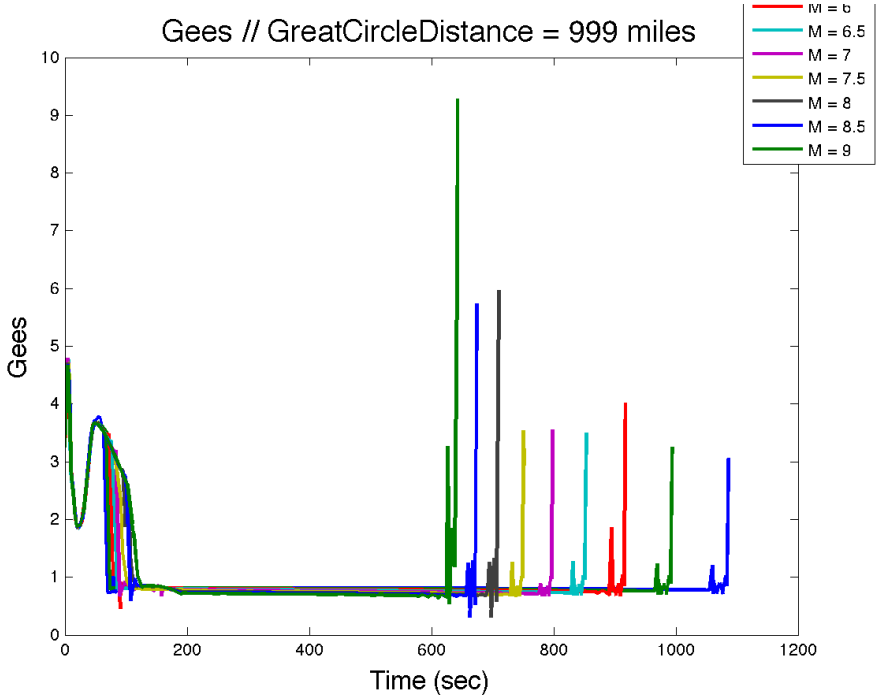

Dynamic Pressure // M = 8.5
Dynamic Pressure, q (psf)
Time (sec)
0
200
400
600
800
1000
1200
1400
1600
1800
2000
0
500
1000
1500
2000
2500
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance

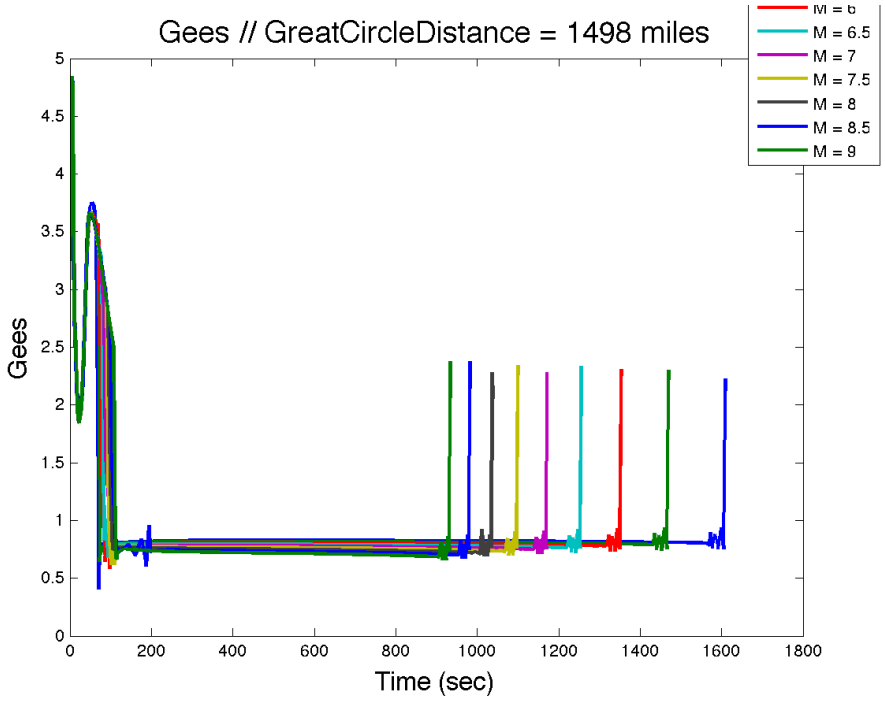

Dynamic Pressure // M = 9.0
Dynamic Pressure, q (psf)
Time (sec)
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
0
200
400
600
800
1000
1200
1400
1600
1800
2000
0
500
1000
1500
2000
2500

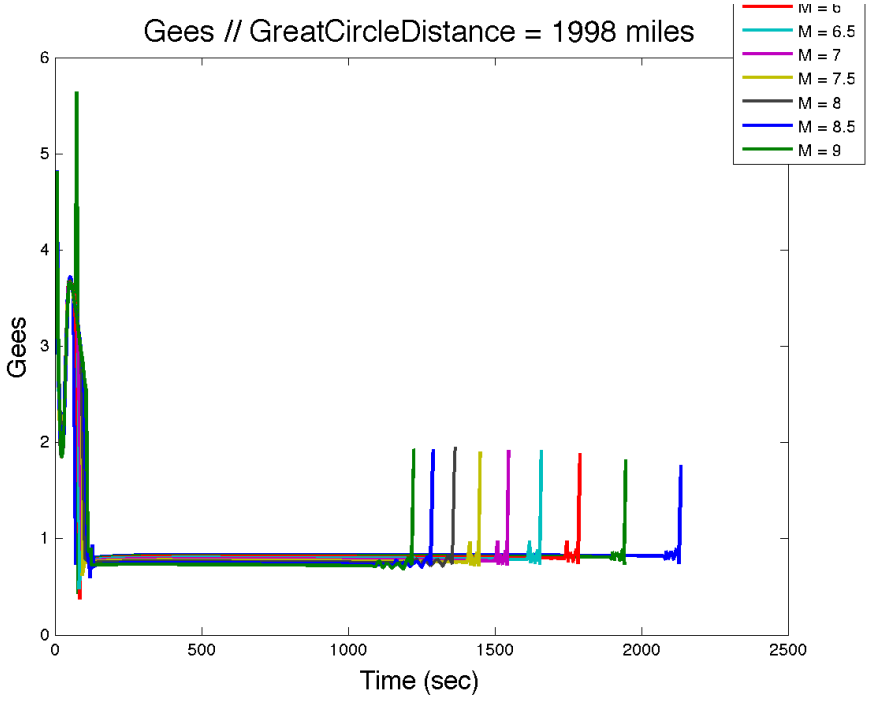
D.5 g's vs. Time

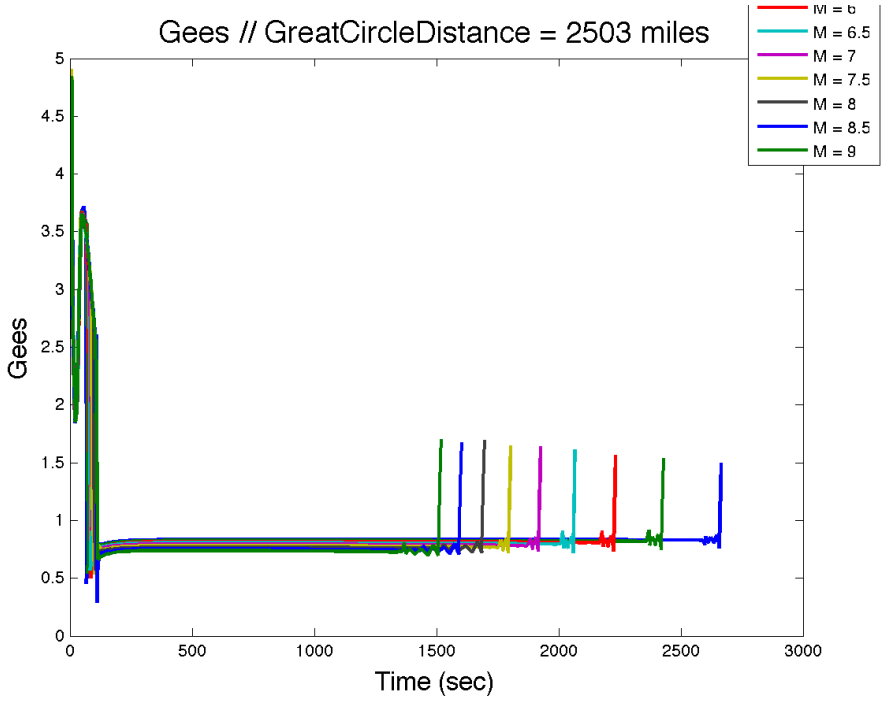

2D Flight Profile--Gees vs Time
GreatCircleLength
GreatCircleLength
GreatCircleLength
GreatCircleLength
GreatCircleLength
M = 8.5
M = 9
Gees
Mach Number
Time (sec)
0
2
4
6
8
10
9
8
7
6
5
0
1000
2000
3000
4000

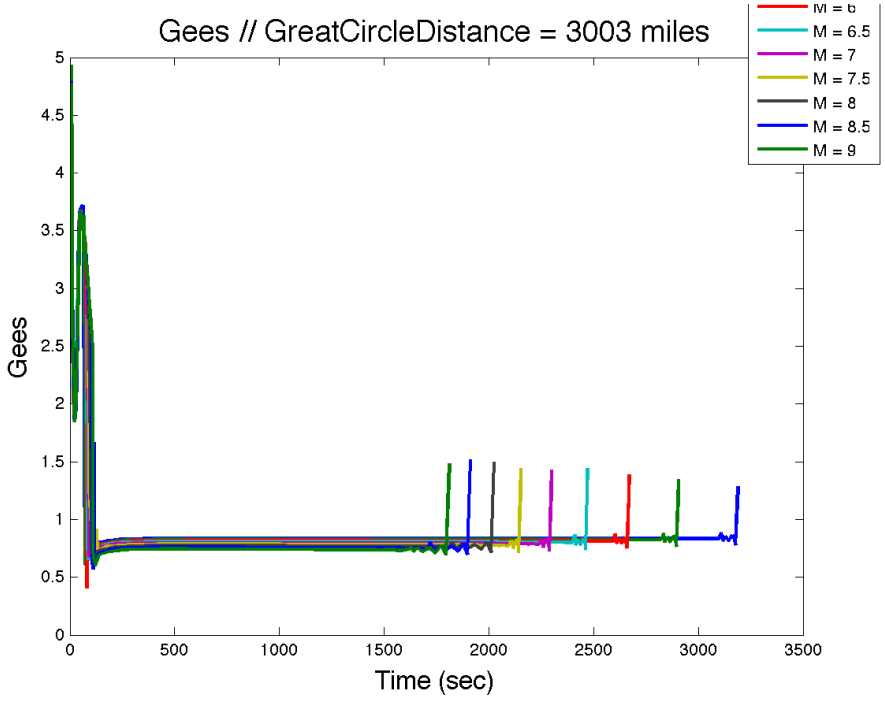

Gees // Great ance = 504 miles
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9
Gees
Time (sec)
0
1
2
3
4
5
6
7
8
0
100
200
300
400
500
600

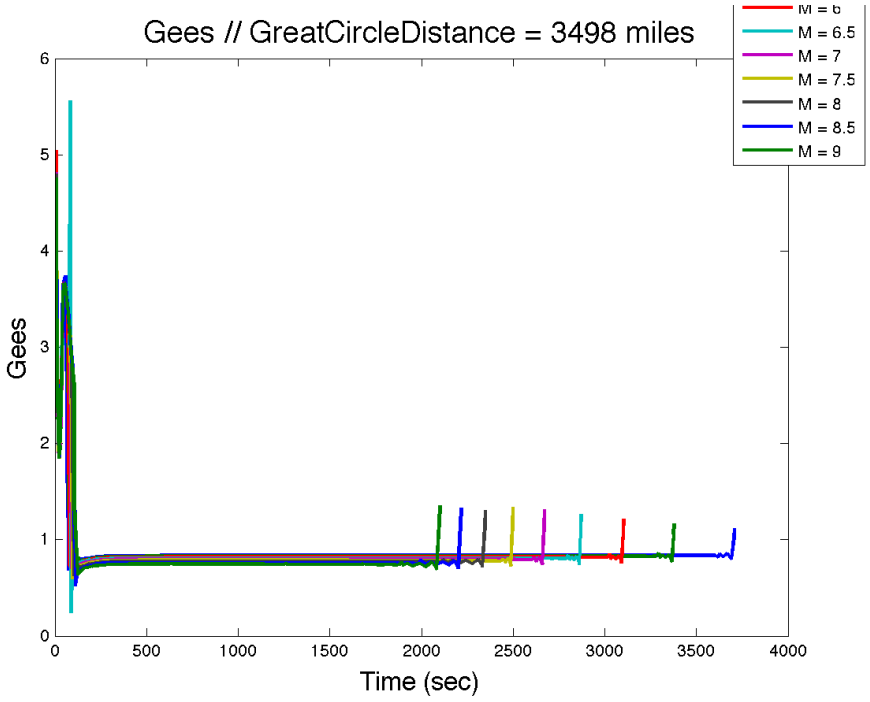

Gees // GreatCircleDistance = 999 miles
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9
Gees
Time (sec)
0
1
2
3
4
5
6
7
8
9
10
0
200
400
600
800
1000
1200

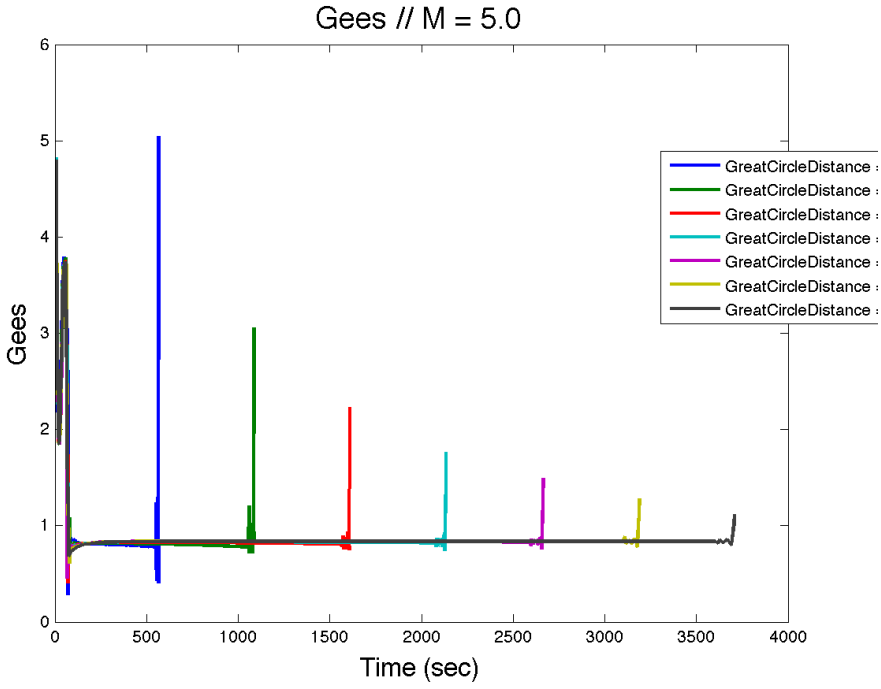

Gees // GreatCircleDistance = 1498 miles
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9
Gees
Time (sec)
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
0
200
400
600
800
1000
1200
1400
1600
1800

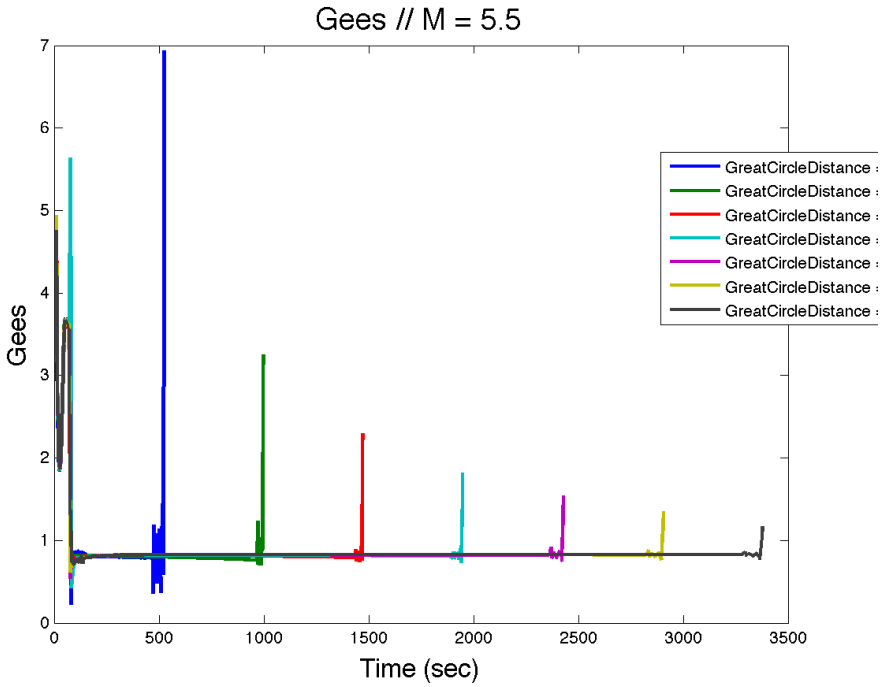

Gees // GreatCircleDistance = 1998 miles
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9
Gees
Time (sec)
0
1
2
3
4
5
6
0
500
1000
1500
2000
2500

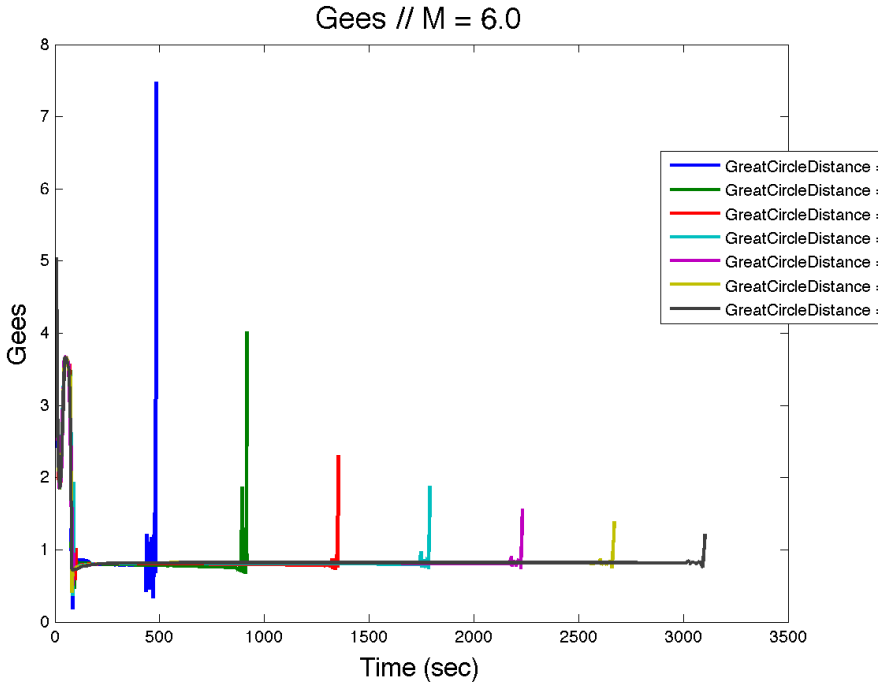

Gees // GreatCircleDistance = 2503 miles
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9
Gees
Time (sec)
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
0
500
1000
1500
2000
2500
3000

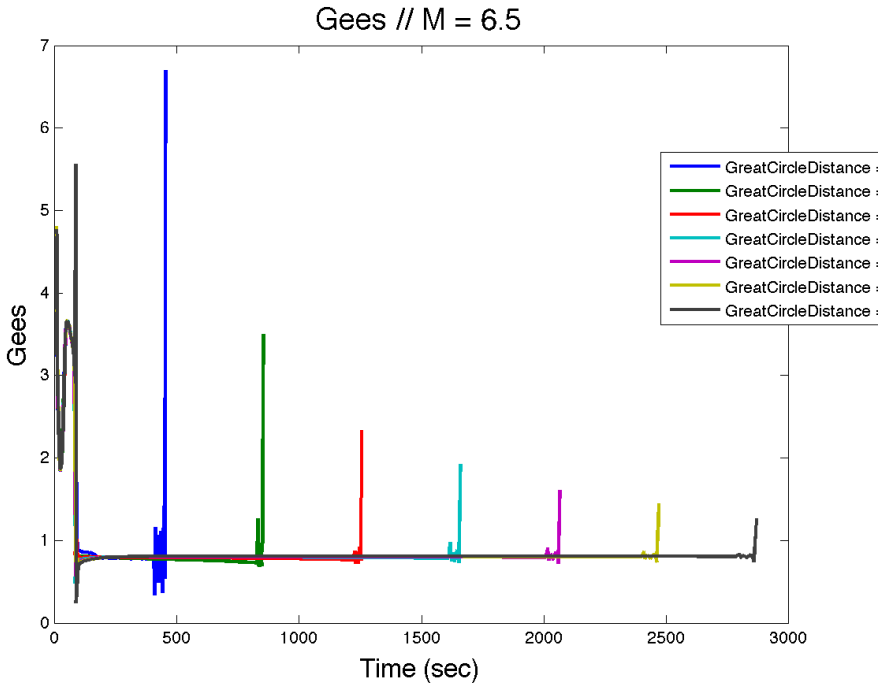

Gees // GreatCircleDistance = 3003 miles
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9
Gees
Time (sec)
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
0
500
1000
1500
2000
2500
3000
3500

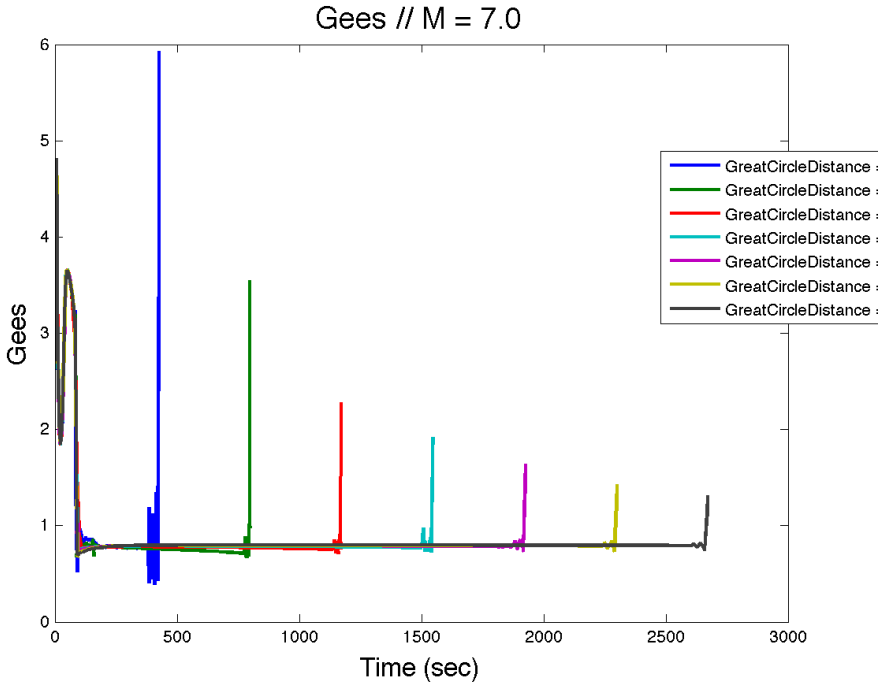

Gees // GreatCircleDistance = 3498 miles
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9
Gees
Time (sec)
0
1
2
3
4
5
6
0
500
1000
1500
2000
2500
3000
3500
4000

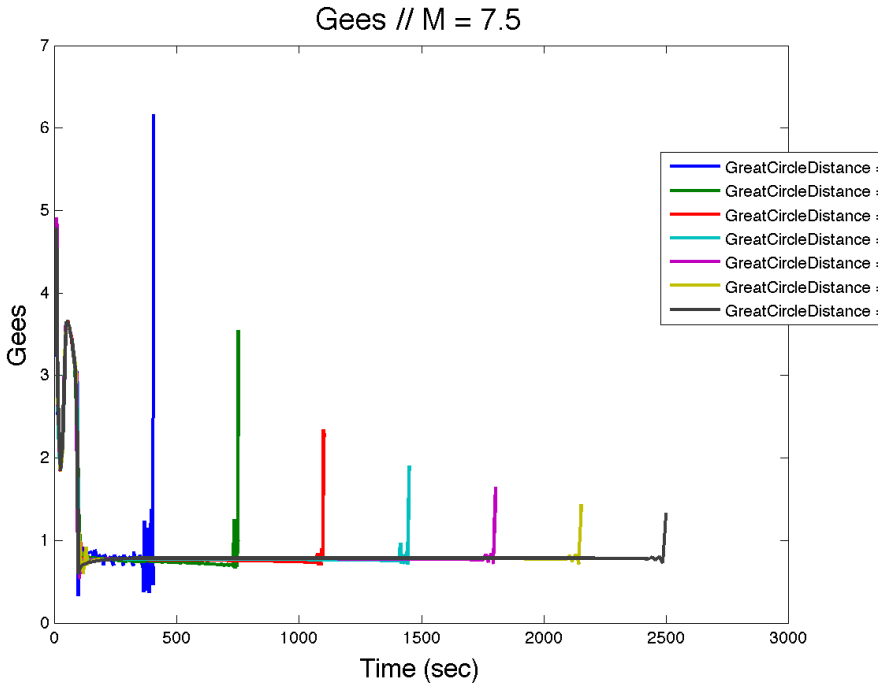

Gees // M = 5.0
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
Gees
Time (sec)
0
1
2
3
4
5
6
0
500
1000
1500
2000
2500
3000
3500
4000

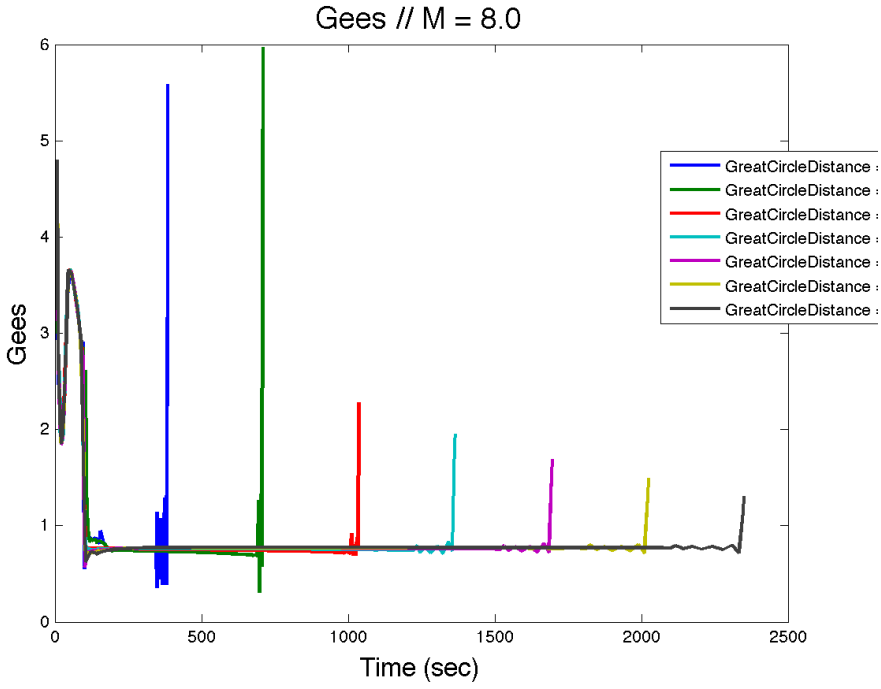

Gees // M = 5.5
Gees
Time (sec)
0
1
2
3
4
5
6
7
0
500
1000
1500
2000
2500
3000
3500
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance

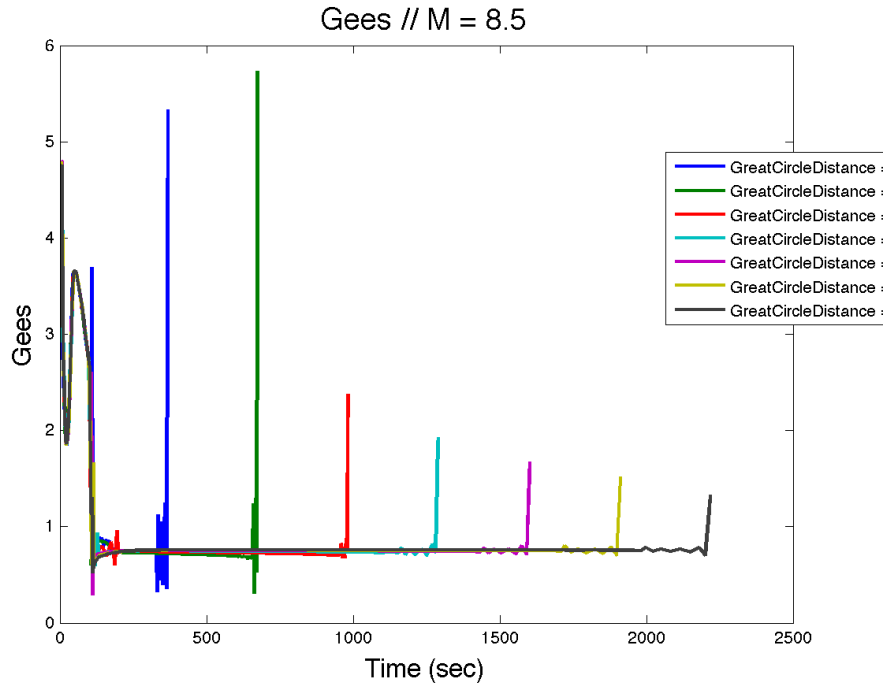

Gees // M = 6.0
Gees
Time (sec)
0
1
2
3
4
5
6
7
8
0
500
1000
1500
2000
2500
3000
3500
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance

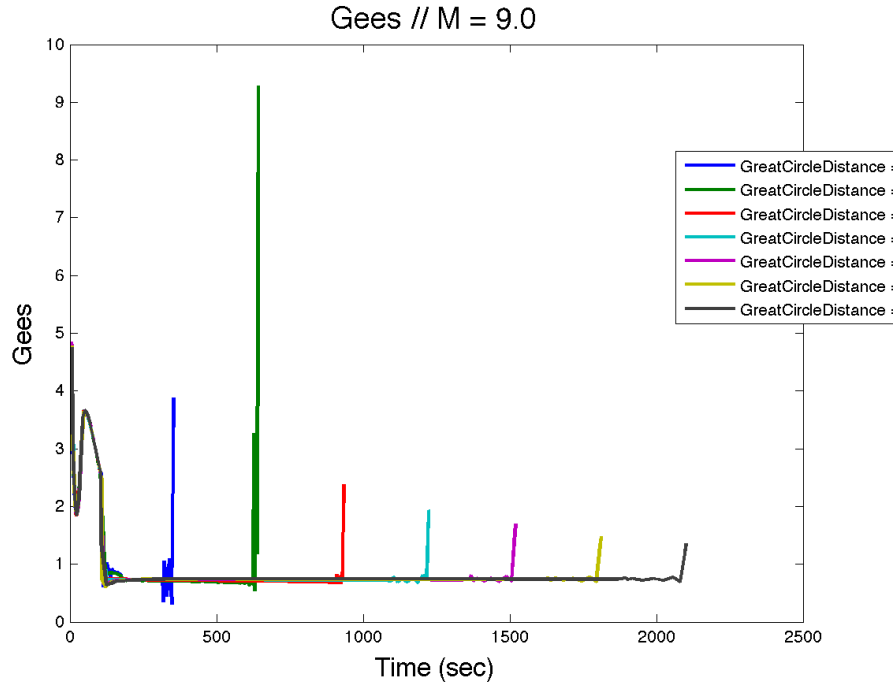

Gees // M = 6.5
Gees
Time (sec)
0
1
2
3
4
5
6
7
0
500
1000
1500
2000
2500
3000
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance


Gees // M = 7.0
Gees
Time (sec)
0
1
2
3
4
5
6
0
500
1000
1500
2000
2500
3000
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance


Gees // M = 7.5
Gees
Time (sec)
0
1
2
3
4
5
6
7
0
500
1000
1500
2000
2500
3000
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance

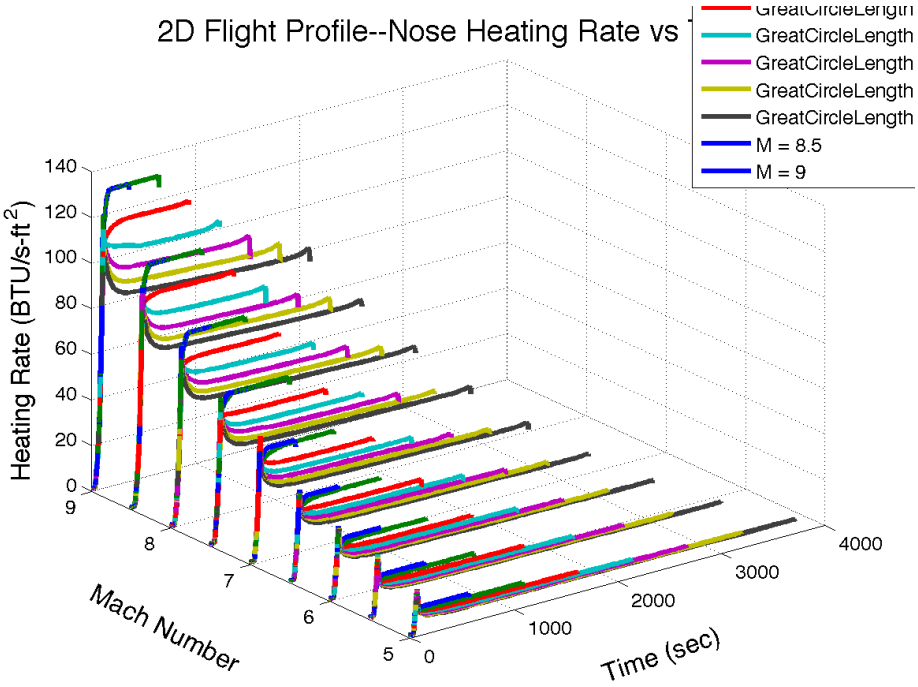

Gees // M = 8.0
Gees
Time (sec)
0
1
2
3
4
5
6
0
500
1000
1500
2000
2500
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance

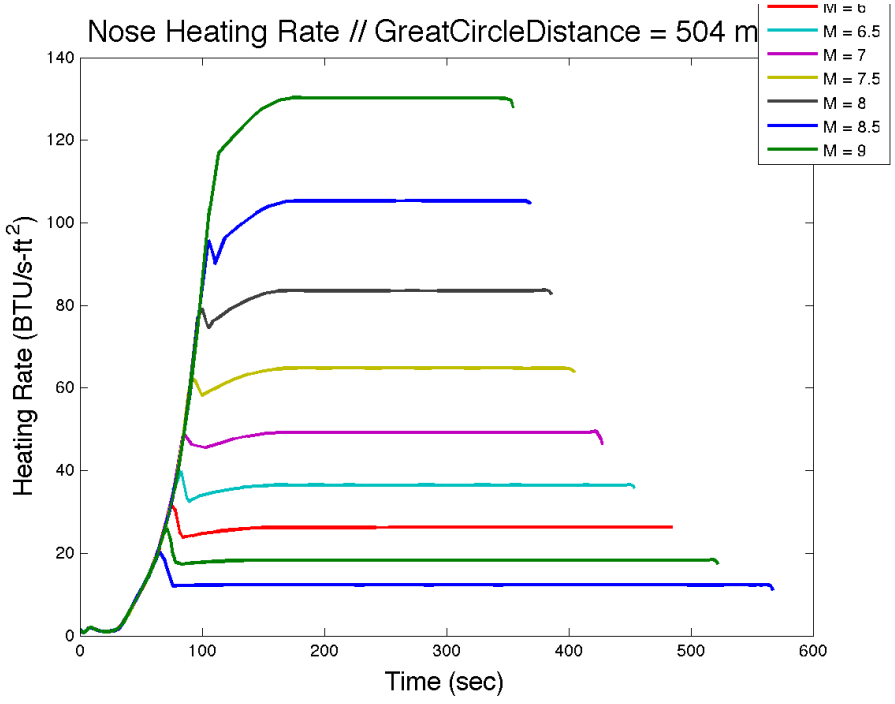

Gees // M = 8.5
Gees
Time (sec)
0
1
2
3
4
5
6
0
500
1000
1500
2000
2500
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance

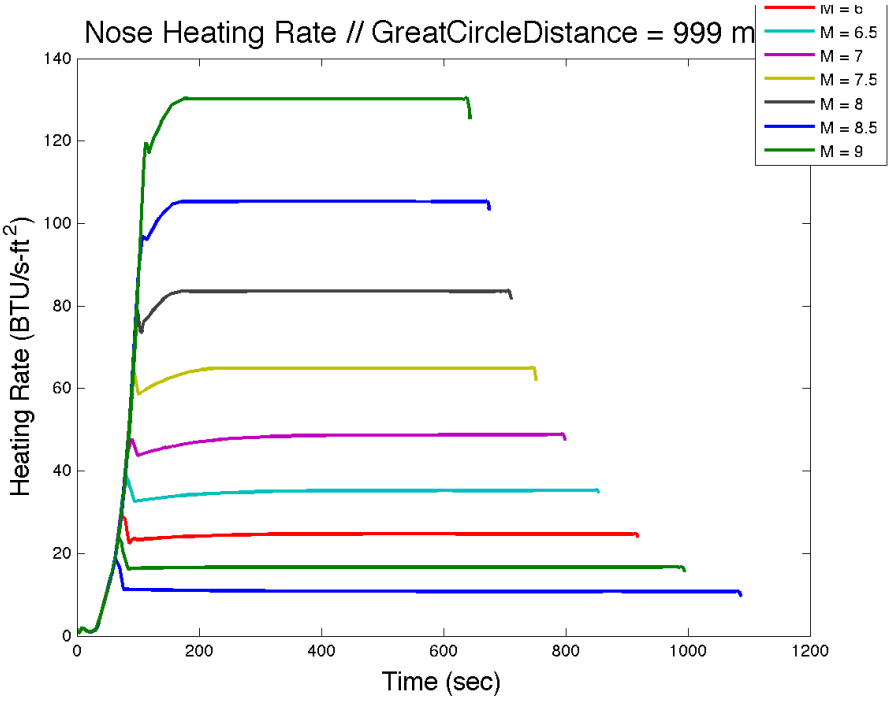

Gees // M = 9.0
Gees
Time (sec)
0
1
2
3
4
5
6
7
8
9
10
0
500
1000
1500
2000
2500
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance

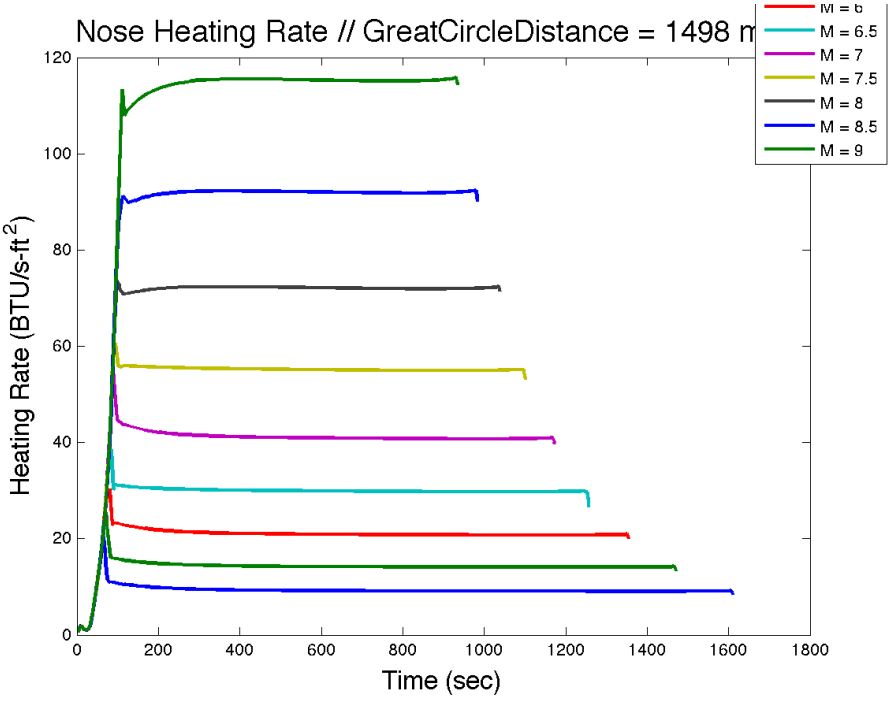
D.6 Thermal Parameters vs. Time

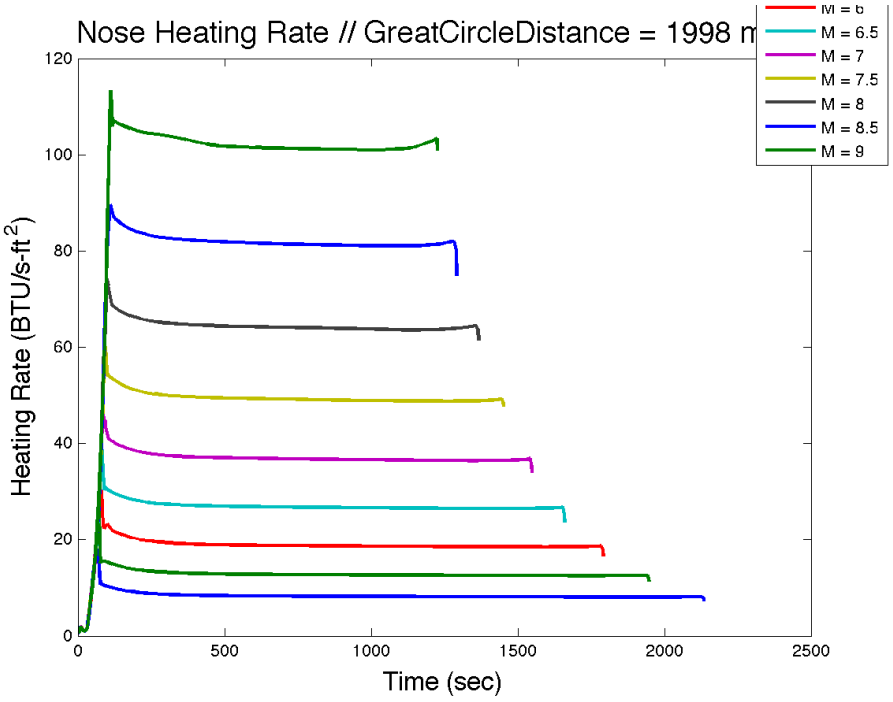
D.6.1 Nose Heating Rate vs. Time.

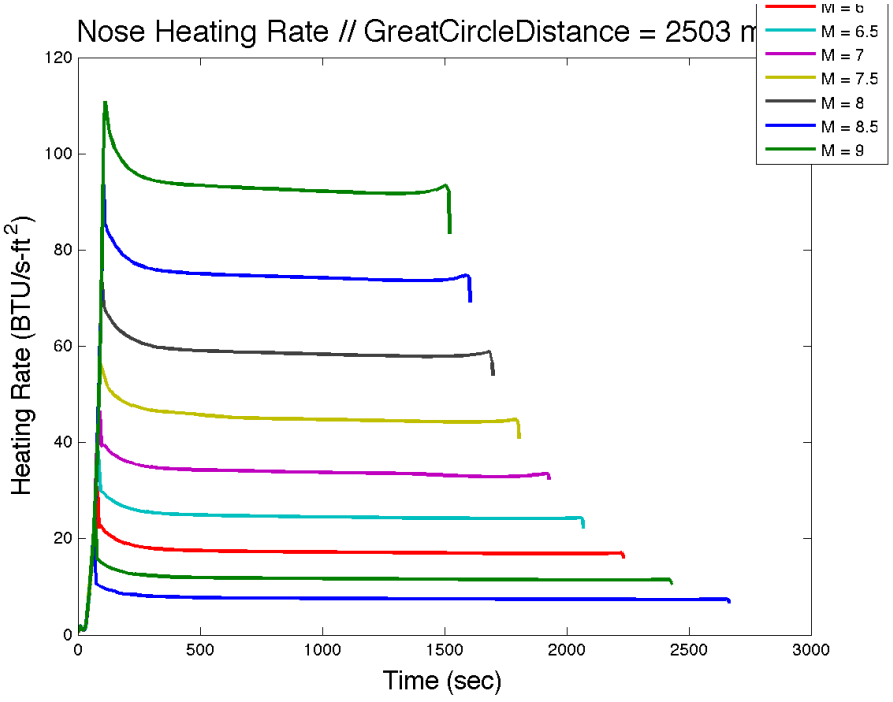


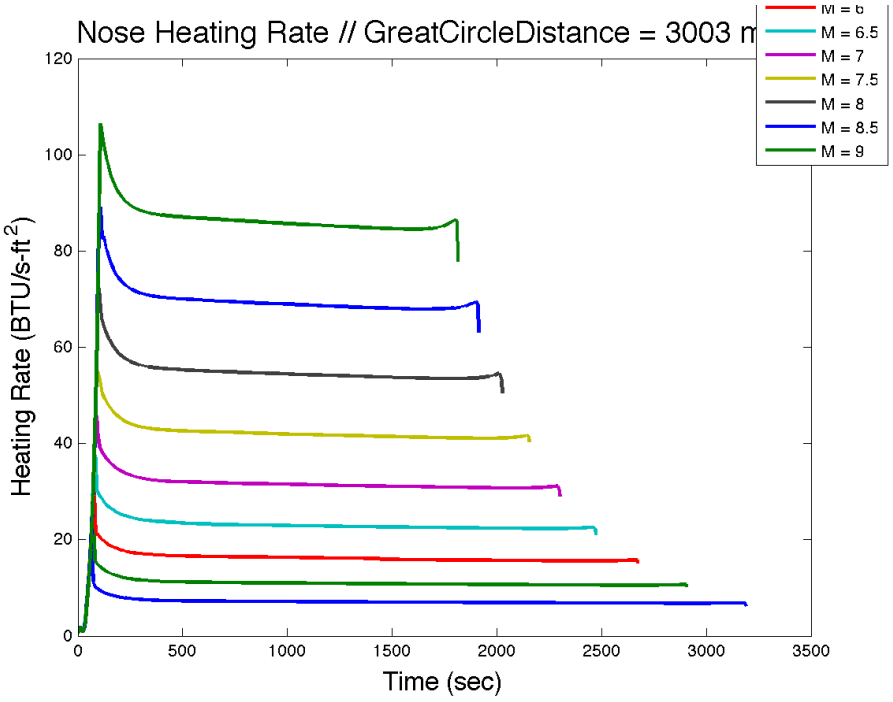


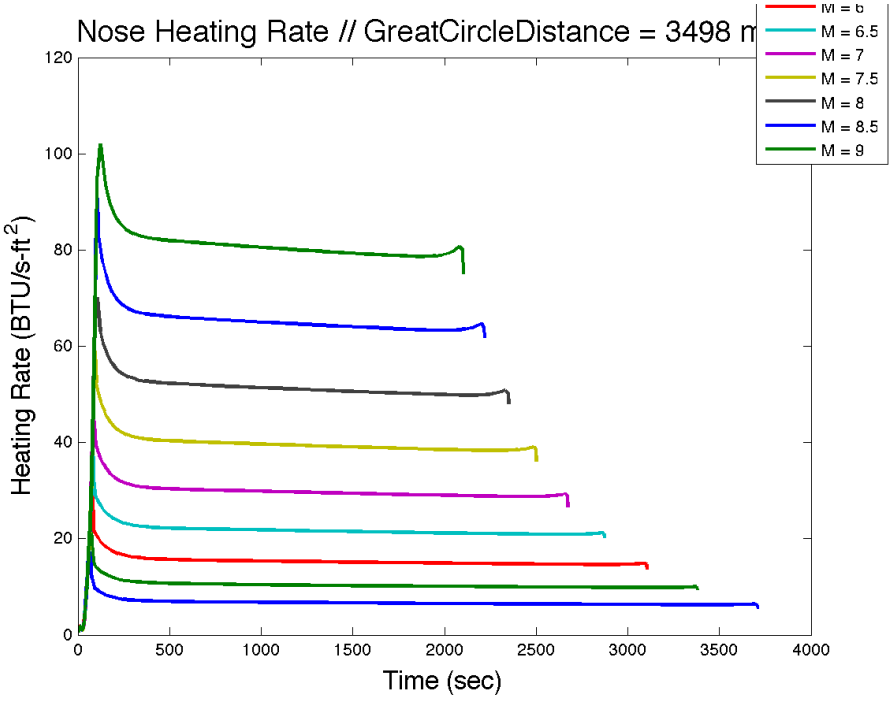

Nose Heating Rate // GreatCircleDistance = 999 m
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9
Heating Rate (BTU/s-ft^2)
Time (sec)
0
20
40
60
80
100
120
140
0
200
400
600
800
1000
1200

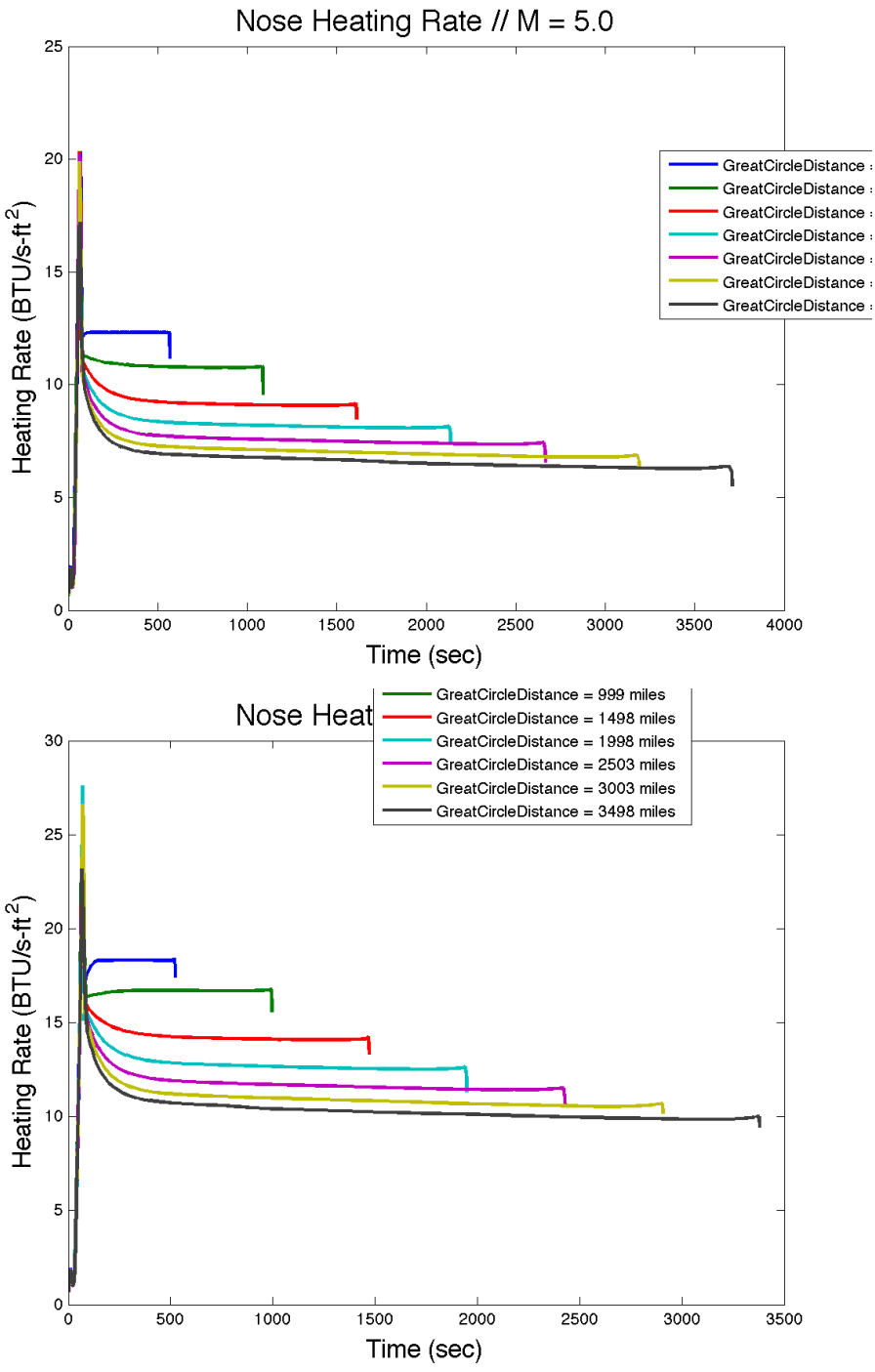

Nose Heating Rate // GreatCircleDistance = 1498 m
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9
Heating Rate (BTU/s-ft^2)
Time (sec)
0
20
40
60
80
100
120
0
200
400
600
800
1000
1200
1400
1600
1800

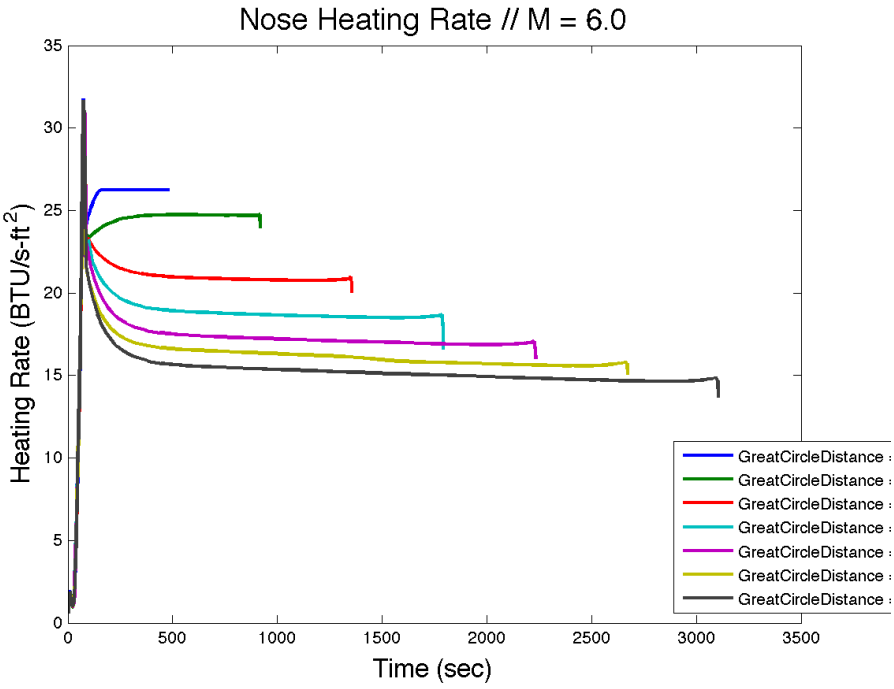

Nose Heating Rate // GreatCircleDistance = 1998 m
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9
Heating Rate (BTU/s-ft^2)
Time (sec)
0
20
40
60
80
100
120
0
500
1000
1500
2000
2500

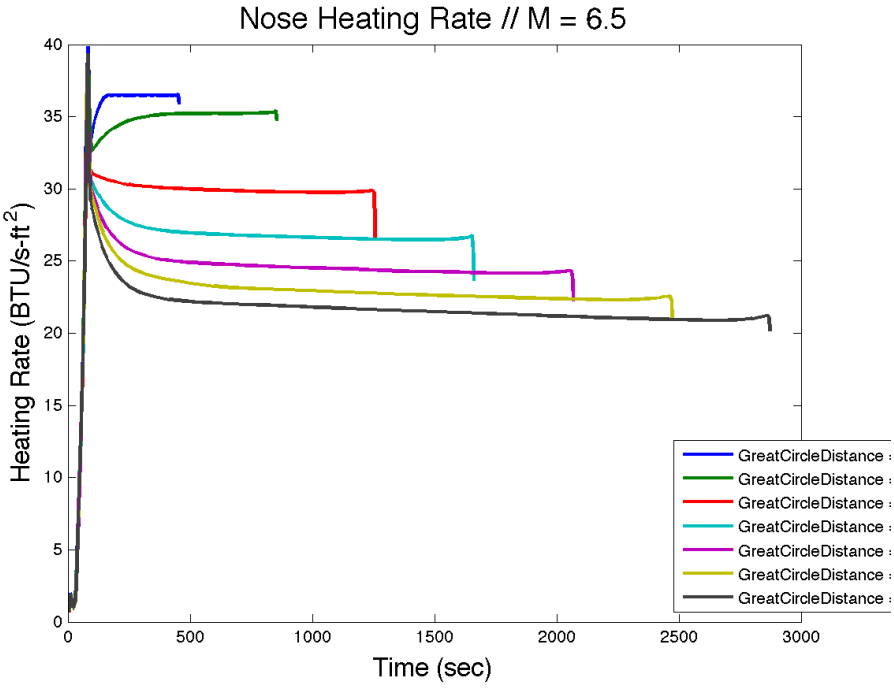

Nose Heating Rate // GreatCircleDistance = 2503 m
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9
Heating Rate (BTU/s-ft^2)
Time (sec)
0
20
40
60
80
100
120
0
500
1000
1500
2000
2500
3000

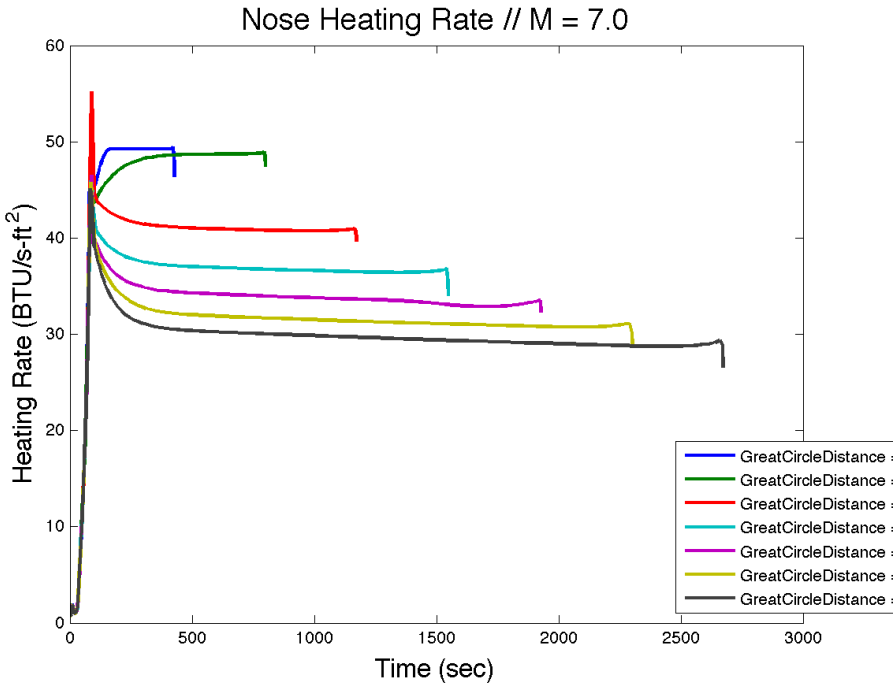

Nose Heating Rate // GreatCircleDistance = 3003 m
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9
Heating Rate (BTU/s-ft^2)
Time (sec)
0
20
40
60
80
100
120
0
500
1000
1500
2000
2500
3000
3500

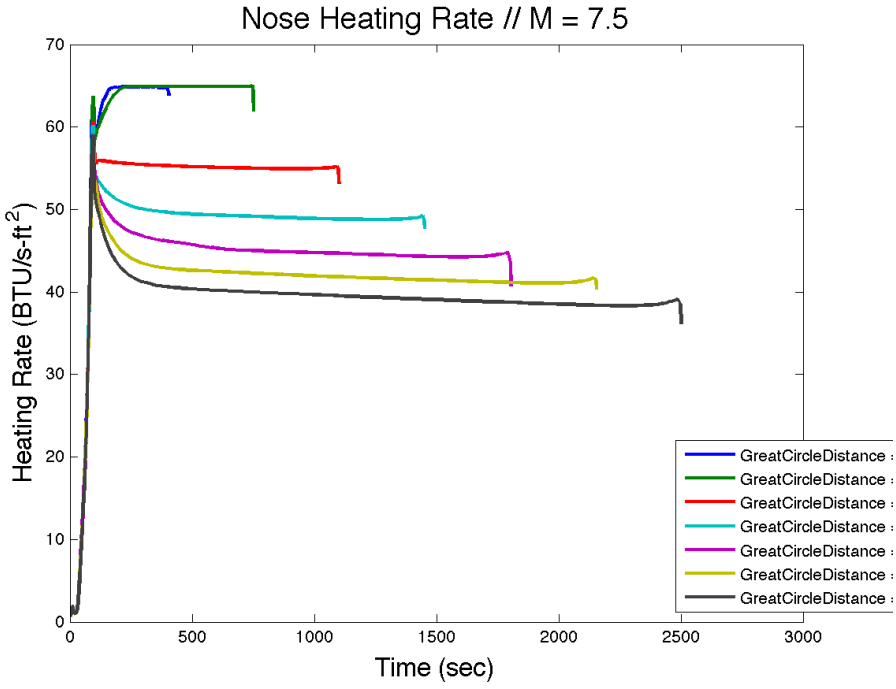

Nose Heating Rate // GreatCircleDistance = 3498 m
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9
Heating Rate (BTU/s-ft^2)
Time (sec)
0
20
40
60
80
100
120
0
500
1000
1500
2000
2500
3000
3500
4000

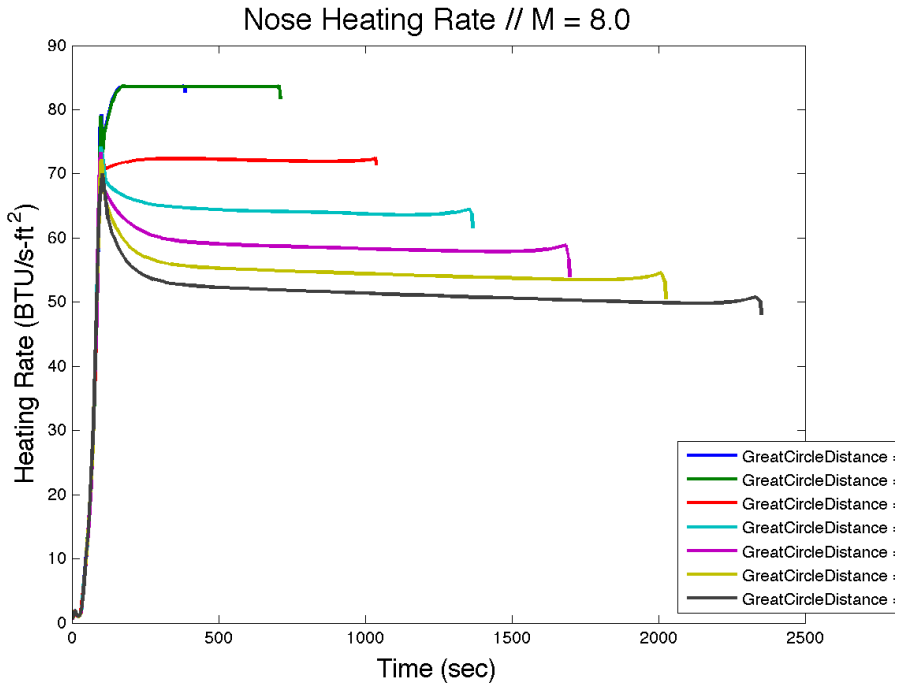

Nose Heating Rate // M = 5.0
Heating Rate (BTU/s-ft^2)
Time (sec)
GreatCircleDistance = 999 miles
GreatCircleDistance = 1498 miles
GreatCircleDistance = 1998 miles
GreatCircleDistance = 2503 miles
GreatCircleDistance = 3003 miles
GreatCircleDistance = 3498 miles

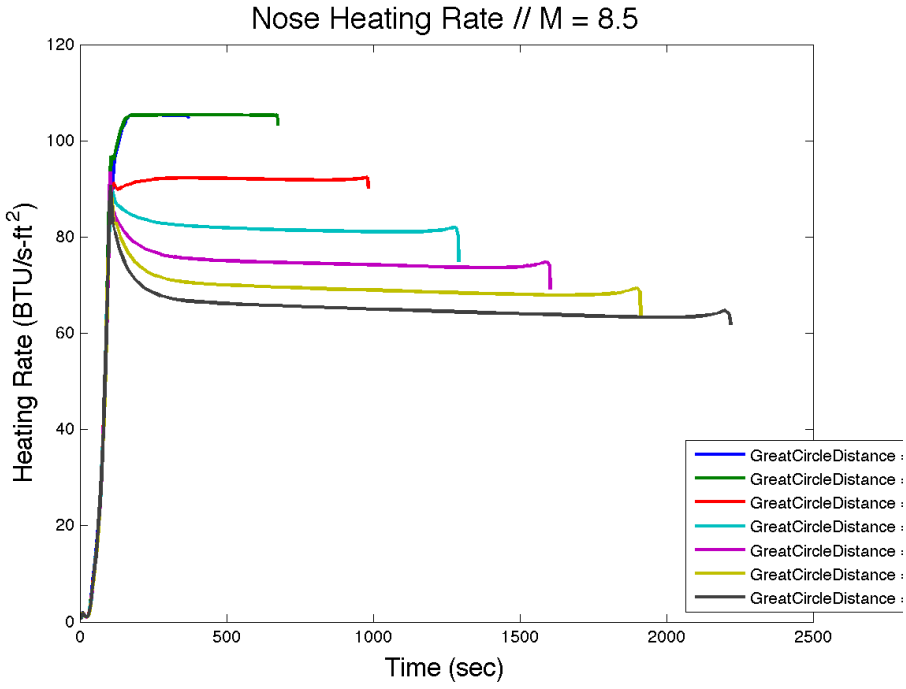

Nose Heating Rate // M = 6.0
Heating Rate (BTU/s-ft^2)
Time (sec)
0
5
10
15
20
25
30
35
0
500
1000
1500
2000
2500
3000
3500
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance

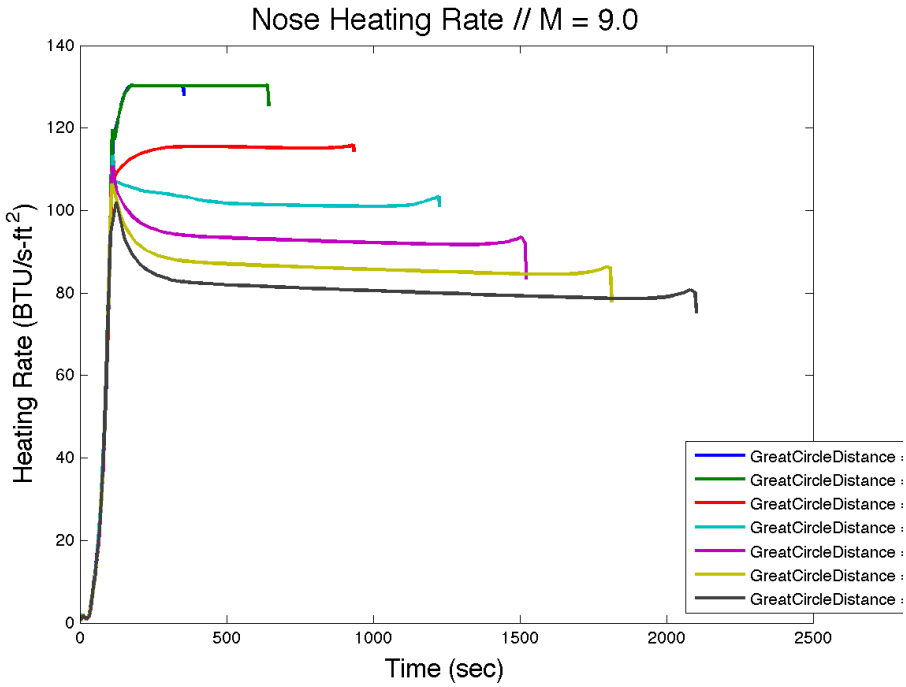

Nose Heating Rate // M = 6.5
Heating Rate (BTU/s-ft^2)
Time (sec)
0
5
10
15
20
25
30
35
40
0
500
1000
1500
2000
2500
3000
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance


Nose Heating Rate // M = 7.0
Heating Rate (BTU/s-ft^2)
Time (sec)
0
10
20
30
40
50
60
0
500
1000
1500
2000
2500
3000
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance

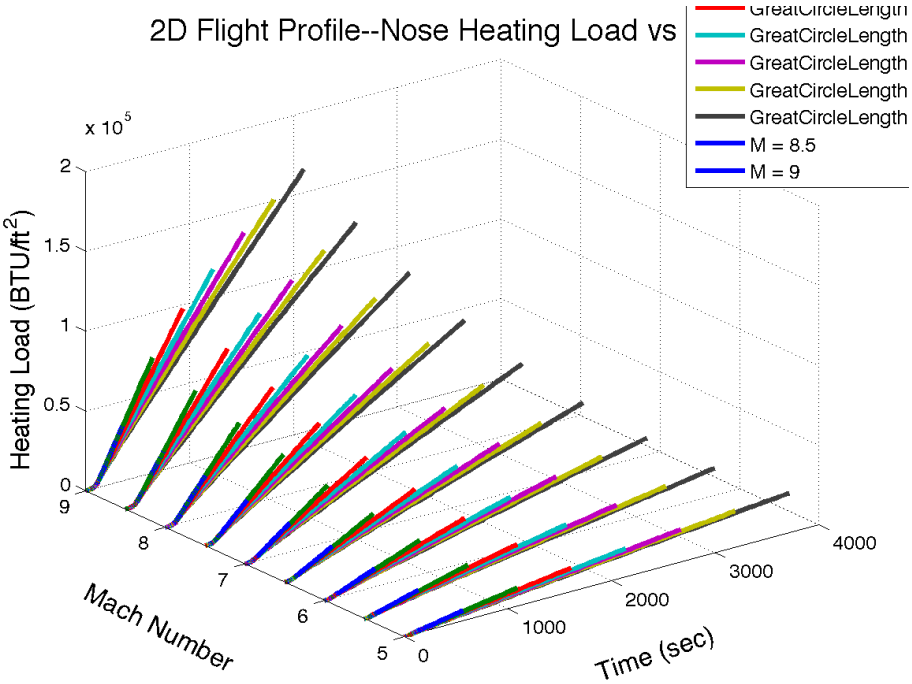

Nose Heating Rate // M = 7.5
Heating Rate (BTU/s-ft^2)
Time (sec)
0
10
20
30
40
50
60
70
0
500
1000
1500
2000
2500
3000
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance

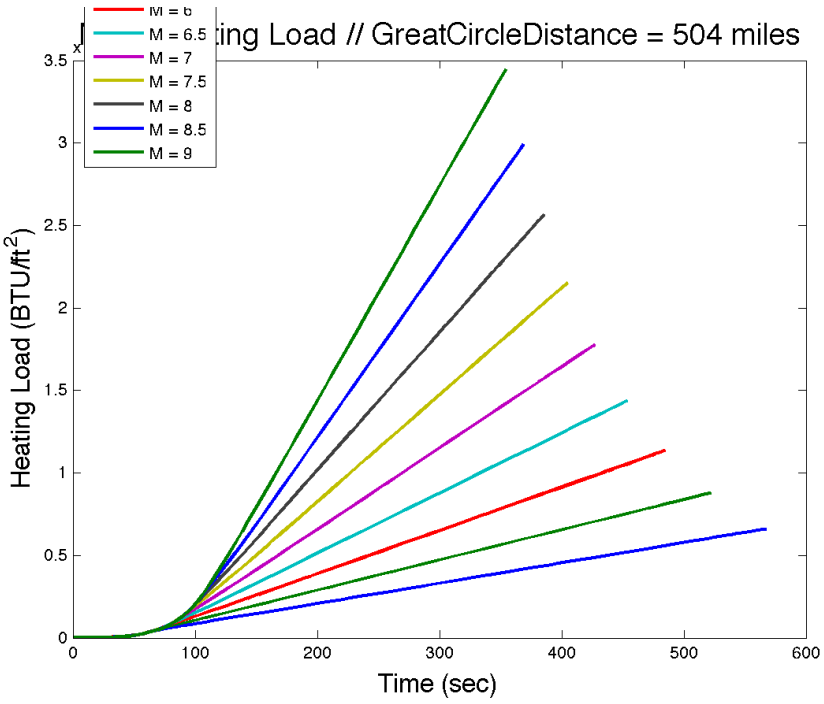

Nose Heating Rate // M = 8.0
Heating Rate (BTU/s-ft^2)
Time (sec)
0
10
20
30
40
50
60
70
80
90
0
500
1000
1500
2000
2500
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance

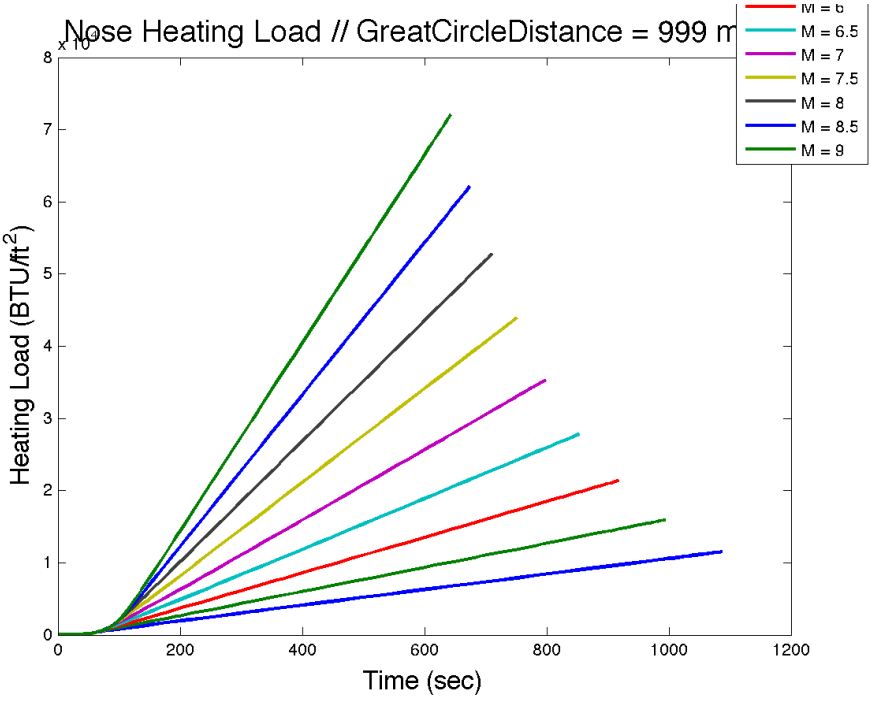

Nose Heating Rate // M = 8.5
Heating Rate (BTU/s-ft^2)
Time (sec)
0
20
40
60
80
100
120
0
500
1000
1500
2000
2500
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance

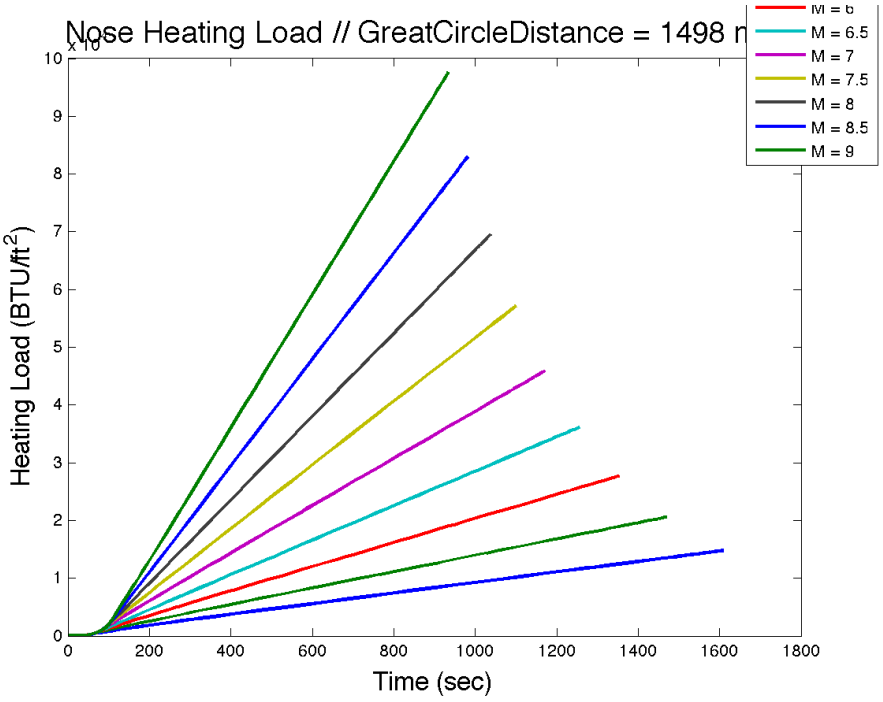


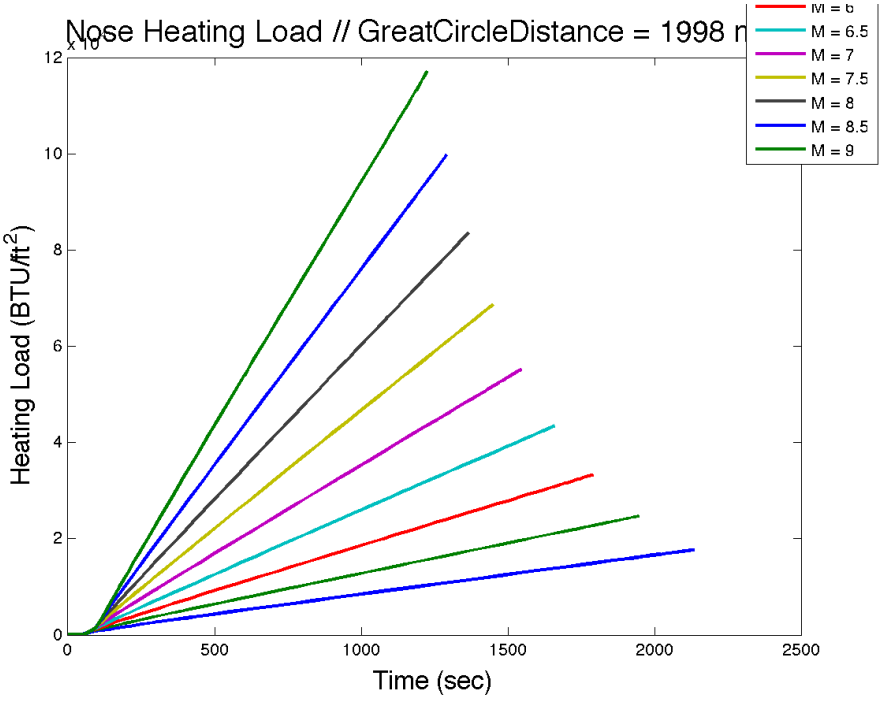
D.6.2 Nose Heating Load vs. Time.

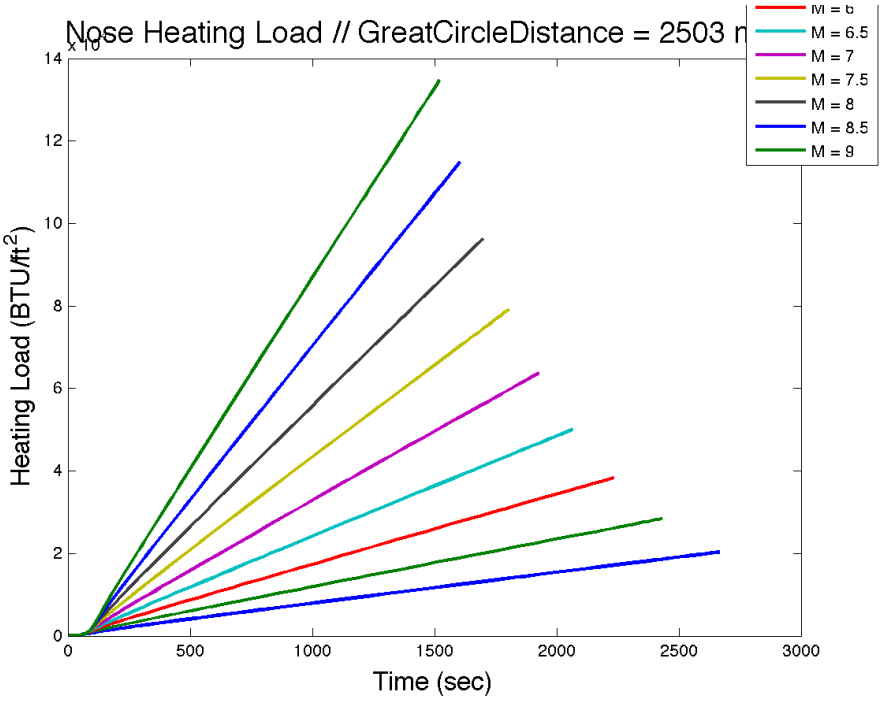


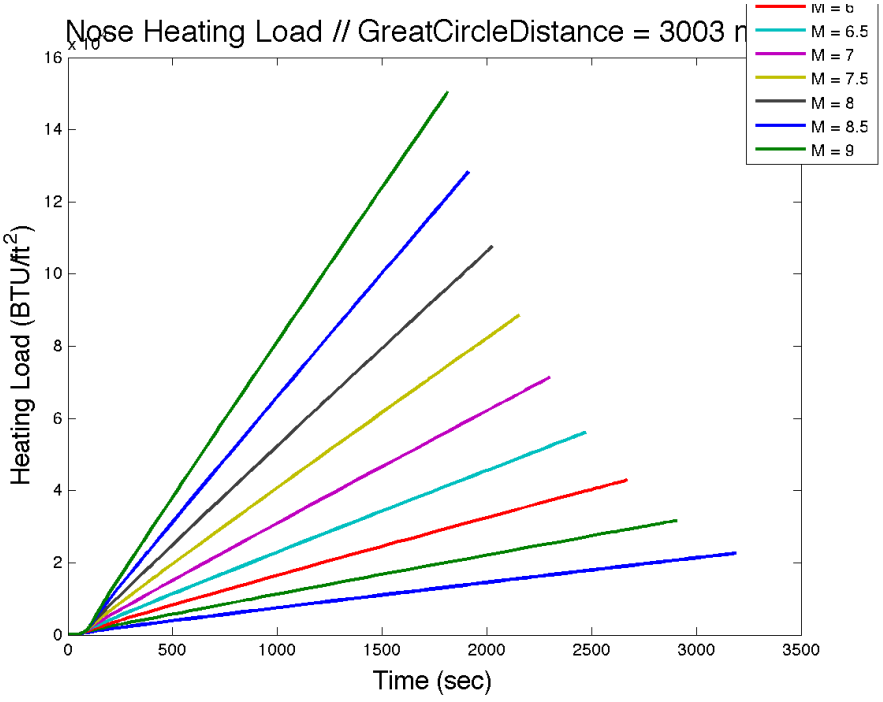

M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9
x 10^2
3.5
3
2.5
2
1.5
1
0.5
0
Heating Load (BTU/ft^2)
0
100
200
300
400
500
600
Time (sec)

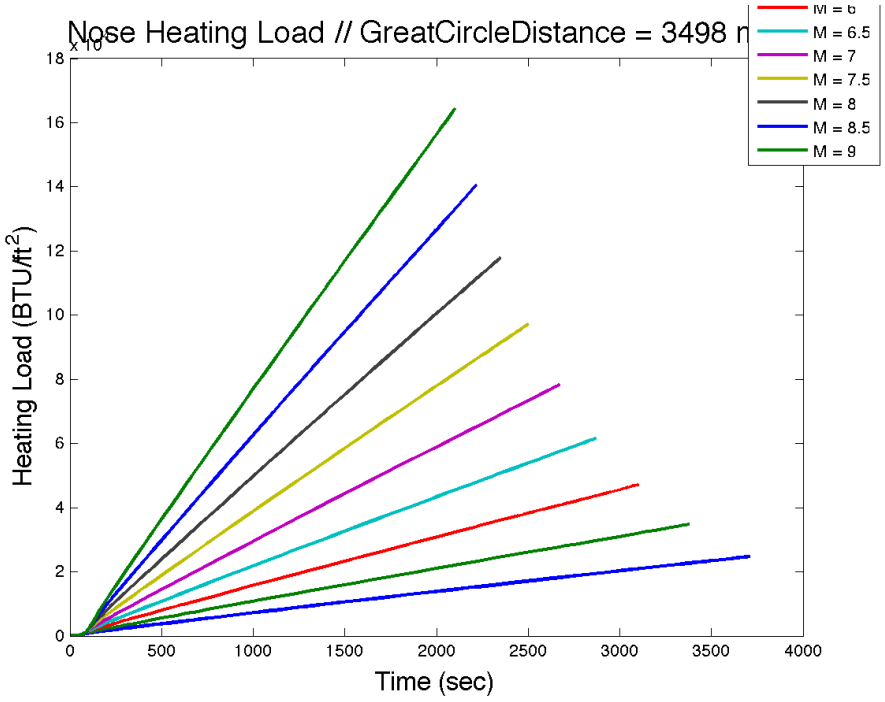

Nose Heating Load // GreatCircleDistance = 999 m
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9
x 10^2
8
7
6
5
4
3
2
1
0
Heating Load (BTU/ft^2)
0
200
400
600
800
1000
1200
Time (sec)

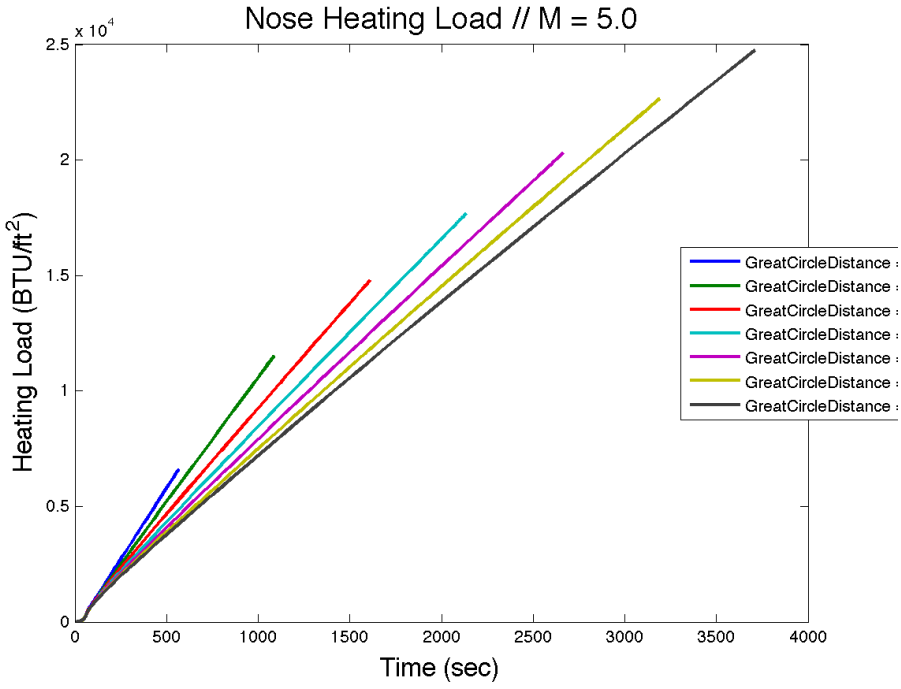

Nose Heating Load // GreatCircleDistance = 1498 n
x 10
Heating Load (BTU/ft2)
Time (sec)
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9

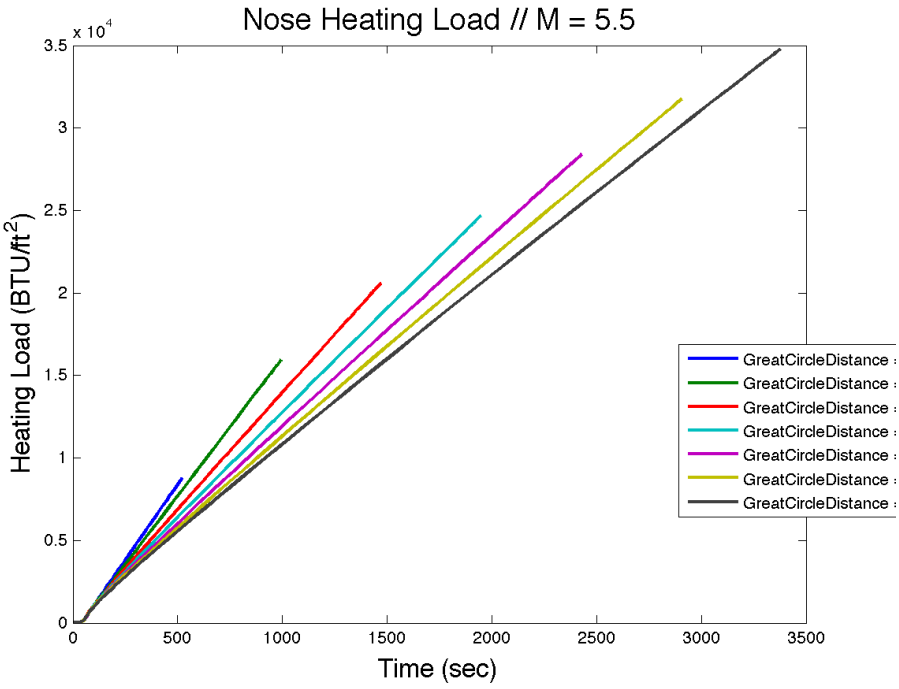

Nose Heating Load // GreatCircleDistance = 1998 n
x 10
Heating Load (BTU/ft2)
Time (sec)
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9

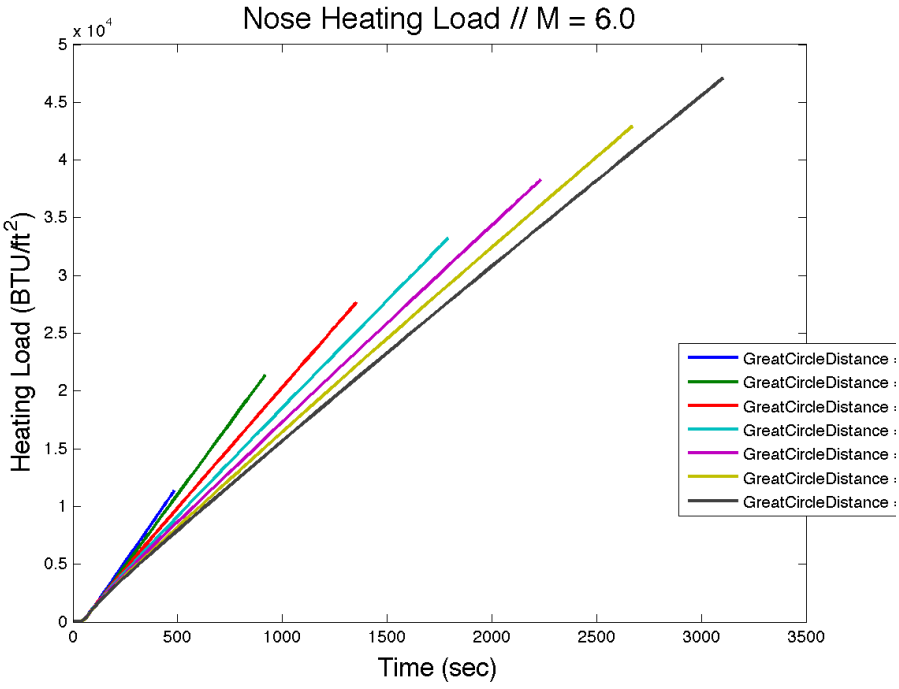

Nose Heating Load // GreatCircleDistance = 2503 n
x 10
14
12
10
8
6
4
2
0
Heating Load (BTU/ft2)
0
500
1000
1500
2000
2500
3000
Time (sec)
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9

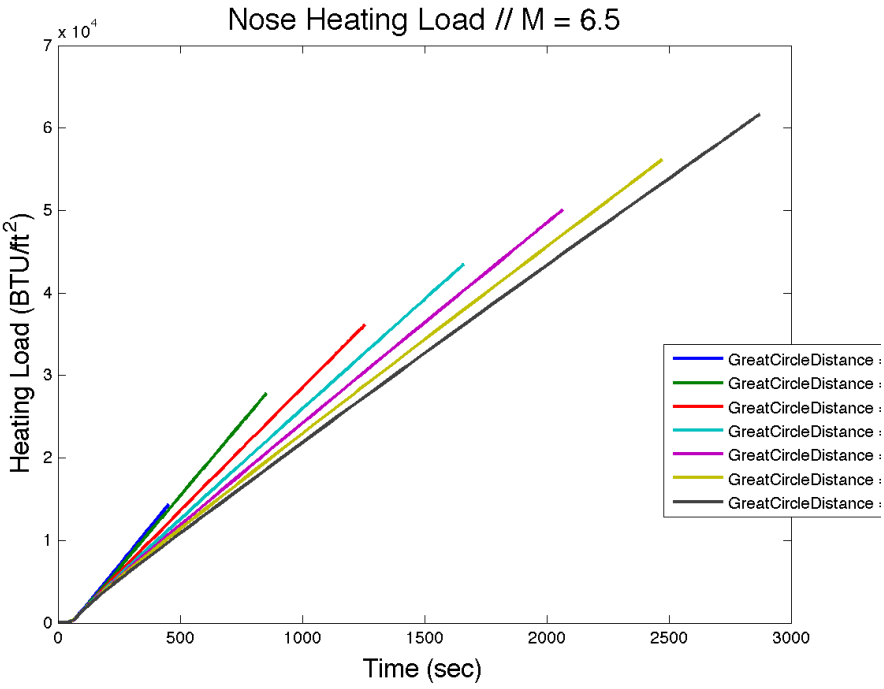

Nose Heating Load // GreatCircleDistance = 3003 n
x 10
16
14
12
10
8
6
4
2
0
Heating Load (BTU/ft2)
0
500
1000
1500
2000
2500
3000
3500
Time (sec)
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9

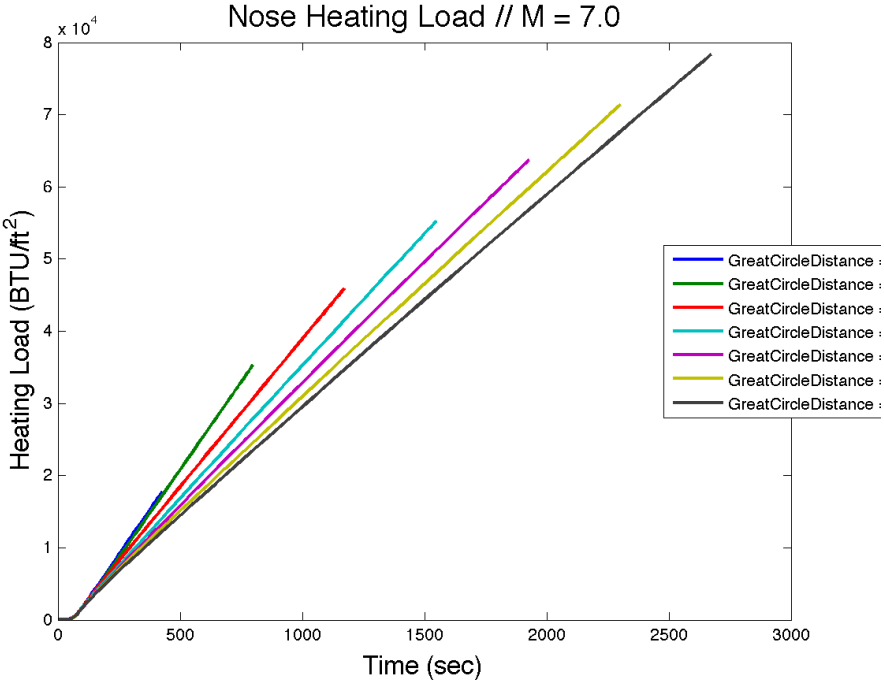

Nose Heating Load // GreatCircleDistance = 3498 n
x 10^4
Heating Load (BTU/ft^2)
Time (sec)
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9

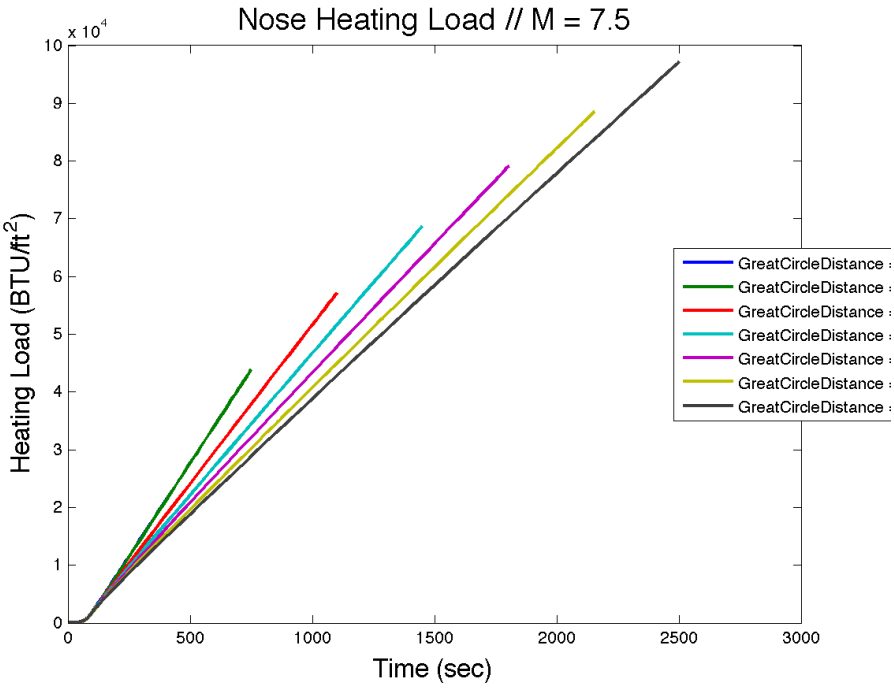

Nose Heating Load // M = 5.0
x 10^4
Heating Load (BTU/ft^2)
Time (sec)
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance

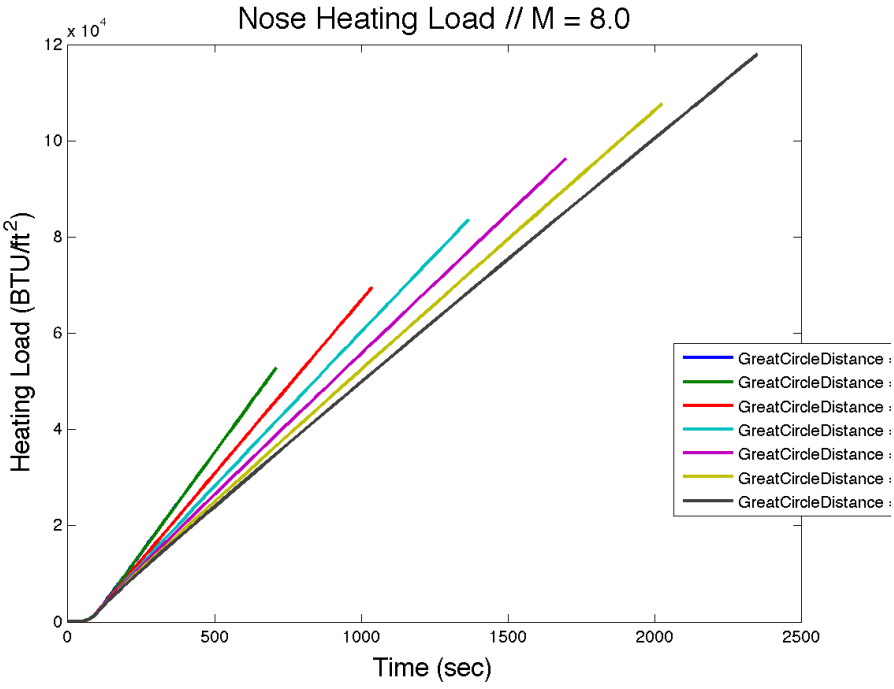

Nose Heating Load // M = 5.5
x 10^4
Heating Load (BTU/ft^2)
Time (sec)
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance

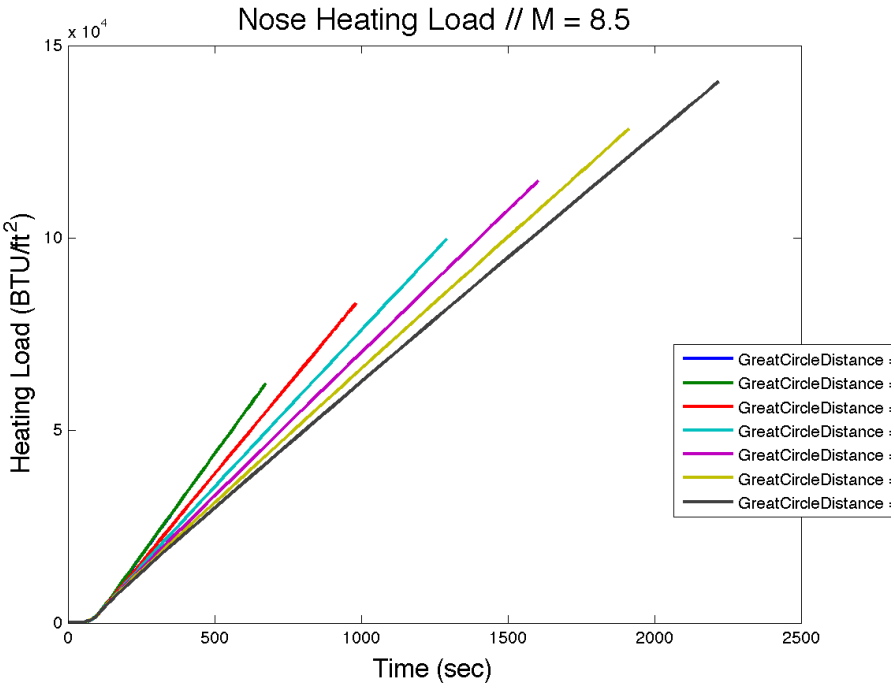

Nose Heating Load // M = 6.0
x 10^4
Heating Load (BTU/ft^2)
Time (sec)
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance

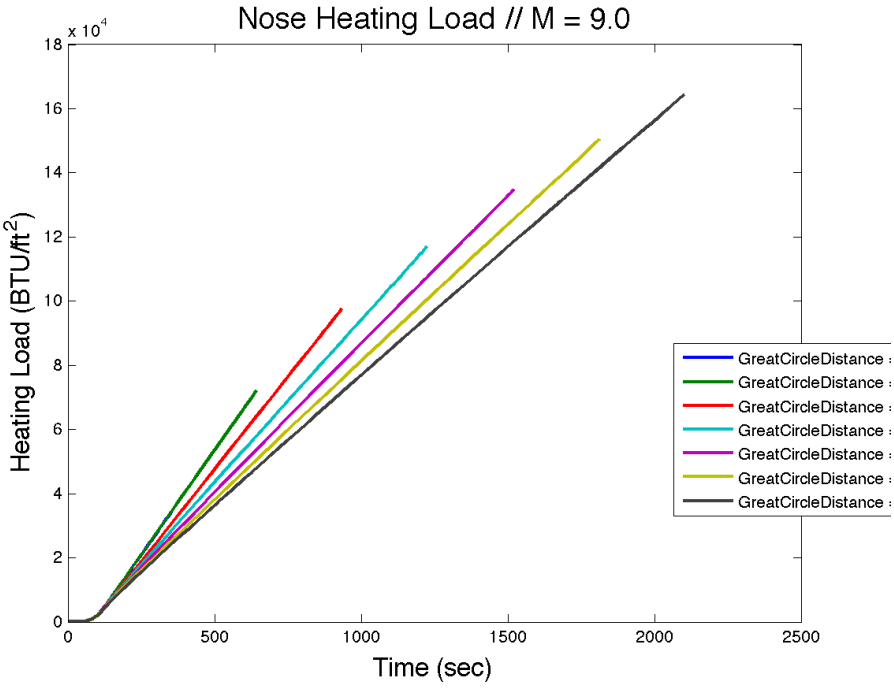

Nose Heating Load // M = 6.5
x 10^4
Heating Load (BTU/ft^2)
Time (sec)
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance


Nose Heating Load // M = 7.0
x 10^4
Heating Load (BTU/ft^2)
Time (sec)
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance

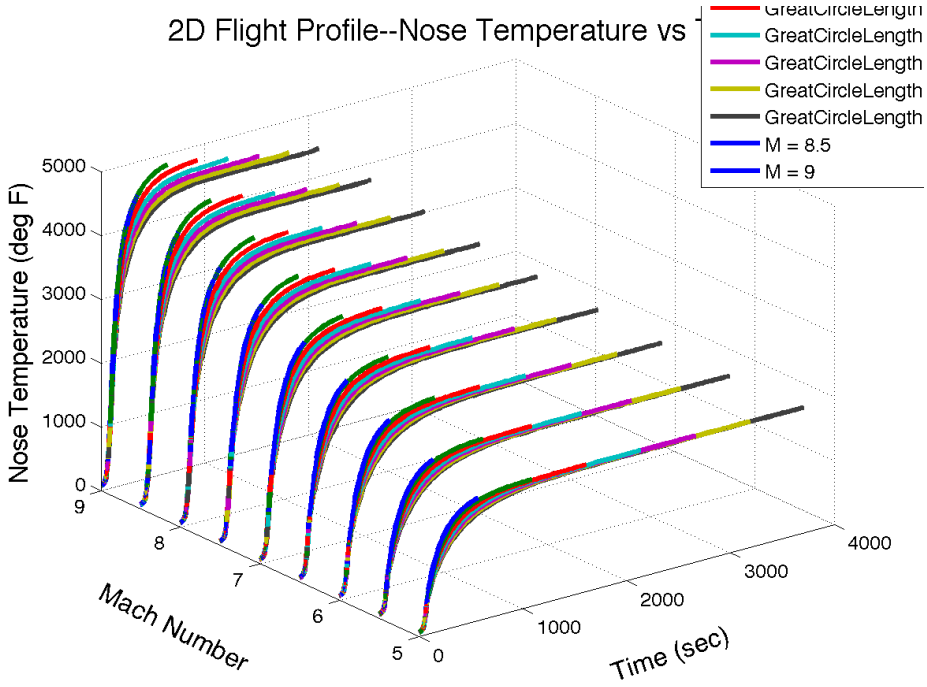

Nose Heating Load // M = 7.5
x 10^4
Heating Load (BTU/ft^2)
Time (sec)
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance

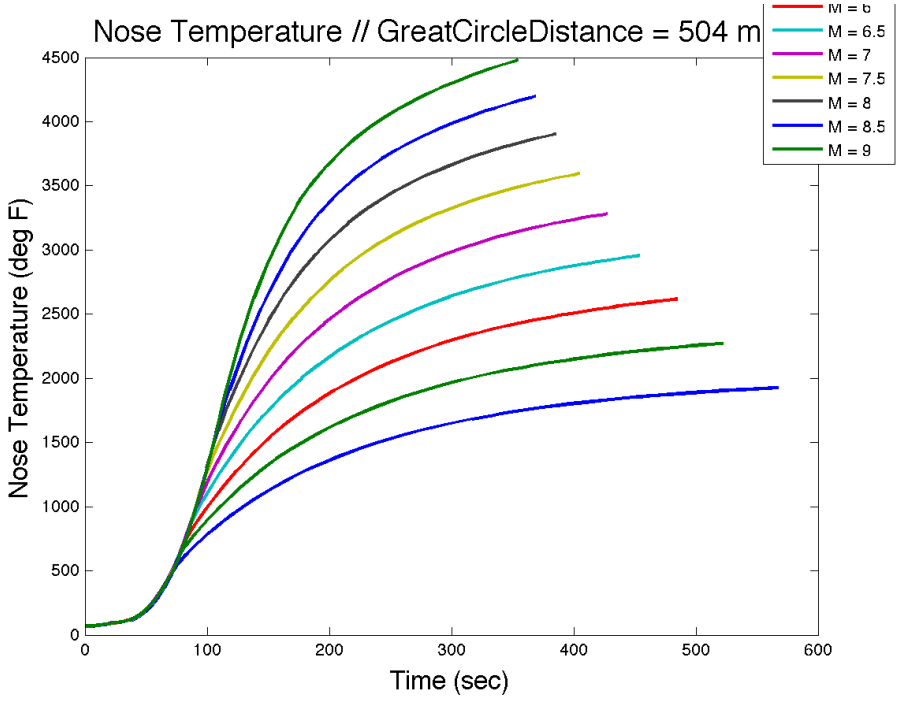

Nose Heating Load // M = 8.0
x 10^4
Heating Load (BTU/ft^2)
Time (sec)
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance

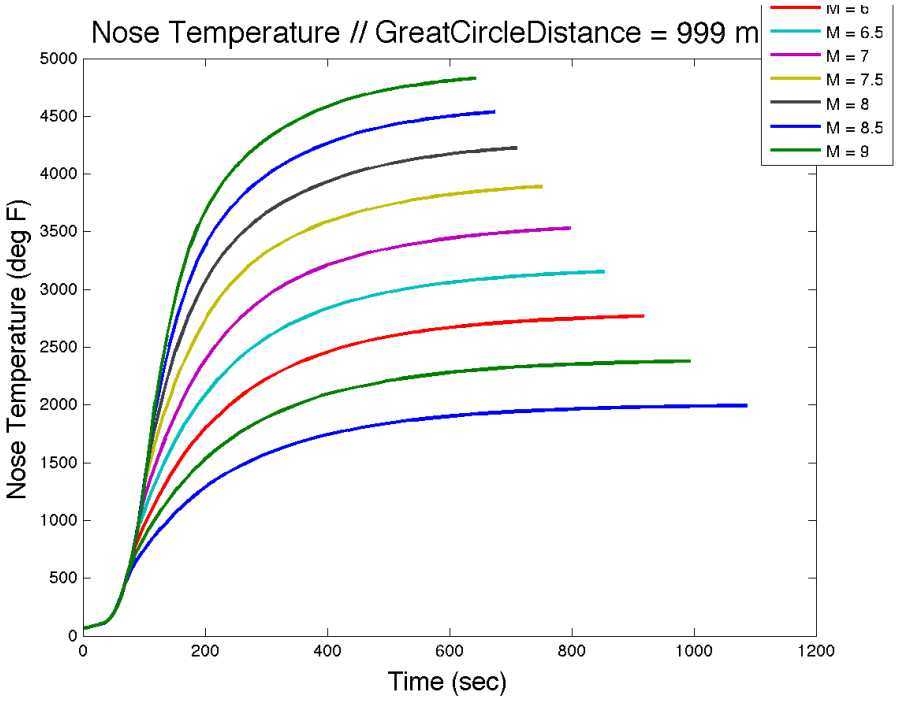

Nose Heating Load // M = 8.5
x 10^4
Heating Load (BTU/ft^2)
Time (sec)
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance

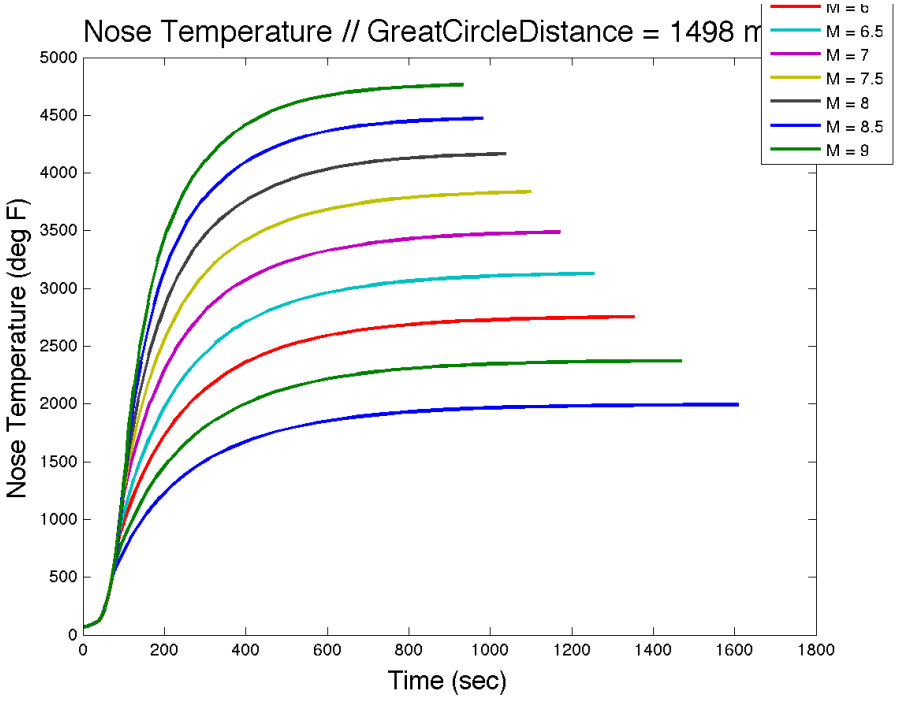

Nose Heating Load // M = 9.0
x 10^4
Heating Load (BTU/ft^2)
Time (sec)
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance

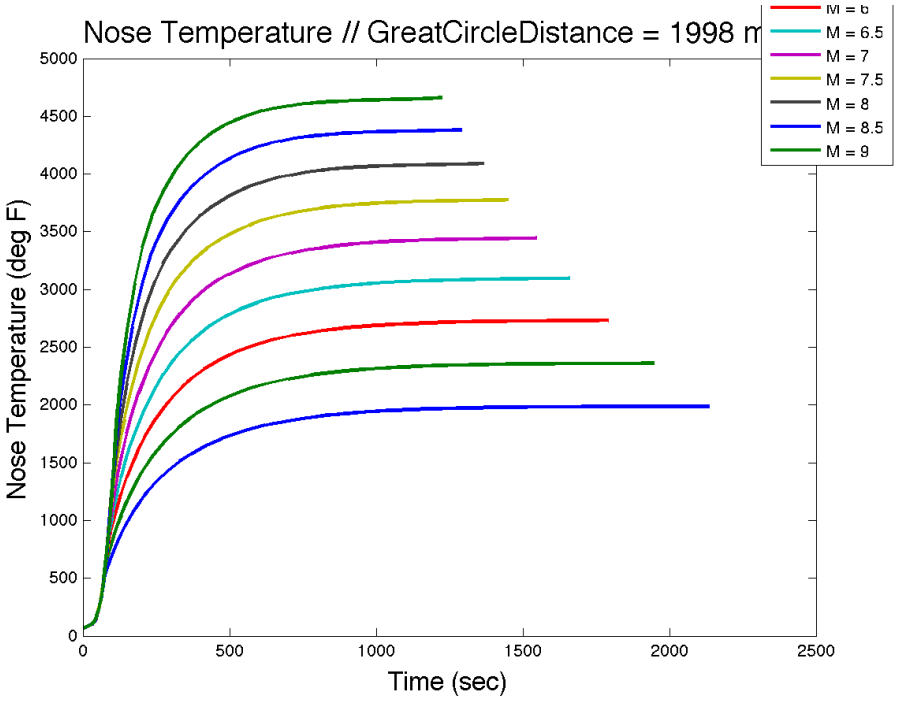
D.6.3 Nose Temperature vs. Time.

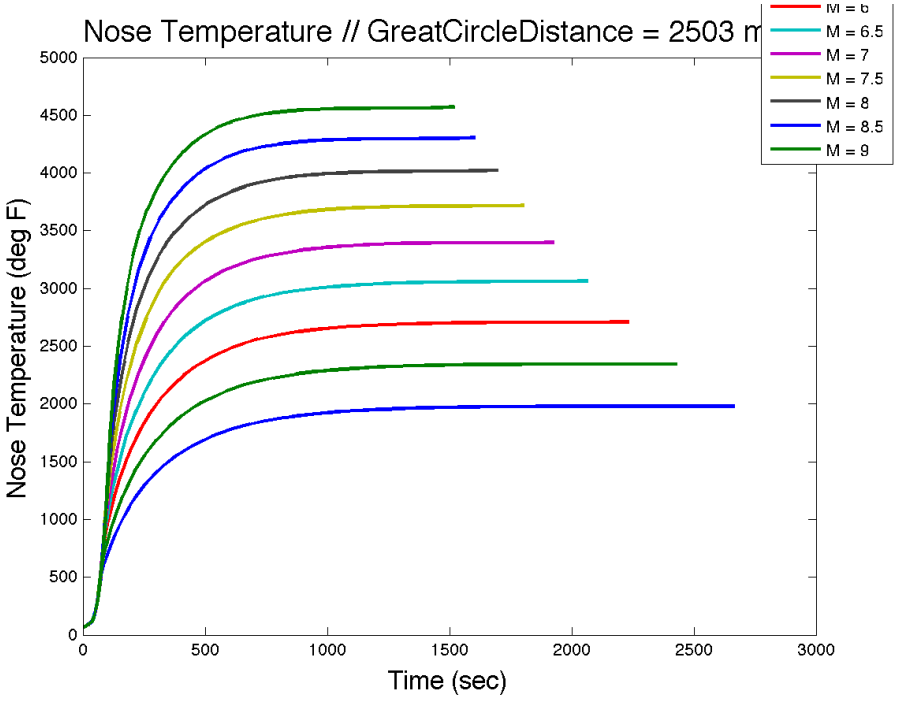


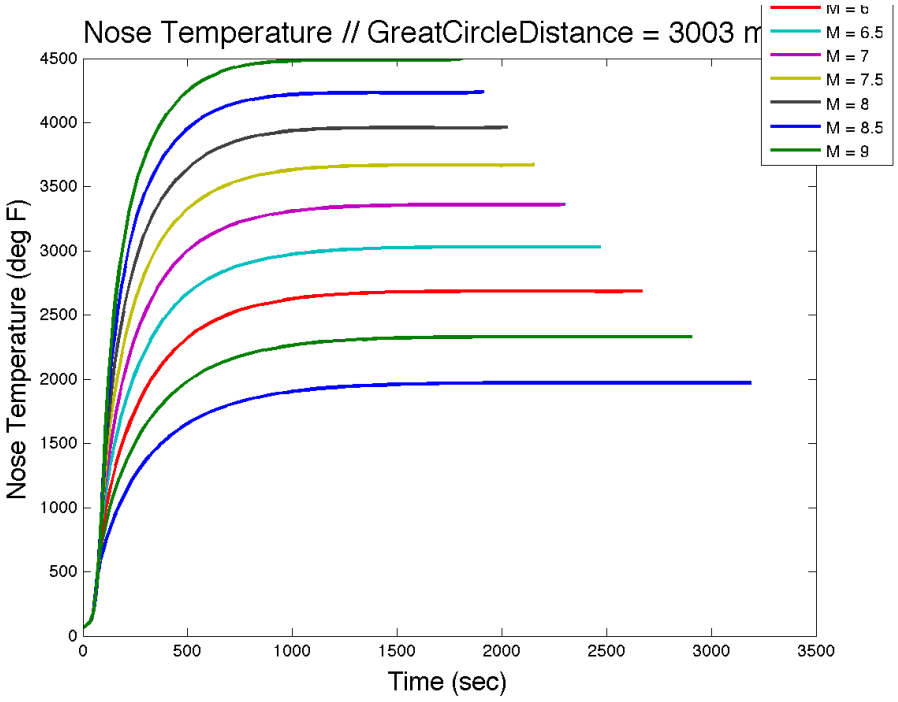


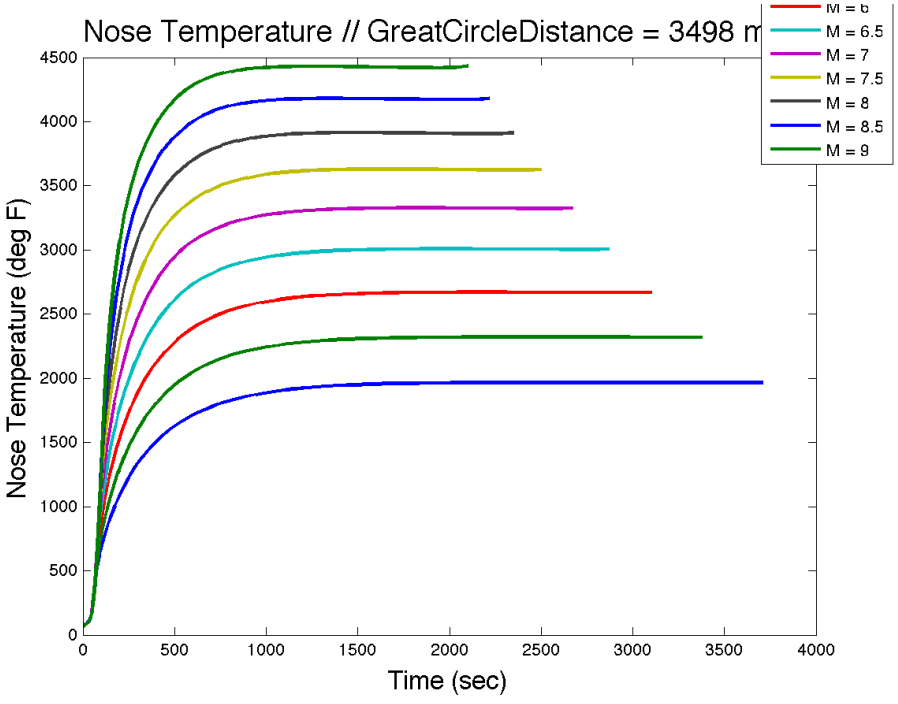

Nose Temperature // GreatCircleDistance = 999 m
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9
Nose Temperature (deg F)
Time (sec)
0
500
1000
1500
2000
2500
3000
3500
4000
4500
5000
0
200
400
600
800
1000
1200

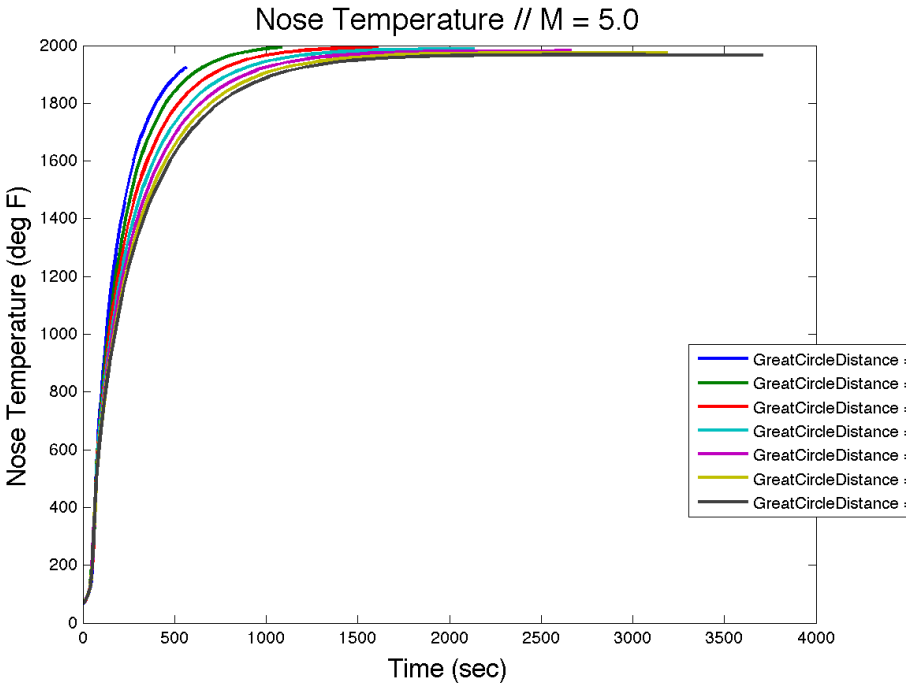

Nose Temperature // GreatCircleDistance = 1498 m
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9
Nose Temperature (deg F)
Time (sec)
0
500
1000
1500
2000
2500
3000
3500
4000
4500
5000
0
200
400
600
800
1000
1200
1400
1600
1800

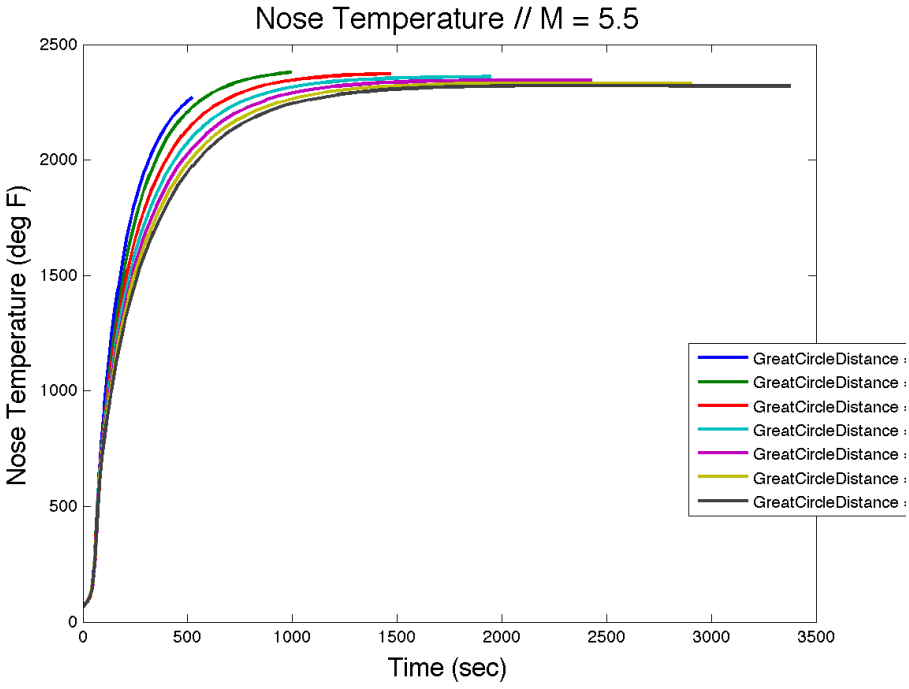

Nose Temperature // GreatCircleDistance = 1998 m
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9
Nose Temperature (deg F)
Time (sec)
0
500
1000
1500
2000
2500
3000
3500
4000
4500
5000

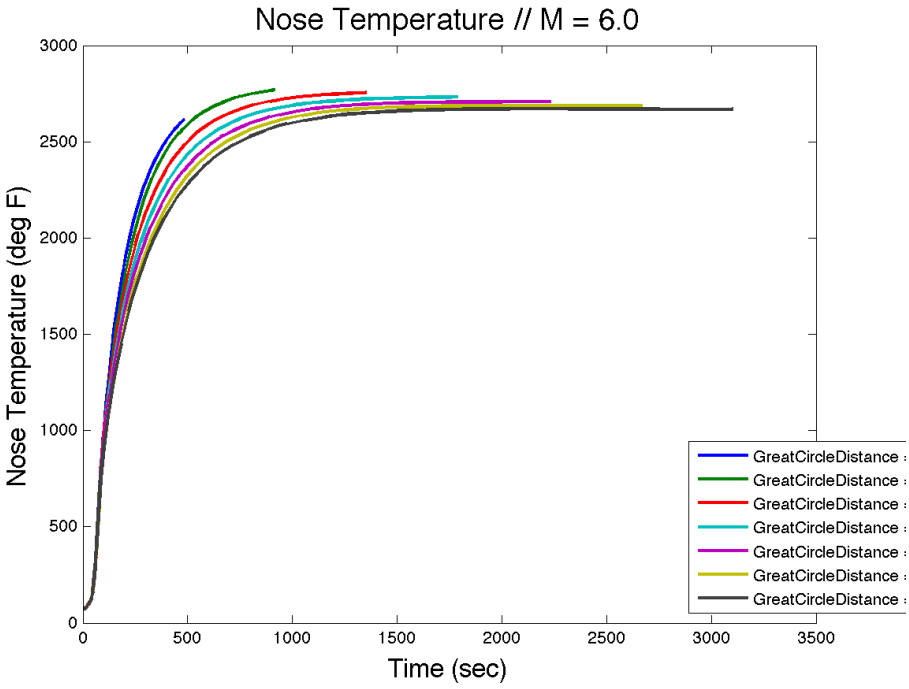

Nose Temperature // GreatCircleDistance = 2503 m
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9
Nose Temperature (deg F)
Time (sec)
0
500
1000
1500
2000
2500
3000
3500
4000
4500
5000

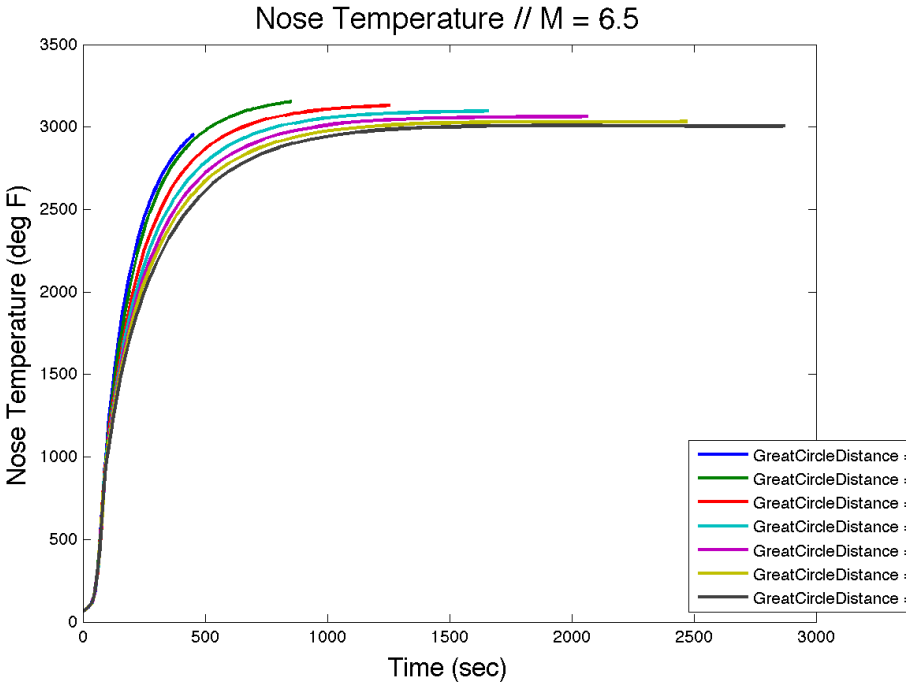

Nose Temperature // GreatCircleDistance = 3003 m
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9
Nose Temperature (deg F)
Time (sec)
0
500
1000
1500
2000
2500
3000
3500
4000
4500

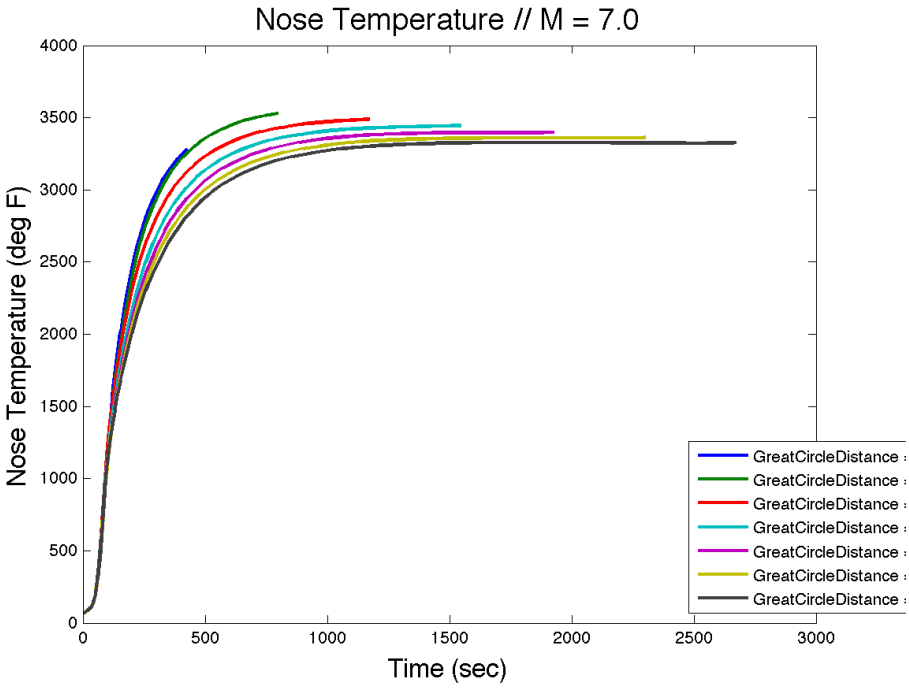

Nose Temperature // GreatCircleDistance = 3498 m
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9
Nose Temperature (deg F)
Time (sec)
0
500
1000
1500
2000
2500
3000
3500
4000
4500

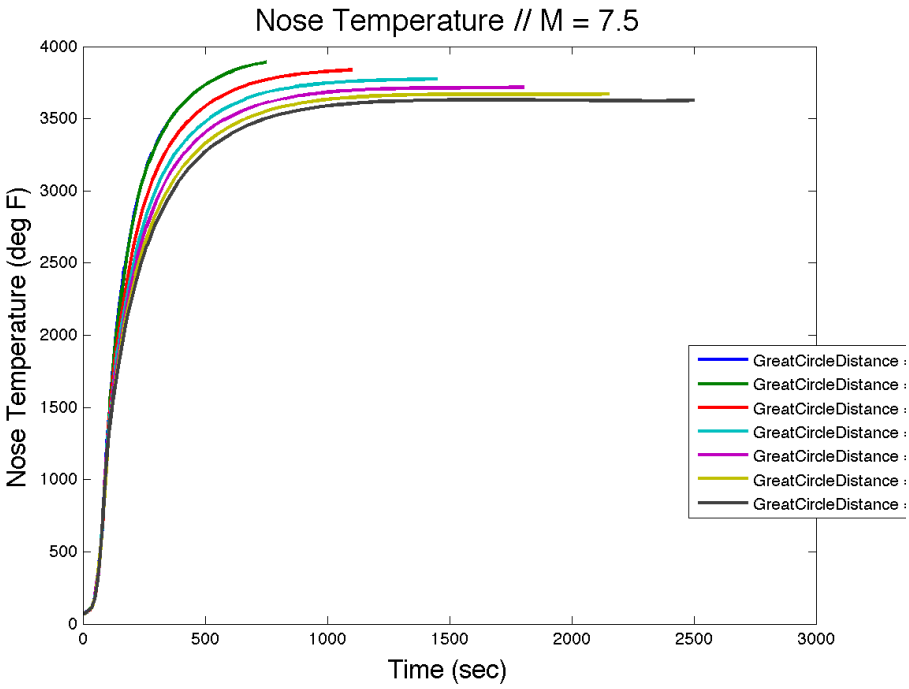

Nose Temperature // M = 5.0
Nose Temperature (deg F)
Time (sec)
0
200
400
600
800
1000
1200
1400
1600
1800
2000
0
500
1000
1500
2000
2500
3000
3500
4000
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance

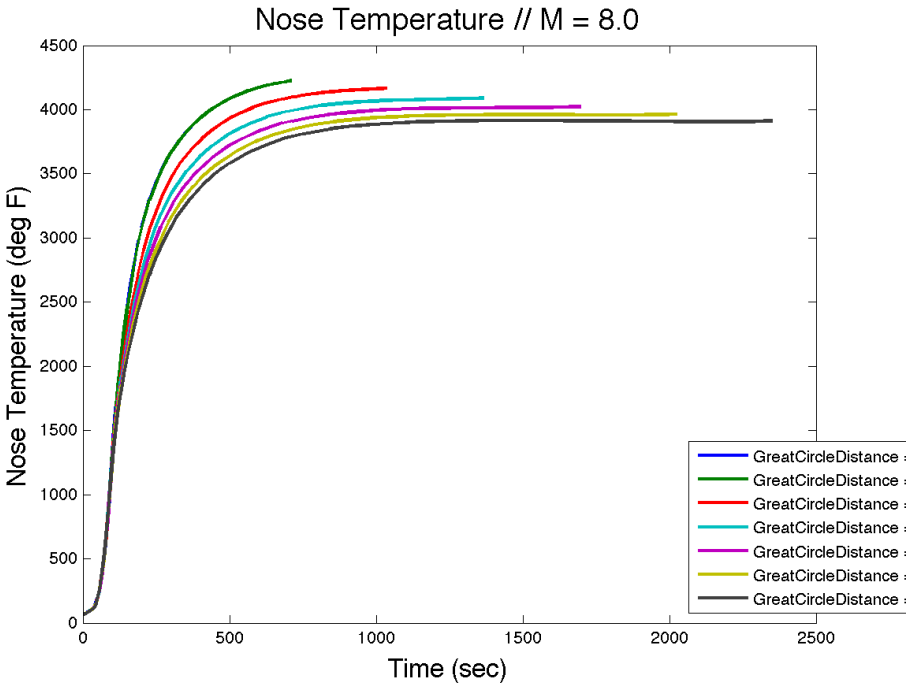

Nose Temperature // M = 5.5
Nose Temperature (deg F)
Time (sec)
0
500
1000
1500
2000
2500
0
500
1000
1500
2000
2500
3000
3500
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance

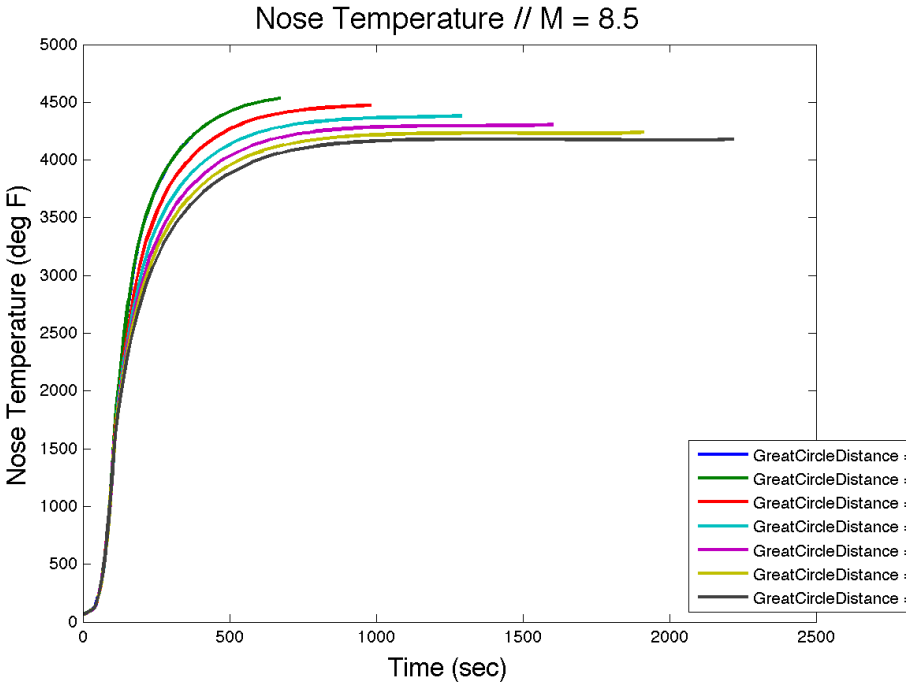

Nose Temperature // M = 6.0
Nose Temperature (deg F)
Time (sec)
0
500
1000
1500
2000
2500
3000
3500
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance

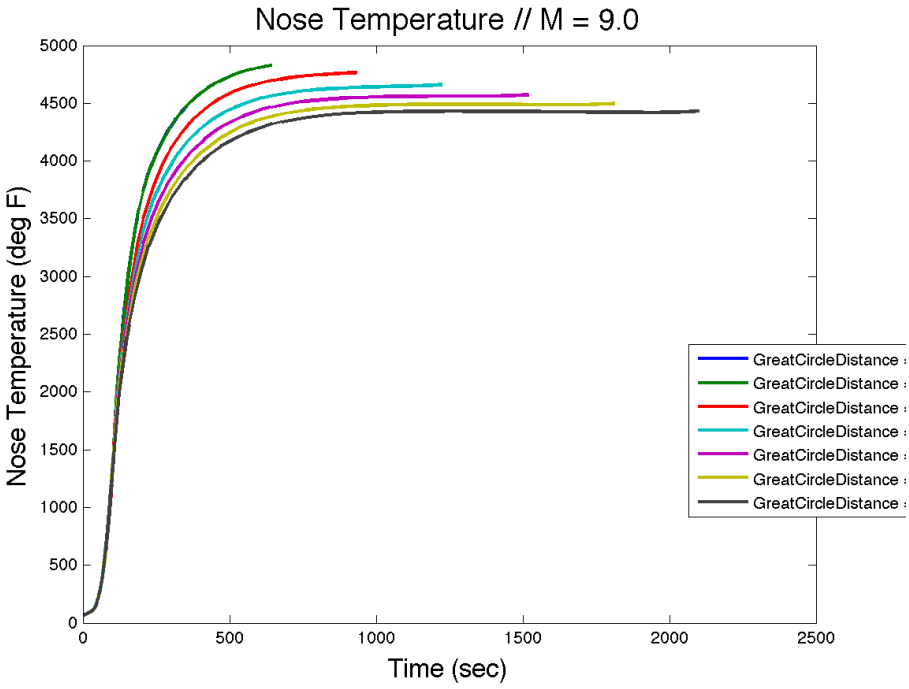

Nose Temperature // M = 6.5
Nose Temperature (deg F)
Time (sec)
0
500
1000
1500
2000
2500
3000
3500
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance


Nose Temperature // M = 7.0
Nose Temperature (deg F)
Time (sec)
0
500
1000
1500
2000
2500
3000
3500
4000
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance

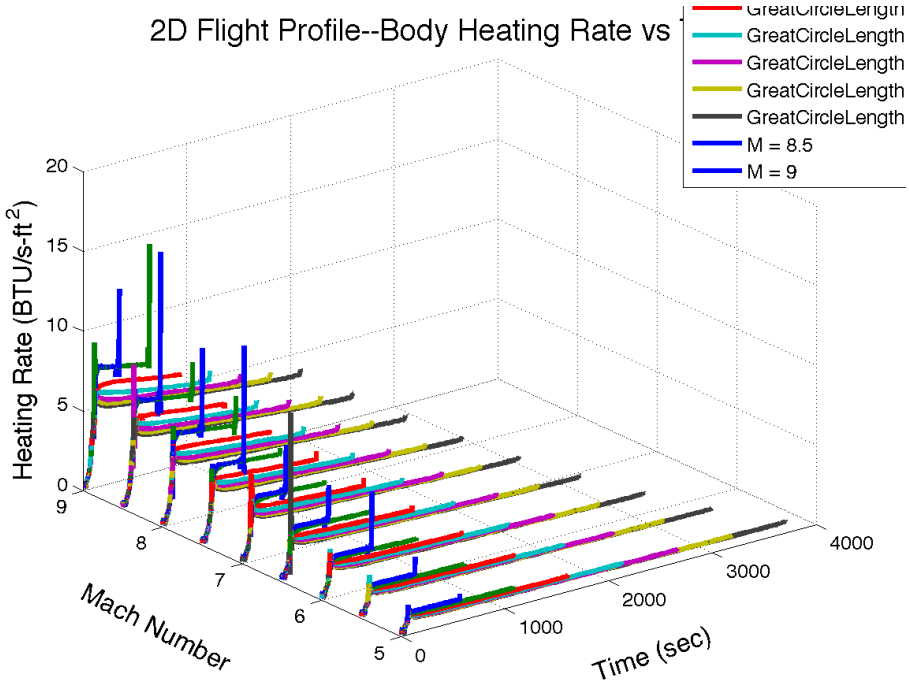

Nose Temperature // M = 7.5
Nose Temperature (deg F)
Time (sec)
0
500
1000
1500
2000
2500
3000
3500
4000
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance

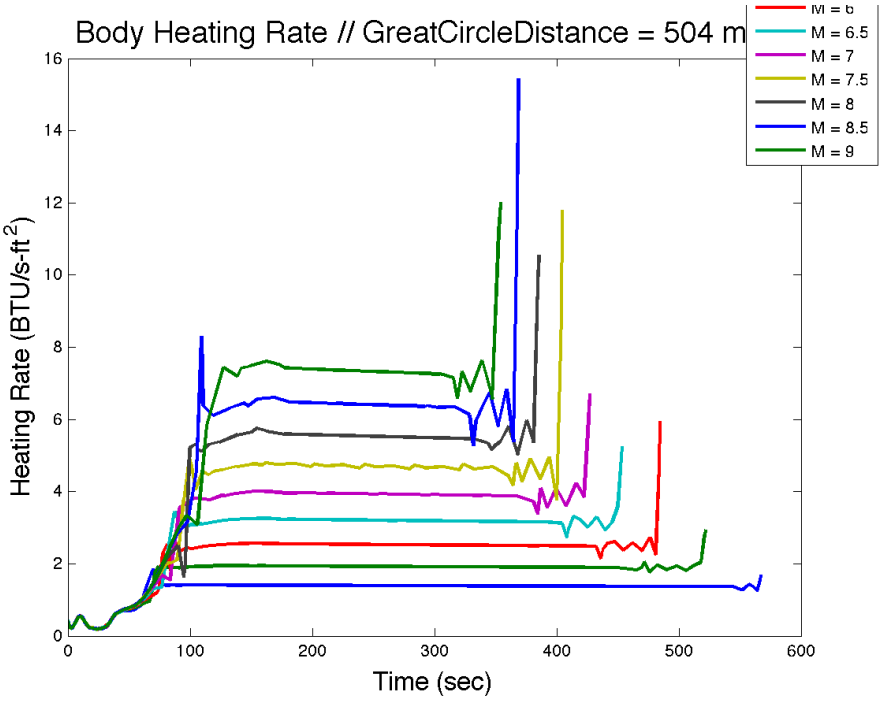

Nose Temperature // M = 8.0
Nose Temperature (deg F)
Time (sec)
0
500
1000
1500
2000
2500
3000
3500
4000
4500
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance

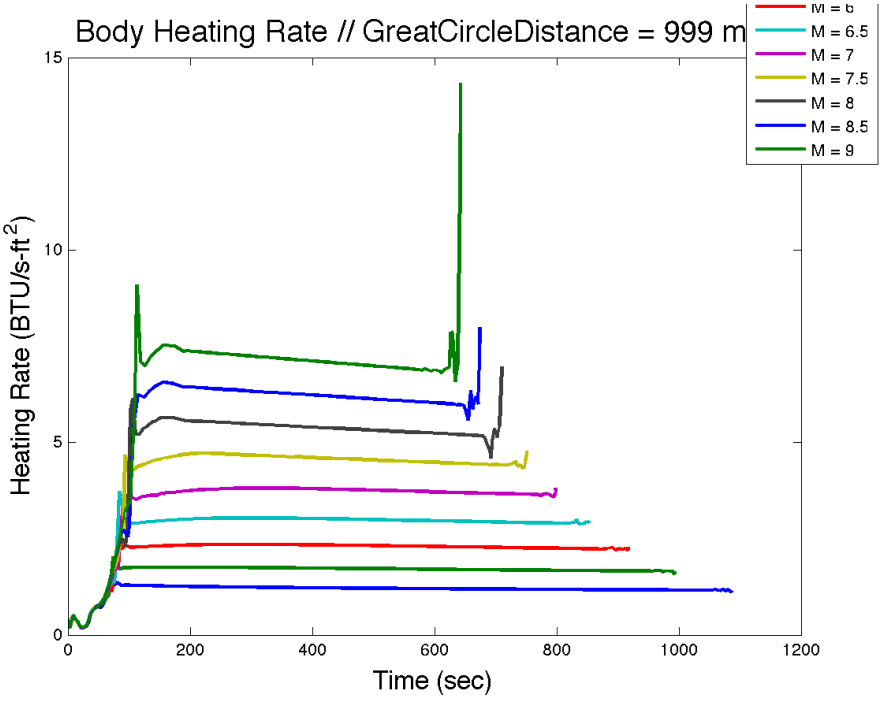

Nose Temperature // M = 8.5
Nose Temperature (deg F)
Time (sec)
0
500
1000
1500
2000
2500
3000
3500
4000
4500
5000
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance

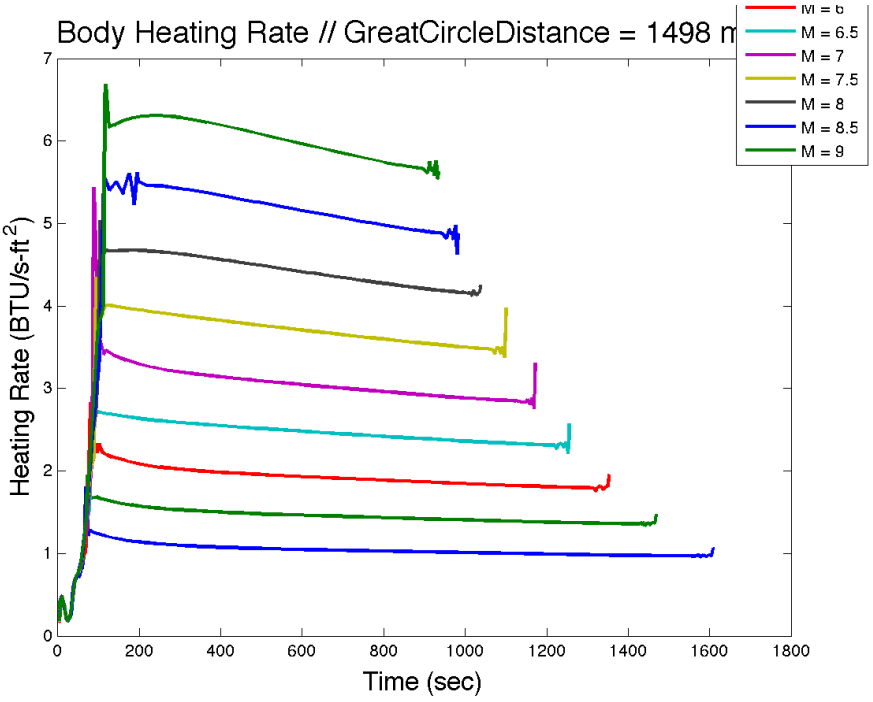


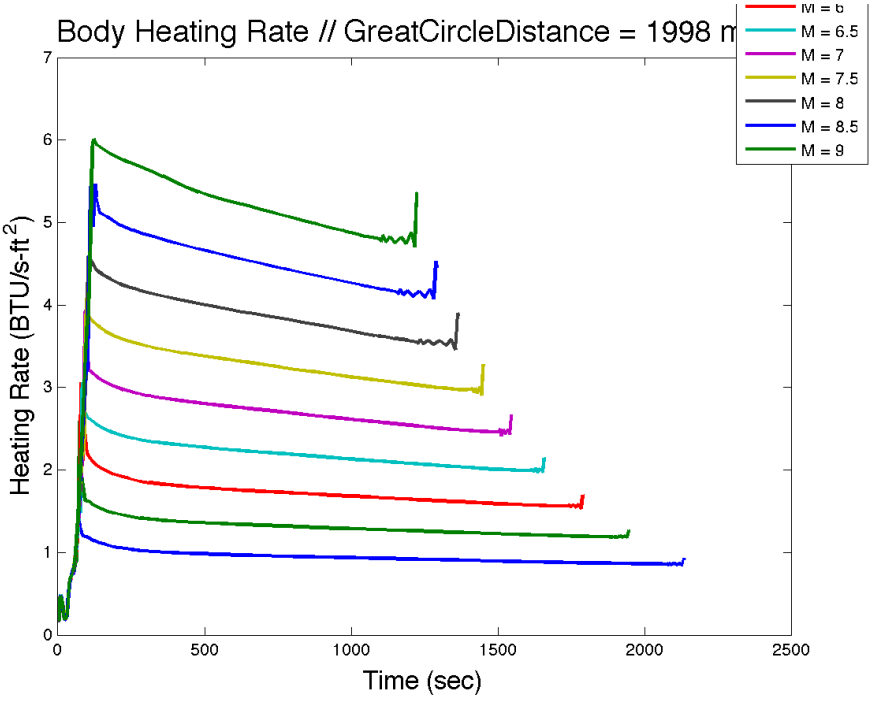
D.6.4 Body Heating Rate vs. Time.

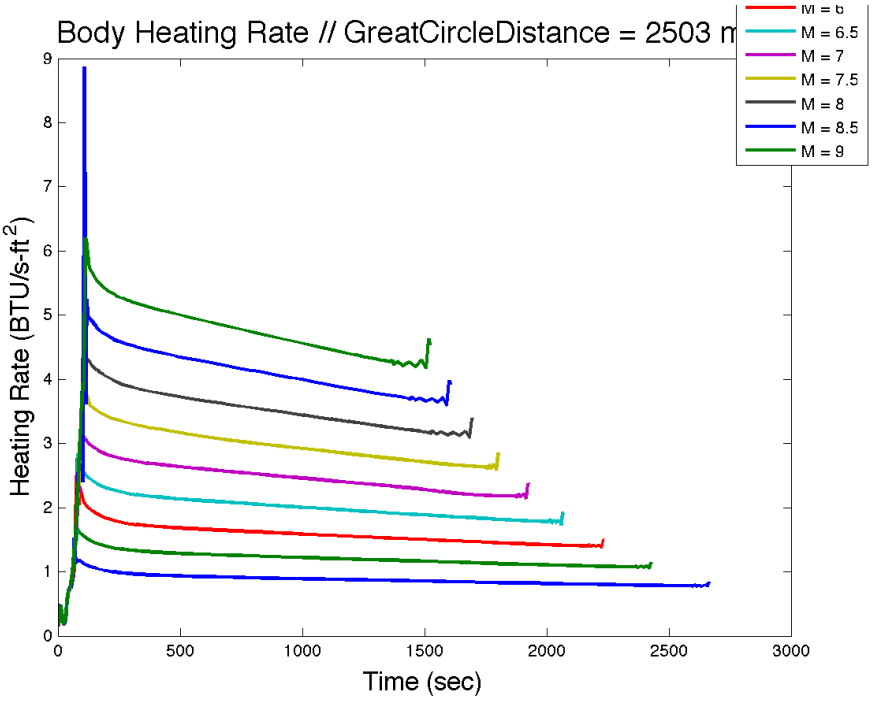


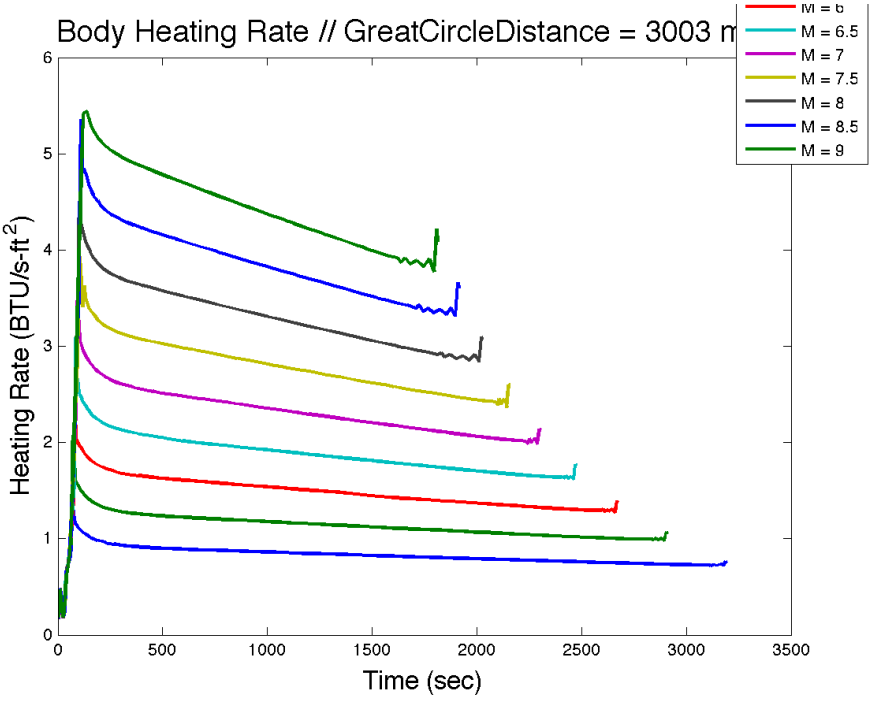

Body Heating Rate // GreatCircleDistance = 504 m
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9
Heating Rate (BTU/s-ft^2)
Time (sec)
0
2
4
6
8
10
12
14
16
0
100
200
300
400
500
600

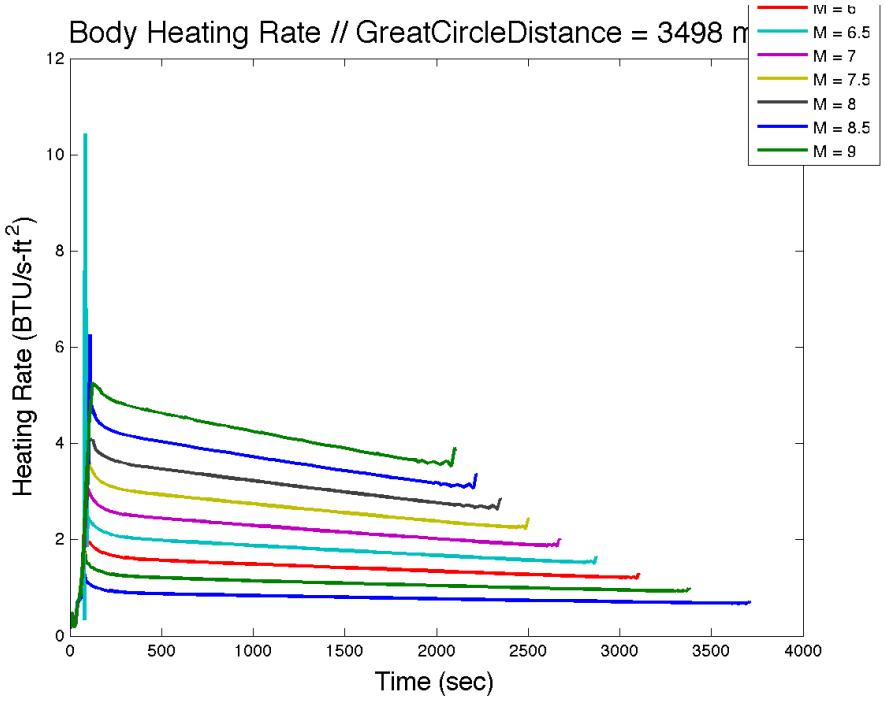

Body Heating Rate // GreatCircleDistance = 999 m
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9
Heating Rate (BTU/s-ft^2)
Time (sec)
0
5
10
15
0
200
400
600
800
1000
1200

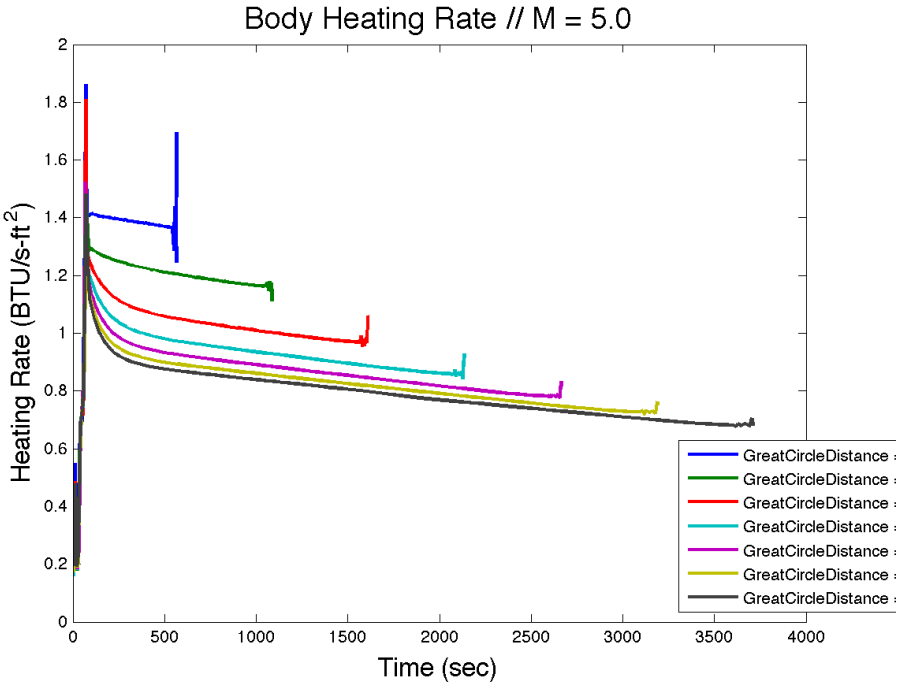

Body Heating Rate // GreatCircleDistance = 1498 m
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9
Heating Rate (BTU/s-ft^2)
Time (sec)
0
1
2
3
4
5
6
7
0
200
400
600
800
1000
1200
1400
1600
1800

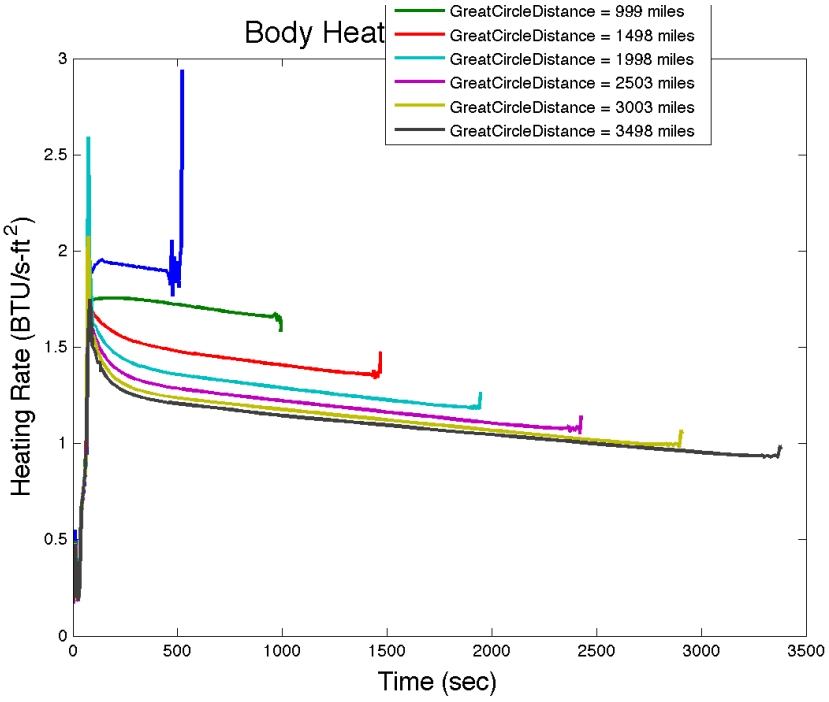

Body Heating Rate // GreatCircleDistance = 1998 m
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9
Heating Rate (BTU/s-ft^2)
Time (sec)
0
1
2
3
4
5
6
7
0
500
1000
1500
2000
2500

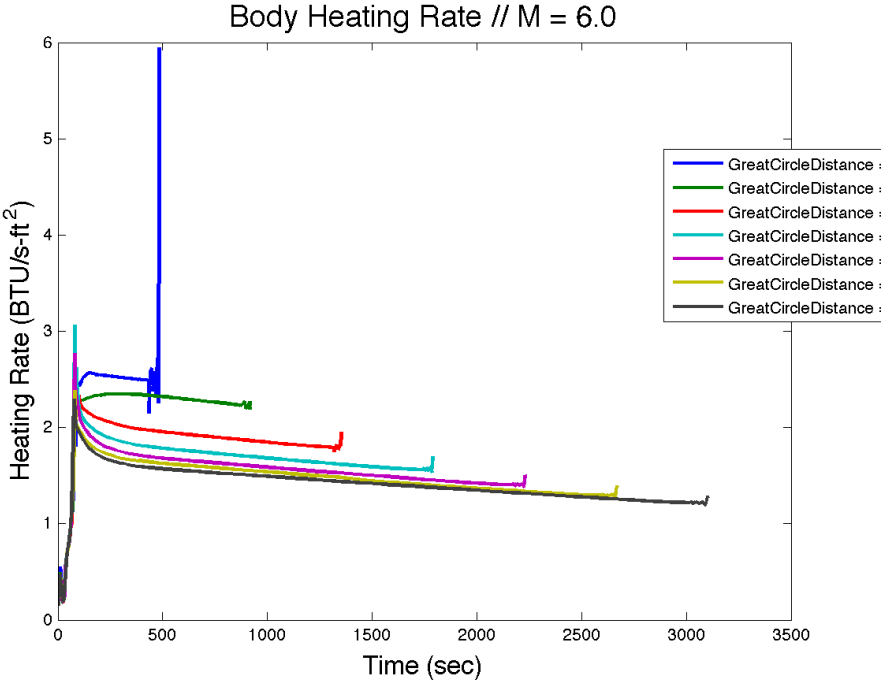

Body Heating Rate // GreatCircleDistance = 2503 m
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9
Heating Rate (BTU/s-ft^2)
Time (sec)
0
1
2
3
4
5
6
7
8
9
0
500
1000
1500
2000
2500
3000

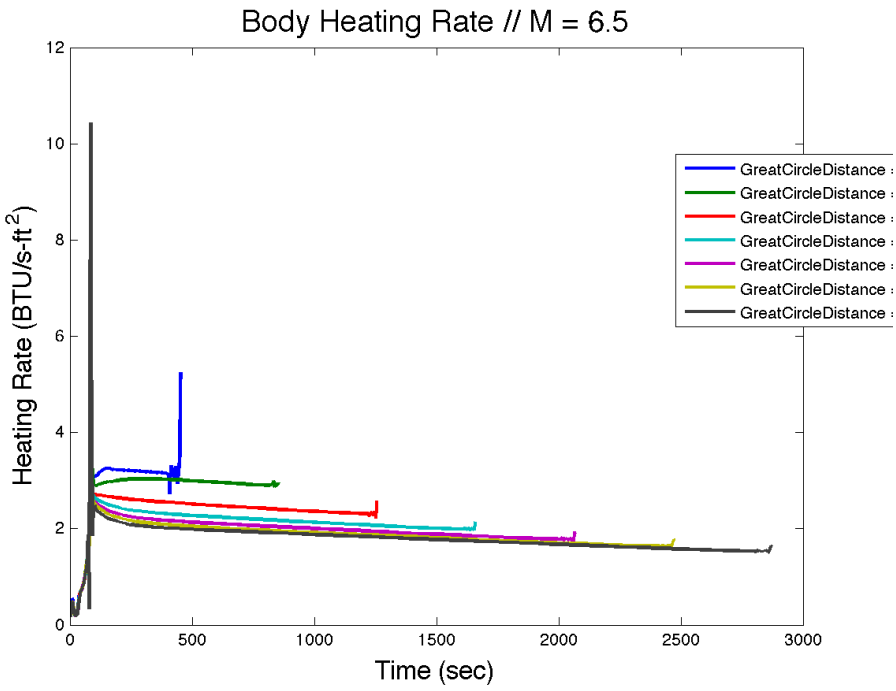

Body Heating Rate // GreatCircleDistance = 3003 m
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9
Heating Rate (BTU/s-ft^2)
Time (sec)
0
1
2
3
4
5
6
0
500
1000
1500
2000
2500
3000
3500

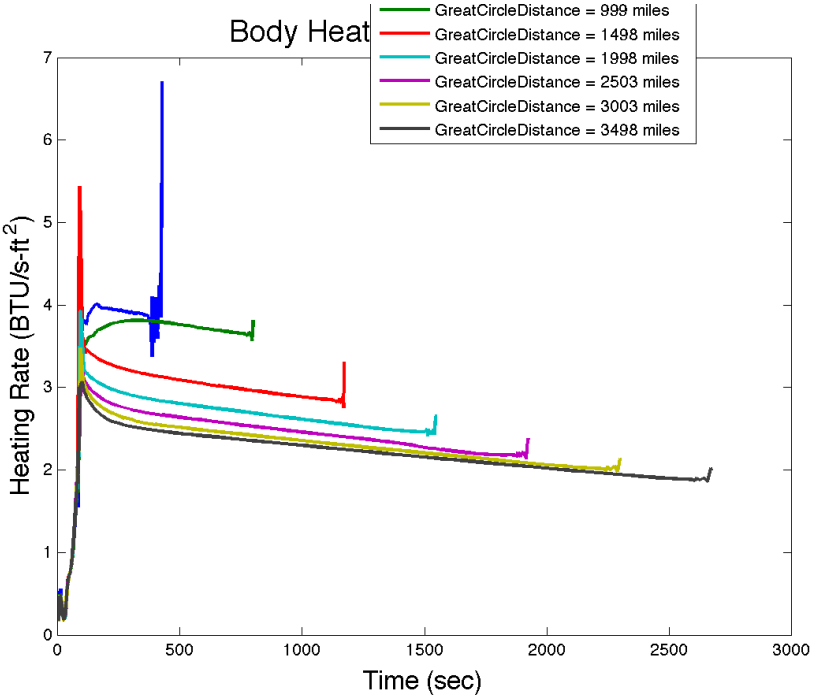

Body Heating Rate // GreatCircleDistance = 3498 m
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9
Heating Rate (BTU/s-ft^2)
Time (sec)
0
2
4
6
8
10
12
0
500
1000
1500
2000
2500
3000
3500
4000

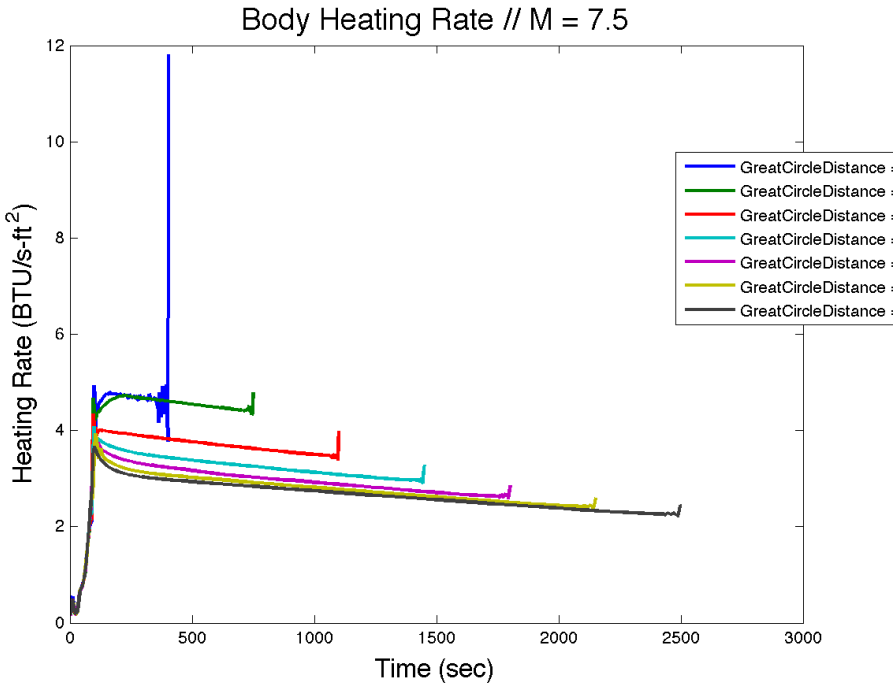

Body Heating Rate // M = 5.0
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
Heating Rate (BTU/s-ft^2)
Time (sec)
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
0
500
1000
1500
2000
2500
3000
3500
4000

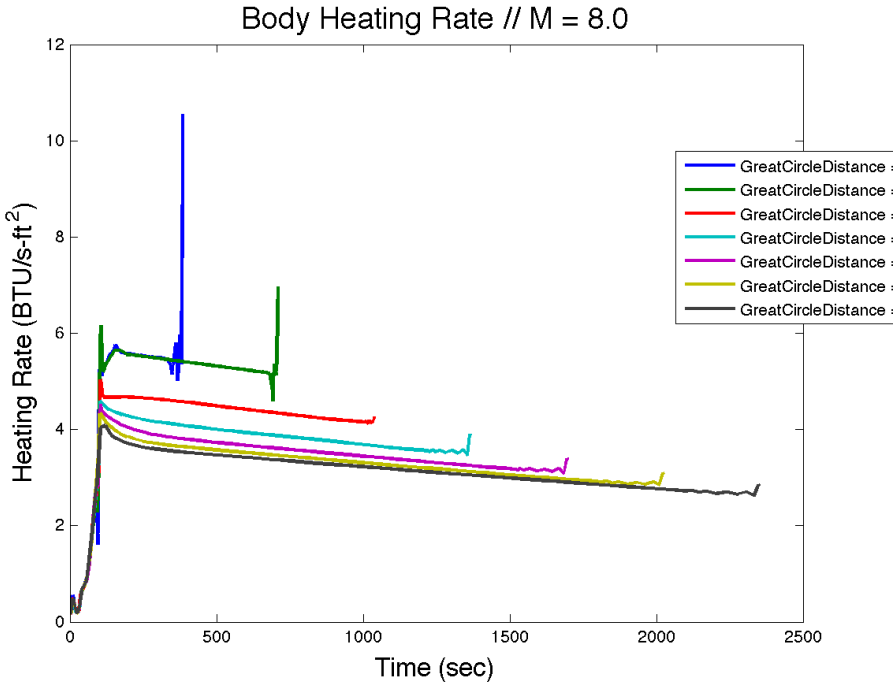

Body Heat
GreatCircleDistance = 999 miles
GreatCircleDistance = 1498 miles
GreatCircleDistance = 1998 miles
GreatCircleDistance = 2503 miles
GreatCircleDistance = 3003 miles
GreatCircleDistance = 3498 miles
Heating Rate (BTU/s-ft^2)
Time (sec)
0
0.5
1
1.5
2
2.5
3
0
500
1000
1500
2000
2500
3000
3500

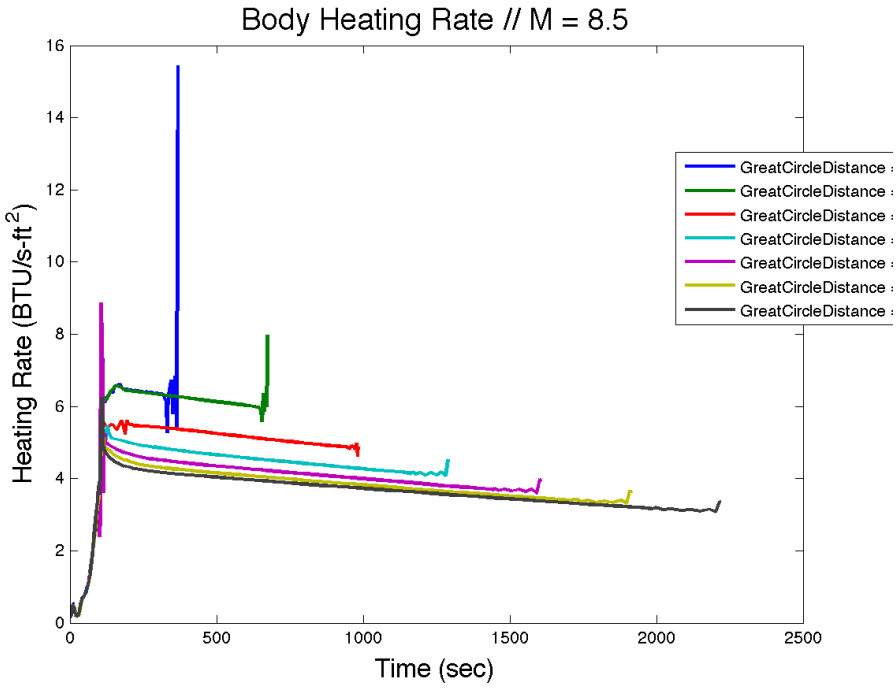

Body Heating Rate // M = 6.0
Heating Rate (BTU/s-ft^2)
Time (sec)
0
1
2
3
4
5
6
0
500
1000
1500
2000
2500
3000
3500

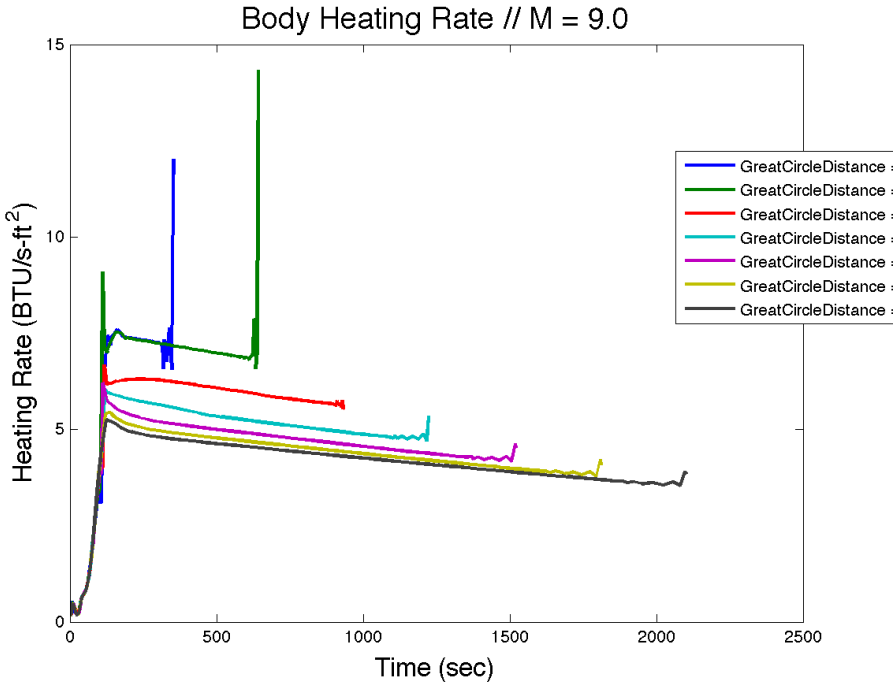

Body Heating Rate // M = 6.5
Heating Rate (BTU/s-ft^2)
Time (sec)
0
2
4
6
8
10
12
0
500
1000
1500
2000
2500
3000
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance


Body Heat
GreatCircleDistance = 999 miles
GreatCircleDistance = 1498 miles
GreatCircleDistance = 1998 miles
GreatCircleDistance = 2503 miles
GreatCircleDistance = 3003 miles
GreatCircleDistance = 3498 miles
Heating Rate (BTU/s-ft^2)
Time (sec)
0
1
2
3
4
5
6
7
0
500
1000
1500
2000
2500
3000

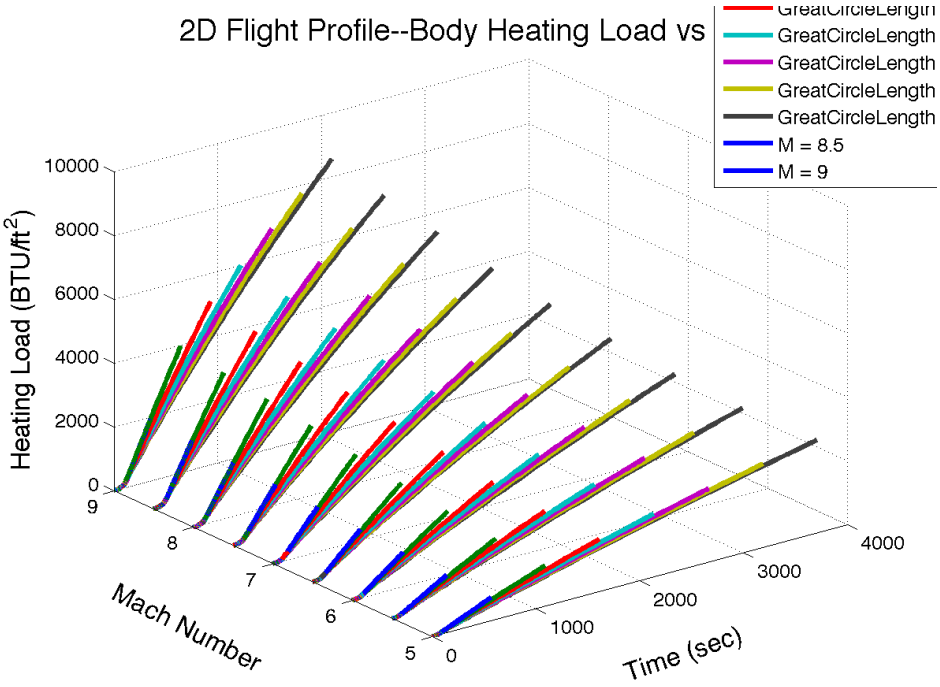

Body Heating Rate // M = 7.5
Heating Rate (BTU/s-ft^2)
Time (sec)
0
2
4
6
8
10
12
0
500
1000
1500
2000
2500
3000
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance

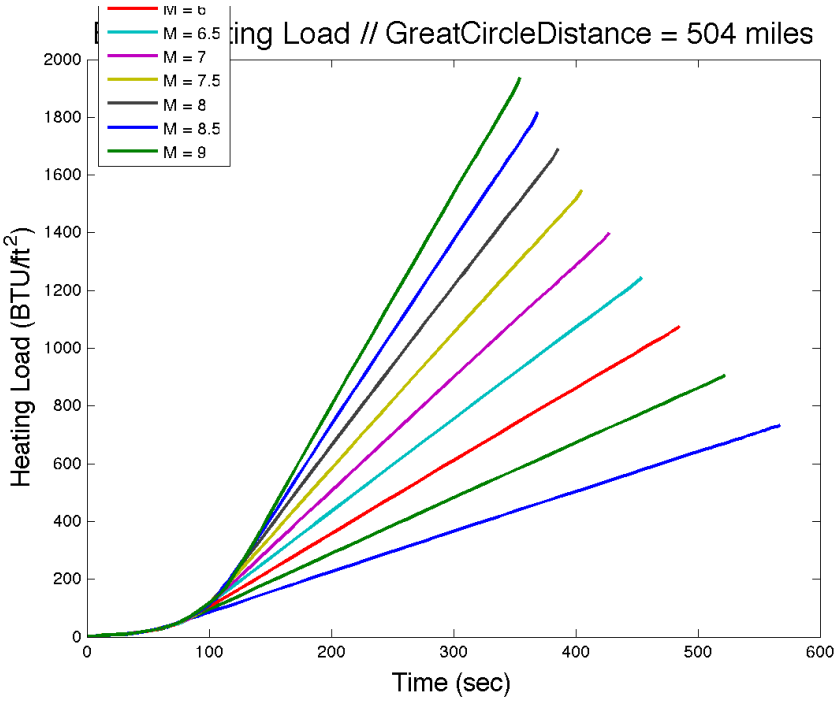

Body Heating Rate // M = 8.0
Heating Rate (BTU/s-ft^2)
Time (sec)
0
2
4
6
8
10
12
0
500
1000
1500
2000
2500
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance

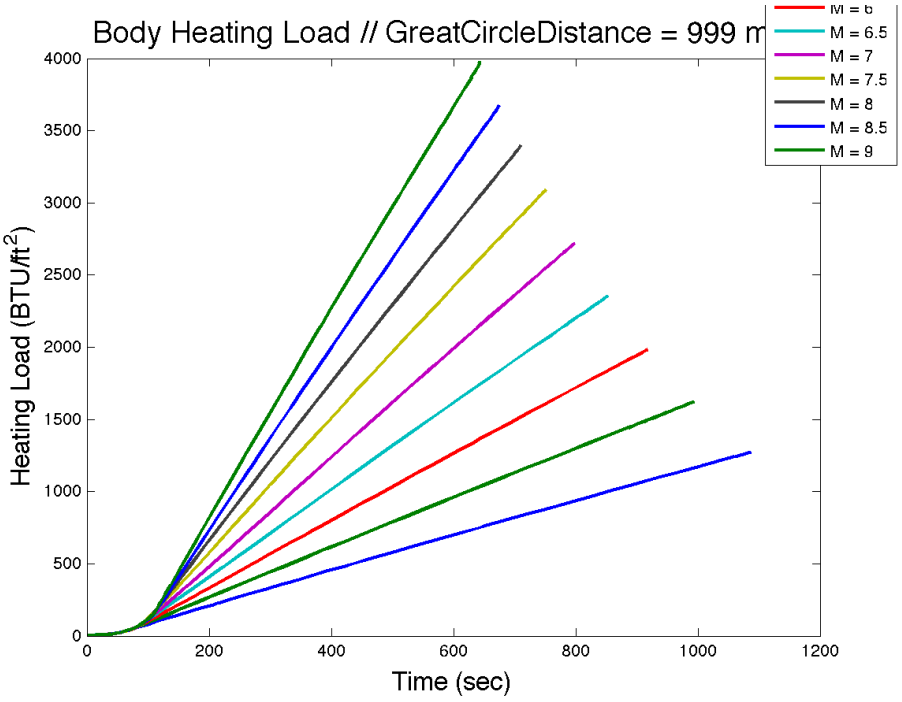

Body Heating Rate // M = 8.5
Heating Rate (BTU/s-ft^2)
Time (sec)
0
2
4
6
8
10
12
14
16
0
500
1000
1500
2000
2500
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance

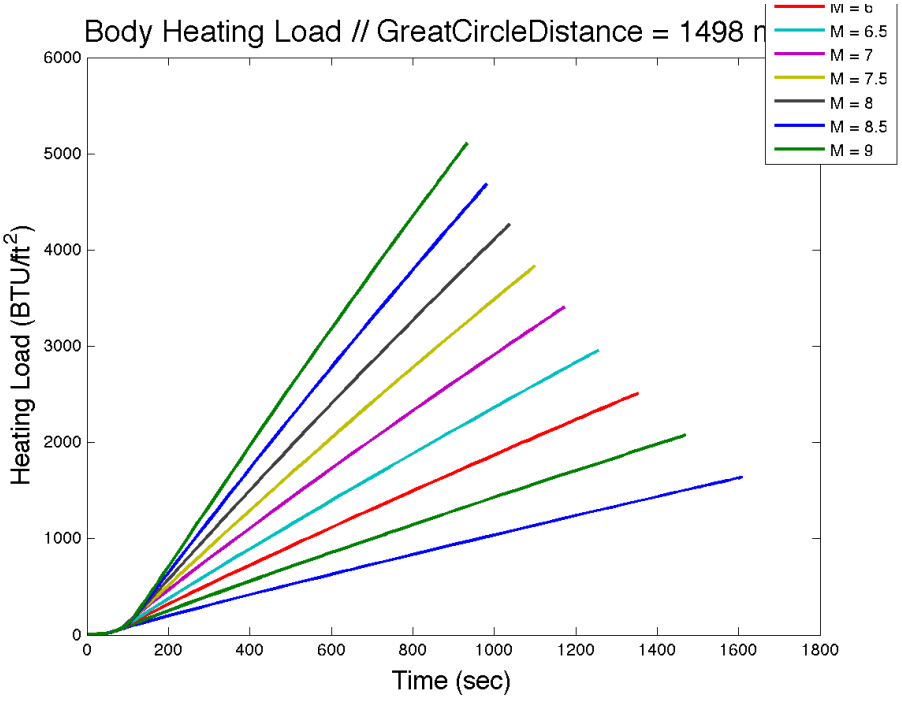

Body Heating Rate // M = 9.0
Heating Rate (BTU/s-ft^2)
Time (sec)
0
5
10
15
0
500
1000
1500
2000
2500
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance

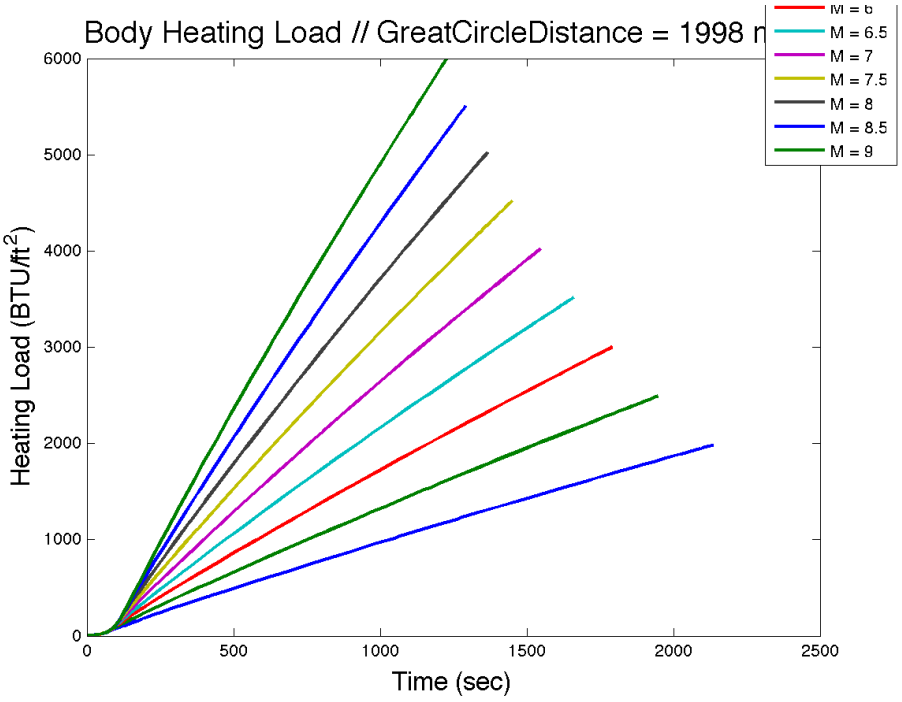
D.6.5 Body Heating Load vs. Time.

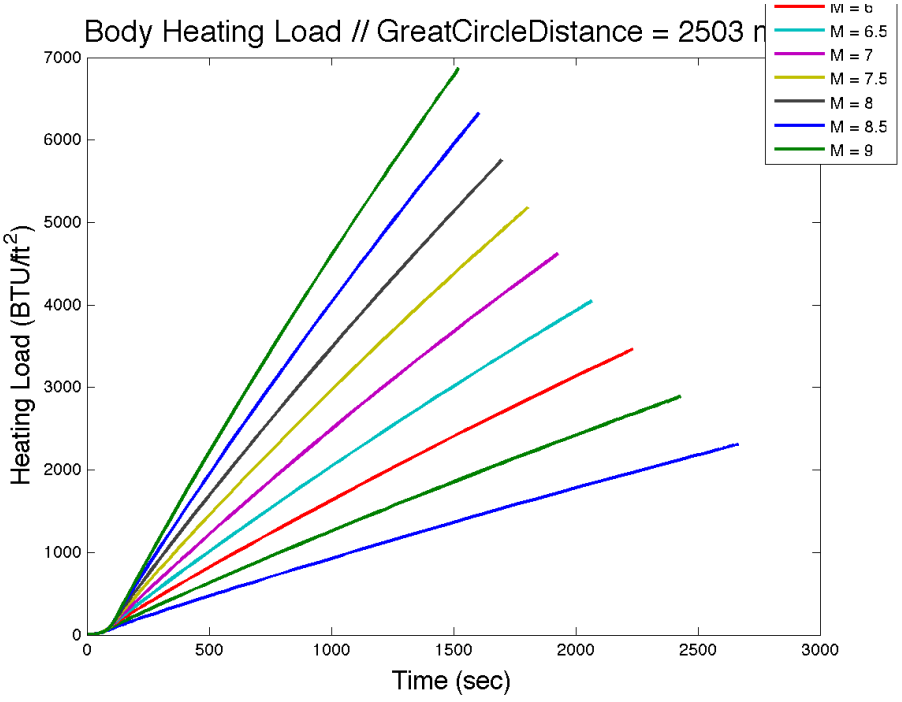


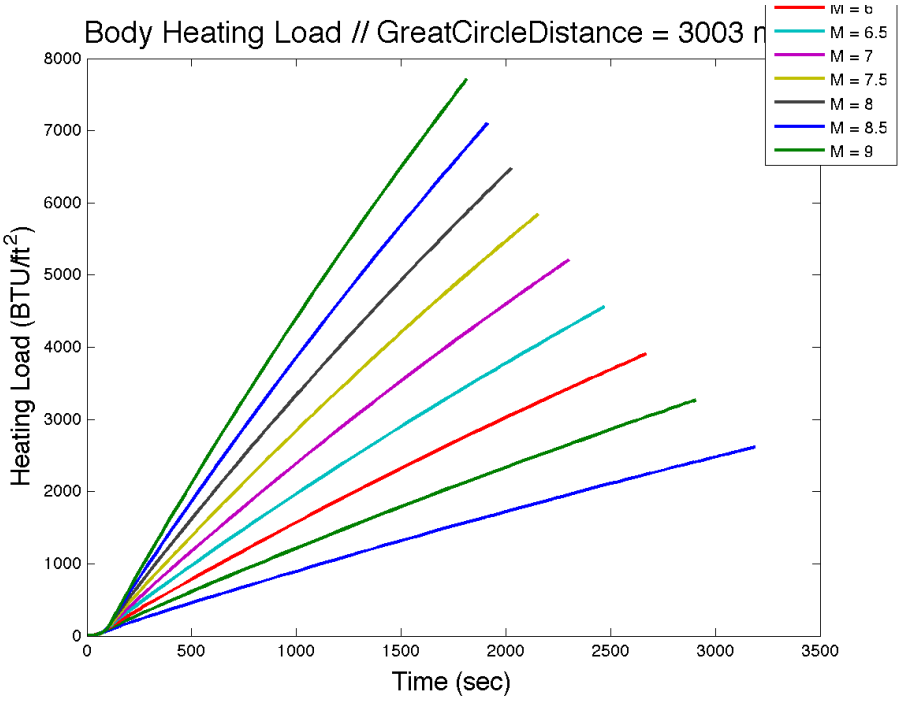


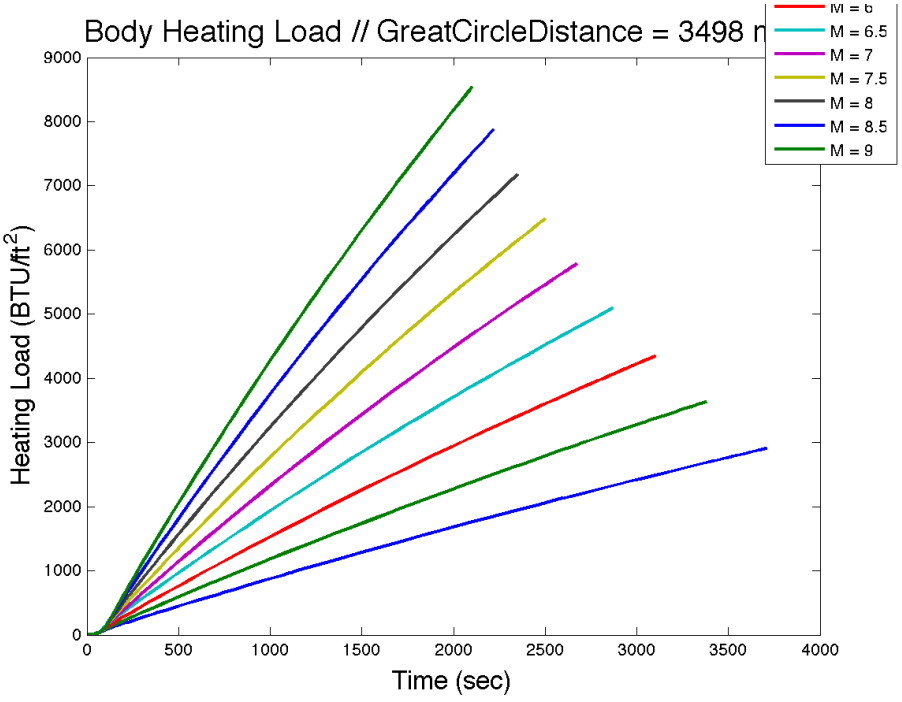

Body Heating Load // GreatCircleDistance = 999 m
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9
Heating Load (BTU/ft2)
Time (sec)
0
500
1000
1500
2000
2500
3000
3500
4000
0
200
400
600
800
1000
1200

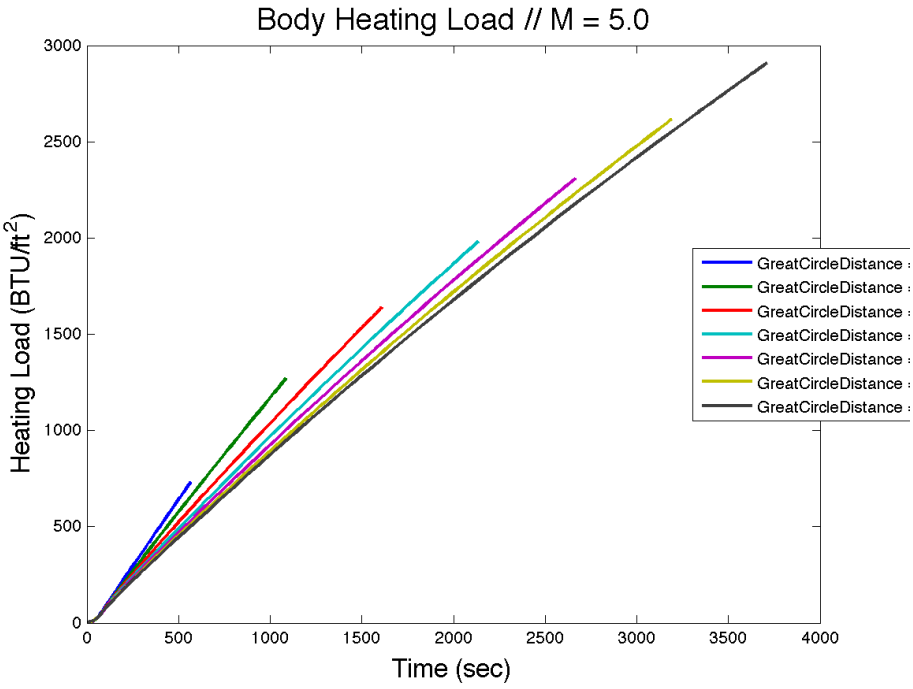

Body Heating Load // GreatCircleDistance = 1498 m
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9
Heating Load (BTU/ft2)
Time (sec)
0
1000
2000
3000
4000
5000
6000
0
200
400
600
800
1000
1200
1400
1600
1800

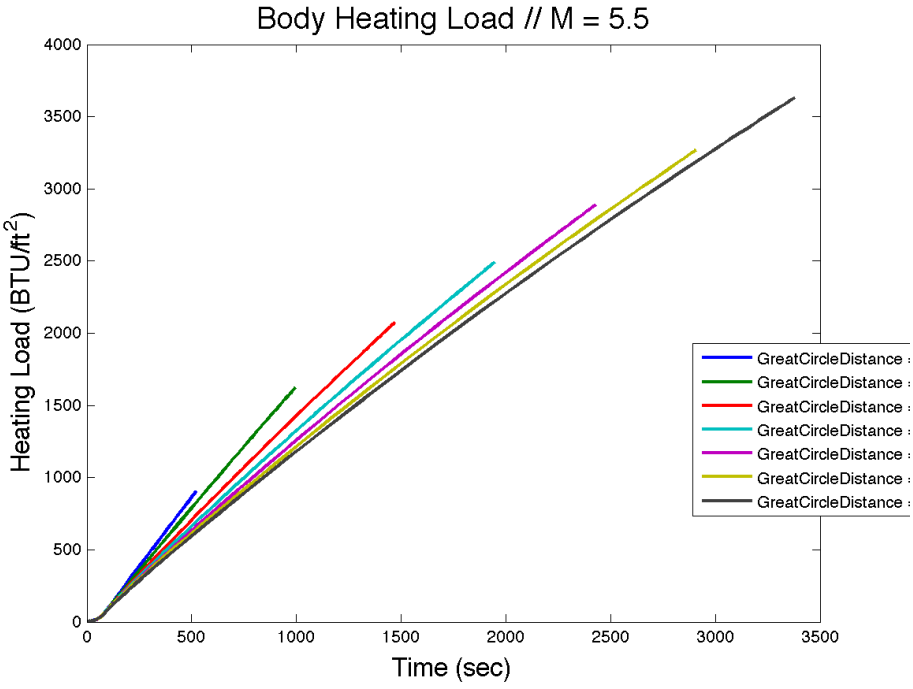

Body Heating Load // GreatCircleDistance = 1998 n
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9
Heating Load (BTU/ft2)
Time (sec)
0
1000
2000
3000
4000
5000
6000
0
500
1000
1500
2000
2500

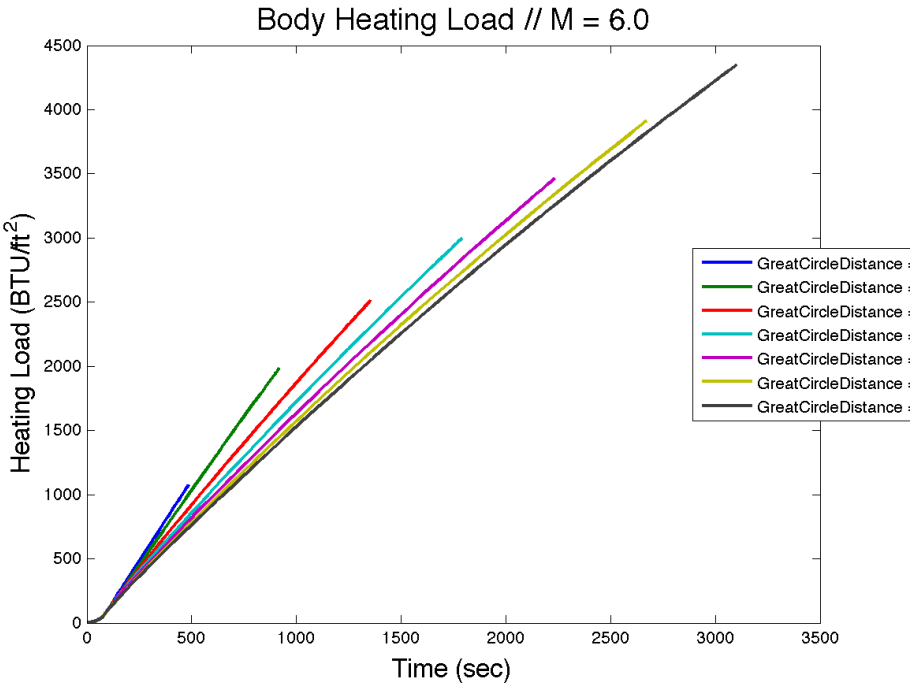

Body Heating Load // GreatCircleDistance = 2503 n
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9
Heating Load (BTU/ft2)
Time (sec)
0
1000
2000
3000
4000
5000
6000
7000
0
500
1000
1500
2000
2500
3000

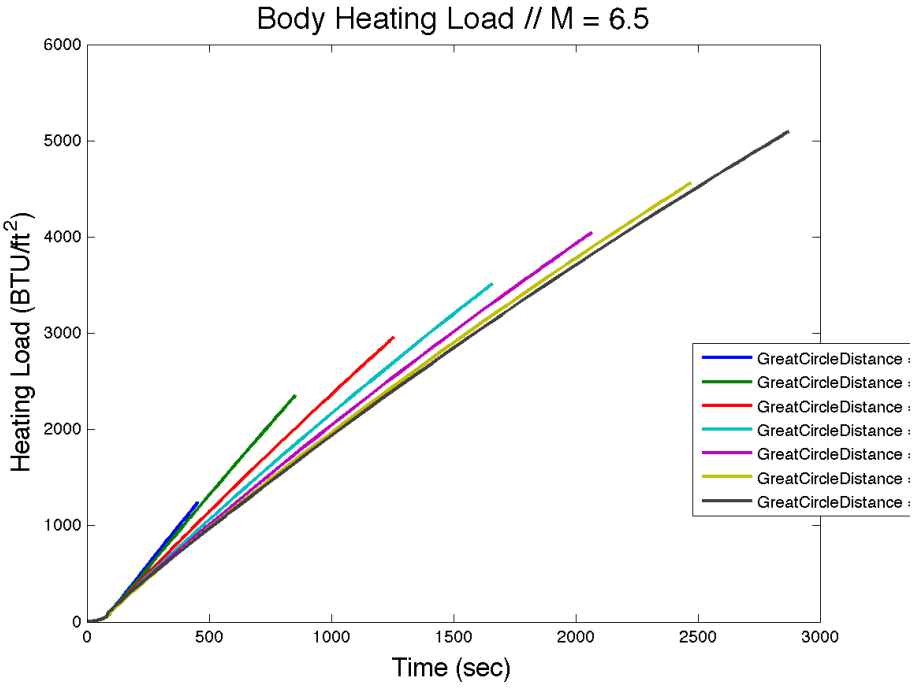

Body Heating Load // GreatCircleDistance = 3003 n
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9
Heating Load (BTU/ft2)
Time (sec)
0
1000
2000
3000
4000
5000
6000
7000
8000
500
1500
2500
3500

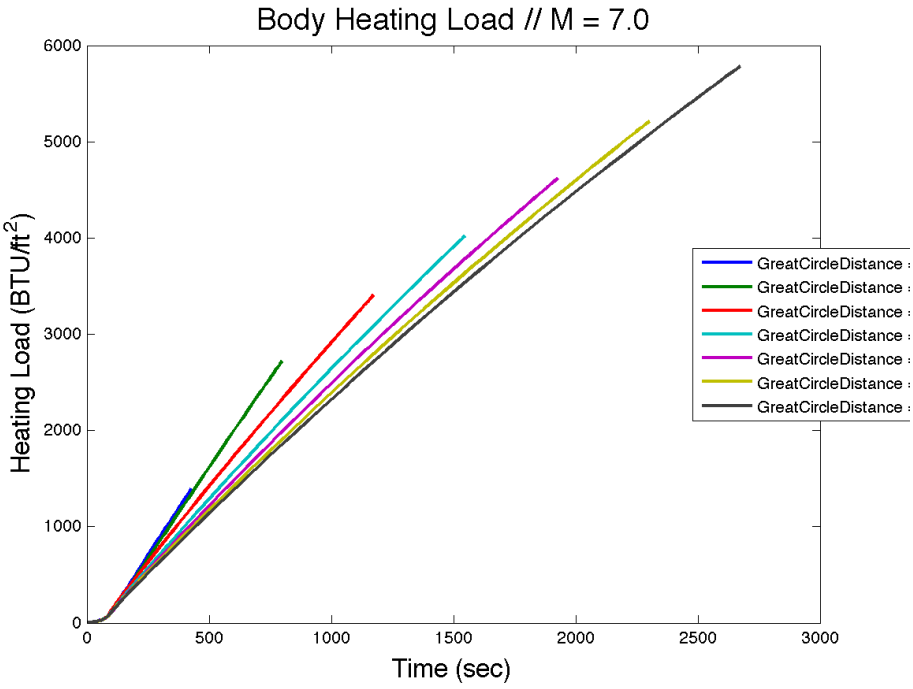

Body Heating Load // GreatCircleDistance = 3498 n
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9
Heating Load (BTU/ft2)
Time (sec)
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
500
1500
2500
3500

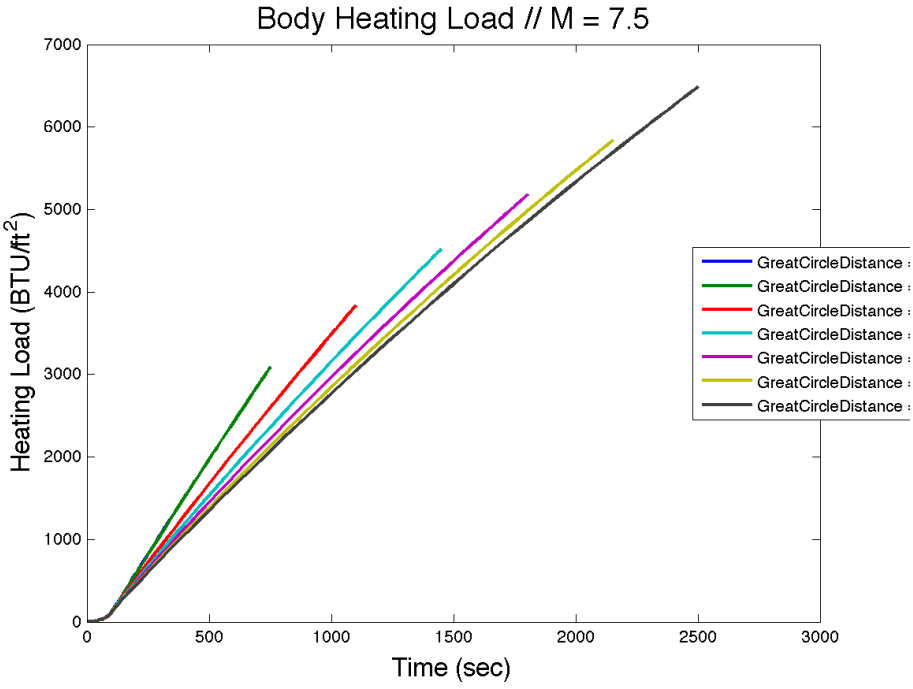

Body Heating Load // M = 5.0
Heating Load (BTU/ft^2)
Time (sec)
0
500
1000
1500
2000
2500
3000
3500
4000
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance

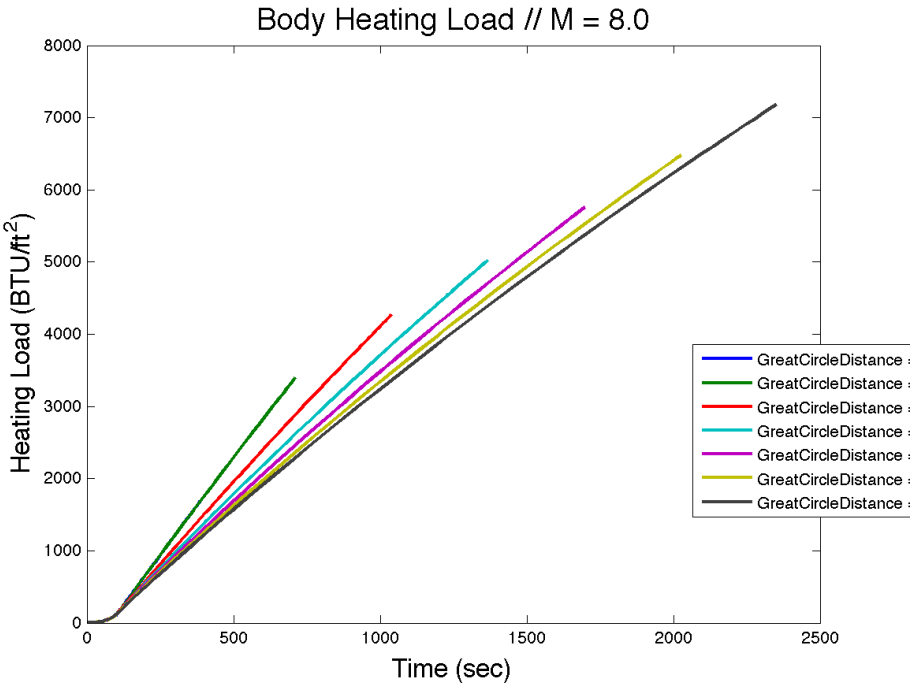

Body Heating Load // M = 5.5
Heating Load (BTU/ft^2)
Time (sec)
0
500
1000
1500
2000
2500
3000
3500
4000
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance

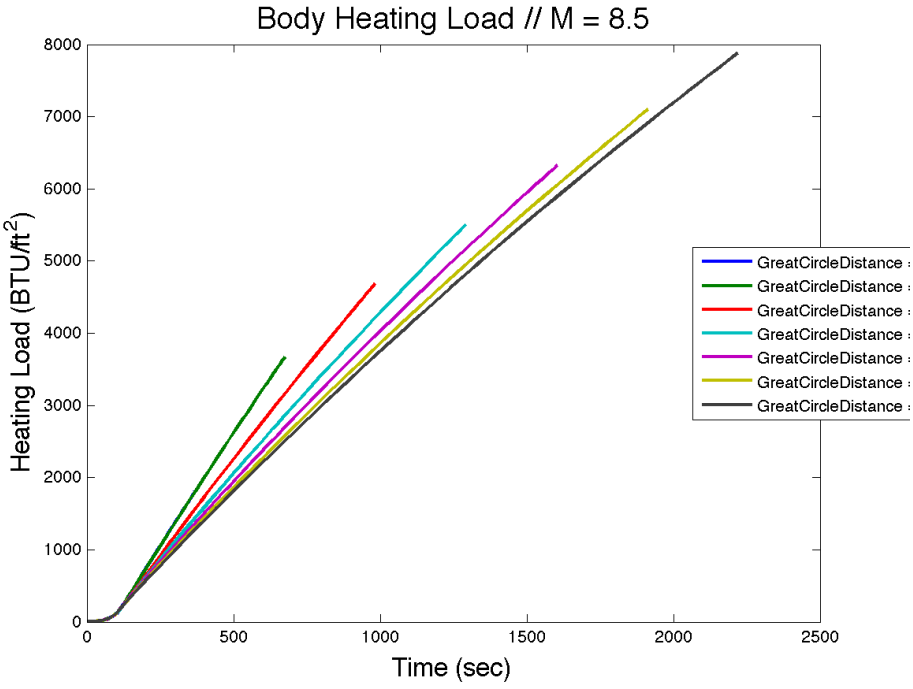

Body Heating Load // M = 6.0
Heating Load (BTU/ft^2)
Time (sec)
0
500
1000
1500
2000
2500
3000
3500
4000
4500
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance

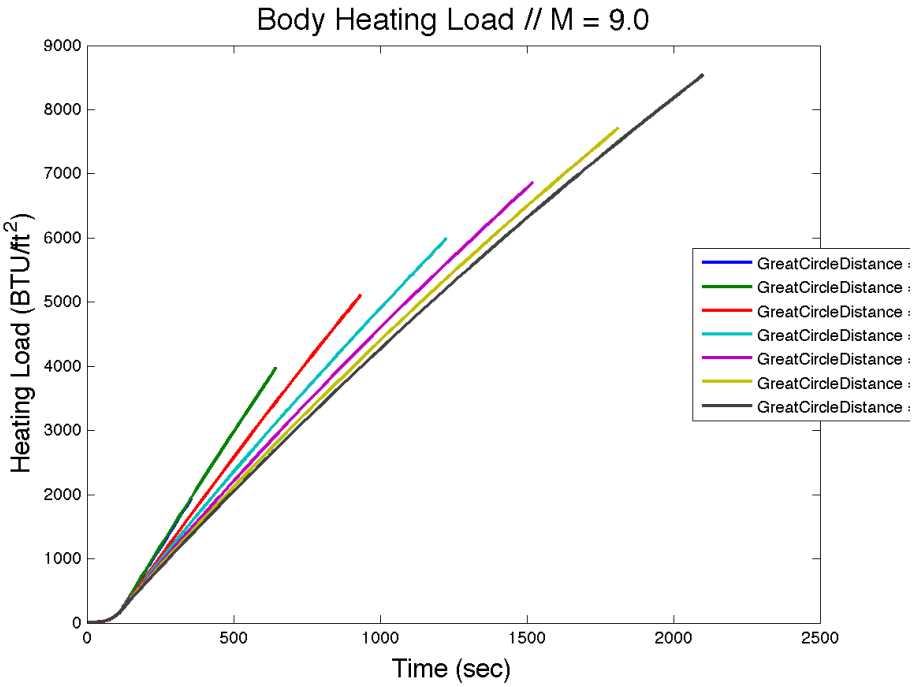

Body Heating Load // M = 6.5
Heating Load (BTU/ft^2)
Time (sec)
0
1000
2000
3000
4000
5000
6000
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance


Body Heating Load // M = 7.0
Heating Load (BTU/ft^2)
Time (sec)
0
1000
2000
3000
4000
5000
6000
0
500
1000
1500
2000
2500
3000
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance

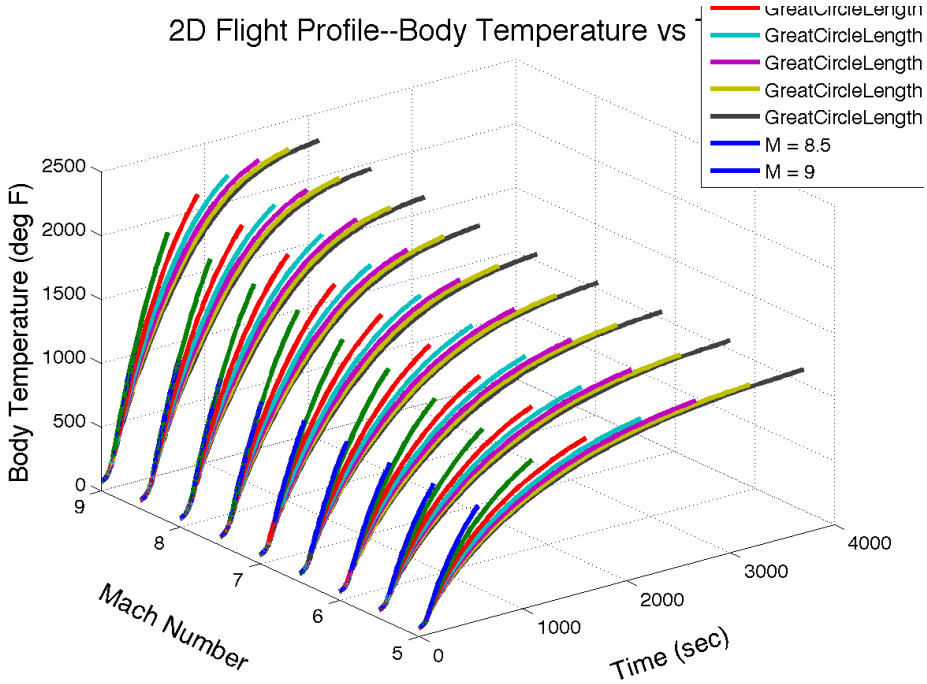

Body Heating Load // M = 7.5
Heating Load (BTU/ft^2)
Time (sec)
0
1000
2000
3000
4000
5000
6000
7000
0
500
1000
1500
2000
2500
3000
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance

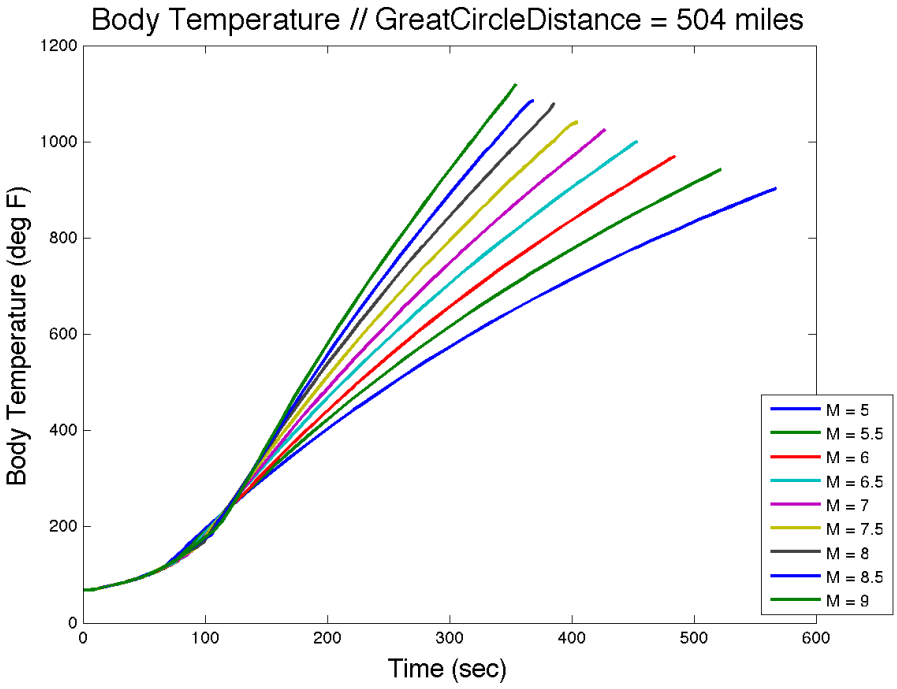

Body Heating Load // M = 8.0
Heating Load (BTU/ft^2)
Time (sec)
0
1000
2000
3000
4000
5000
6000
7000
8000
0
500
1000
1500
2000
2500
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance

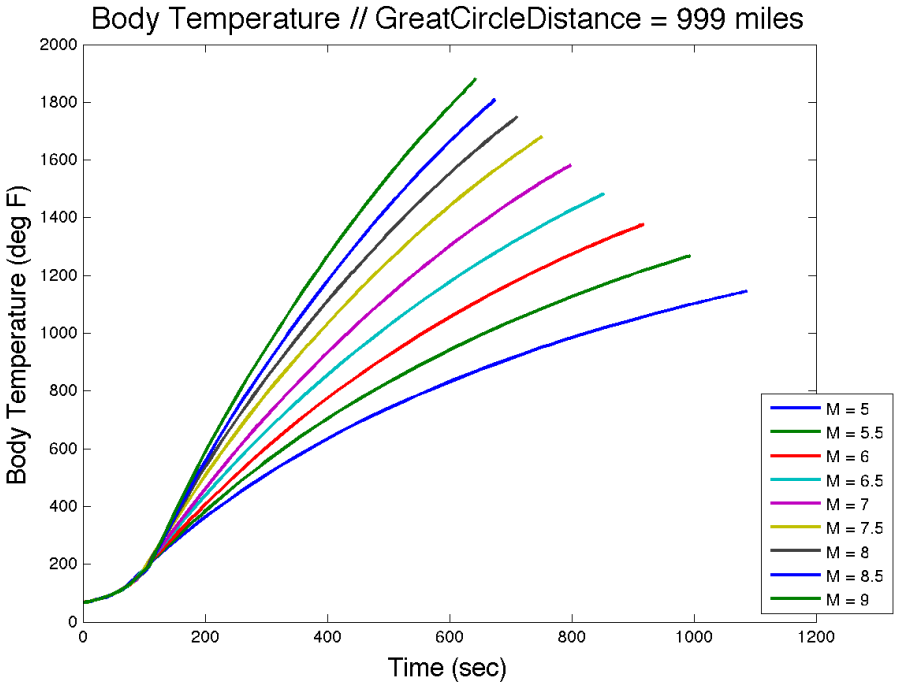

Body Heating Load // M = 8.5
Heating Load (BTU/ft^2)
Time (sec)
0
1000
2000
3000
4000
5000
6000
7000
8000
0
500
1000
1500
2000
2500
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance

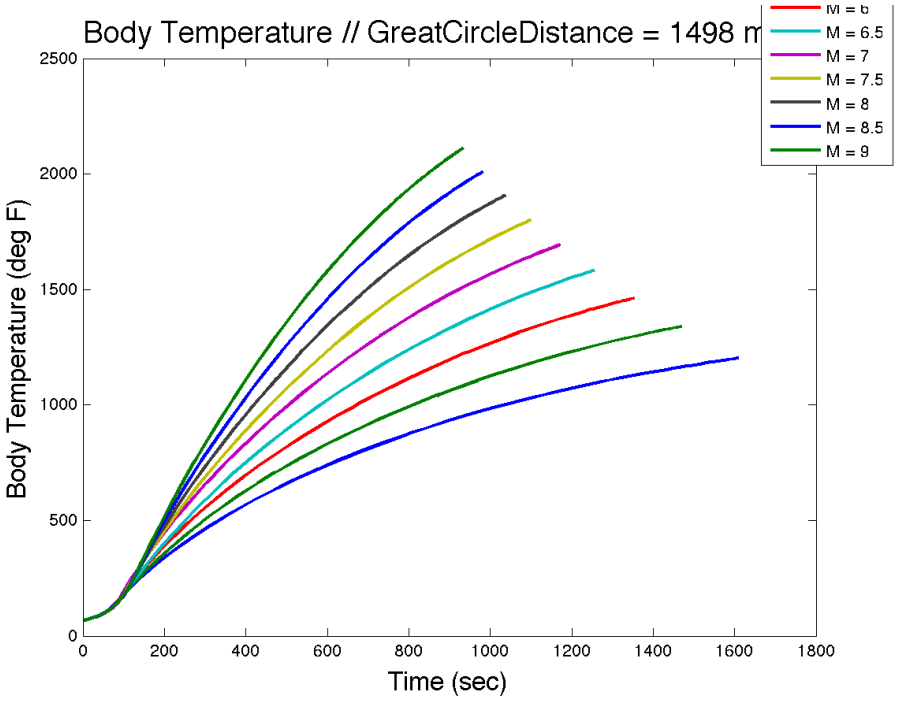


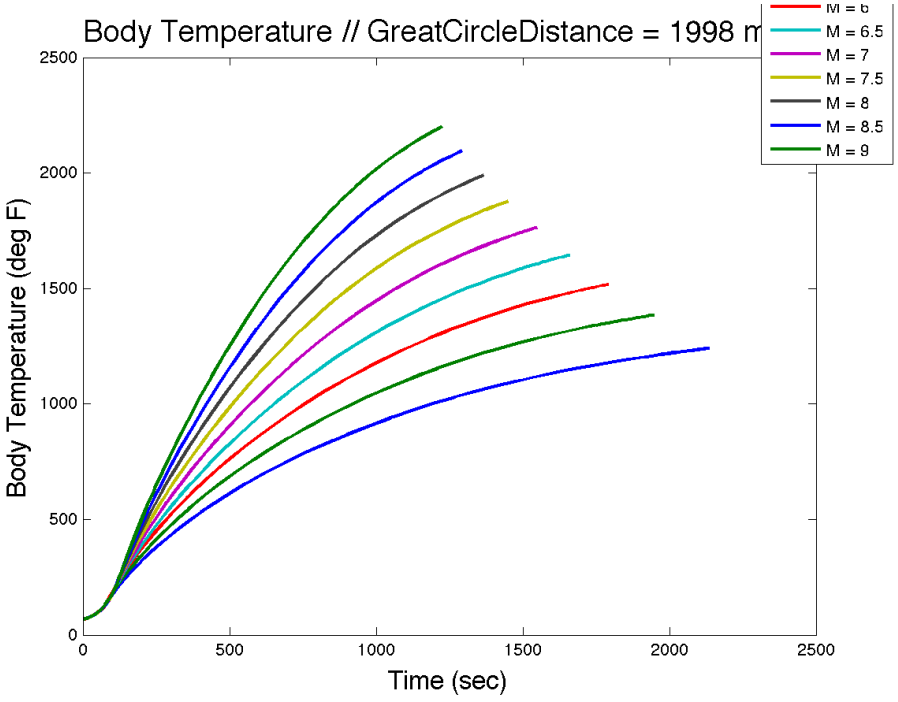
D.6.6 Body Temperature vs. Time.

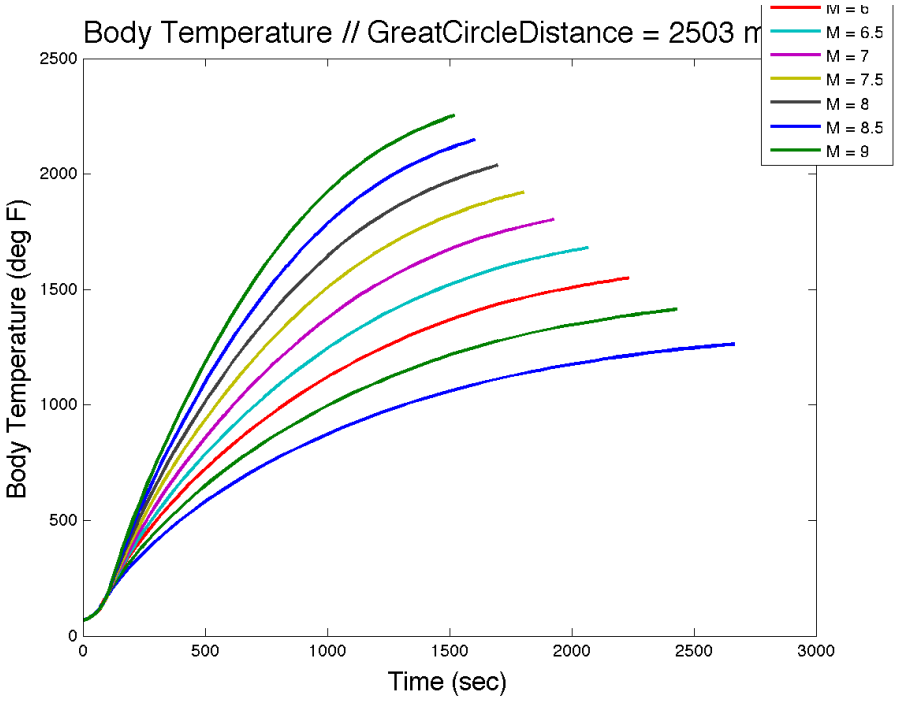


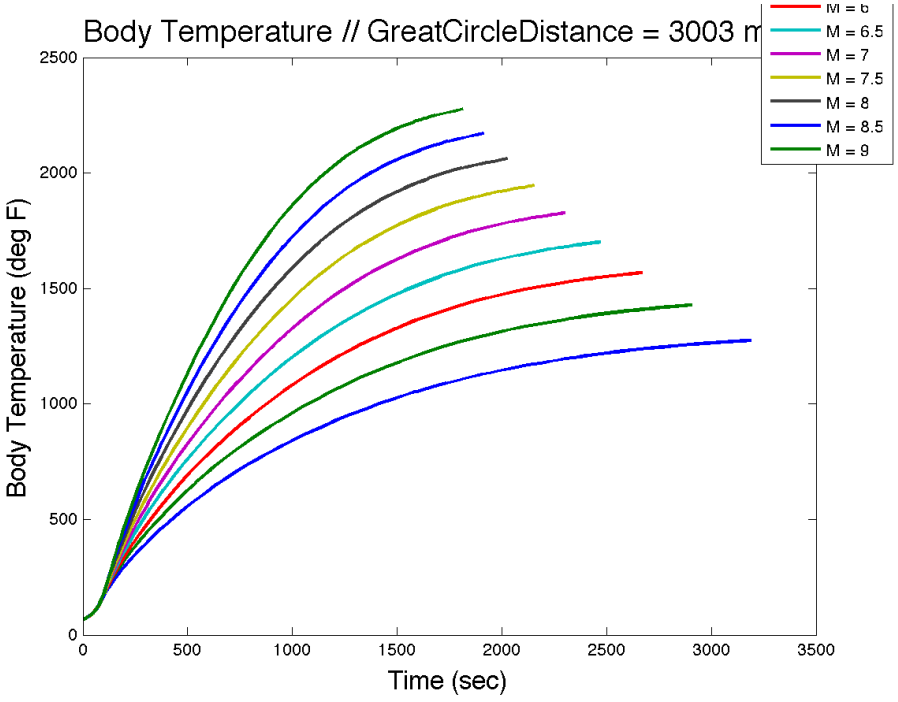

Body Temperature // GreatCircleDistance = 504 miles
Body Temperature (deg F)
Time (sec)
0
200
400
600
800
1000
1200
0
100
200
300
400
500
600
M = 5
M = 5.5
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9

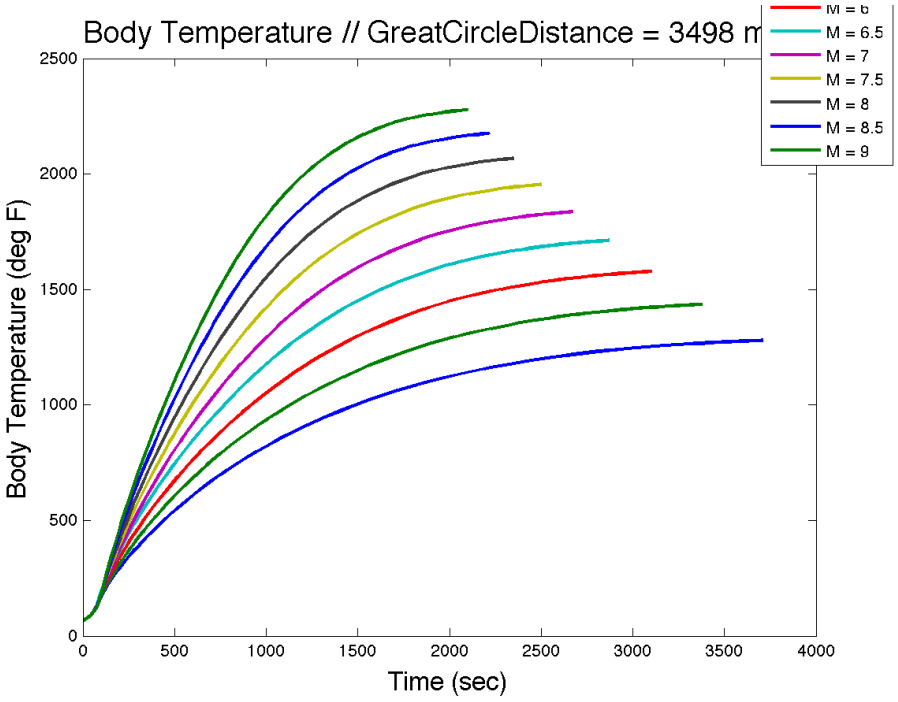

Body Temperature // GreatCircleDistance = 999 miles
Body Temperature (deg F)
Time (sec)
0
200
400
600
800
1000
1200
1400
1600
1800
2000
0
200
400
600
800
1000
1200
M = 5
M = 5.5
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9

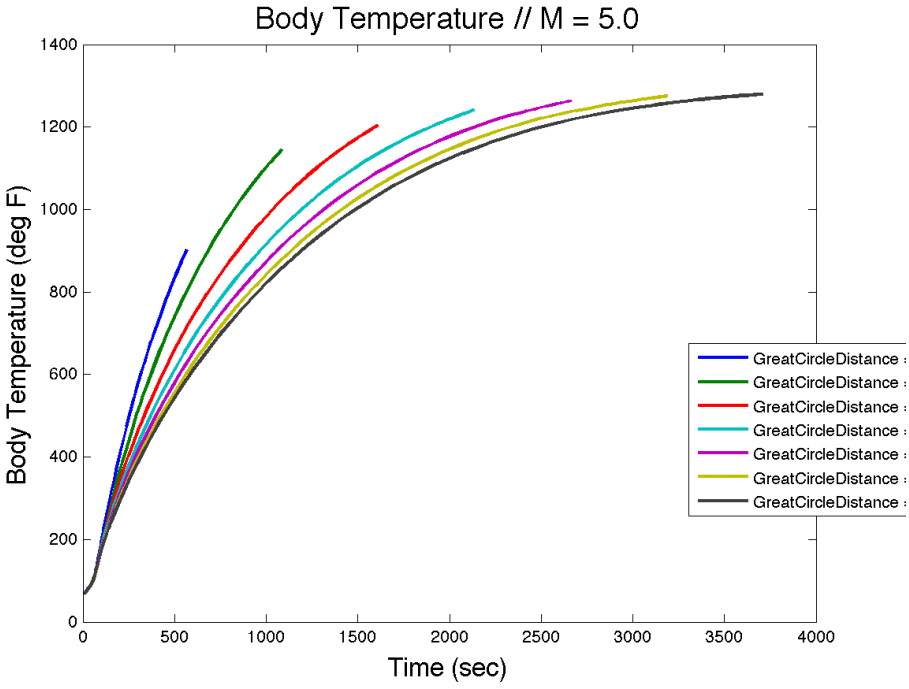

Body Temperature // GreatCircleDistance = 1498 m
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9
Body Temperature (deg F)
Time (sec)
0
500
1000
1500
2000
2500
0
200
400
600
800
1000
1200
1400
1600
1800

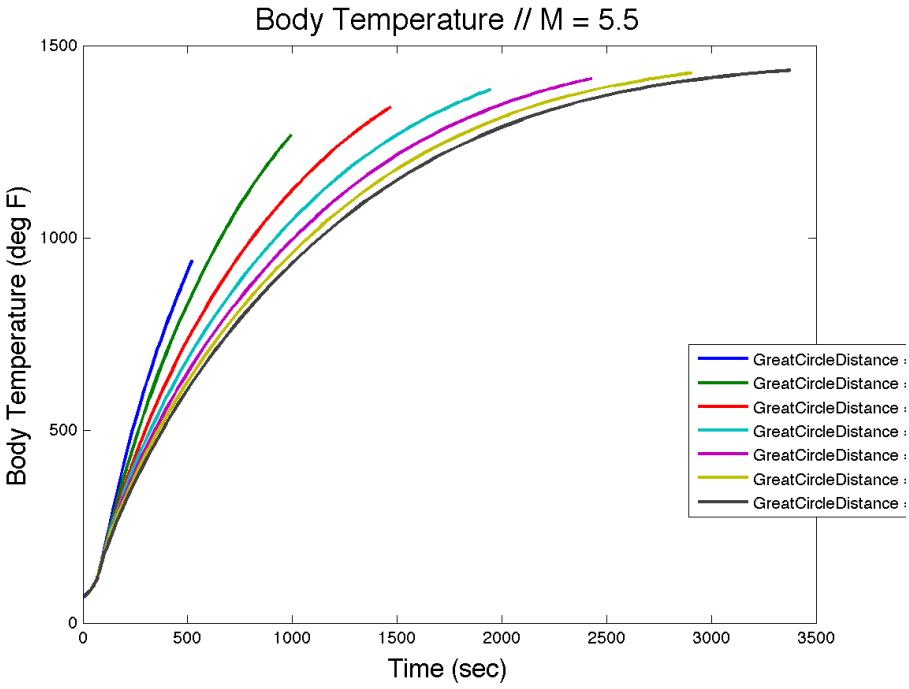

Body Temperature // GreatCircleDistance = 1998 m
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9
Body Temperature (deg F)
Time (sec)
0
500
1000
1500
2000
2500
0
500
1000
1500
2000
2500

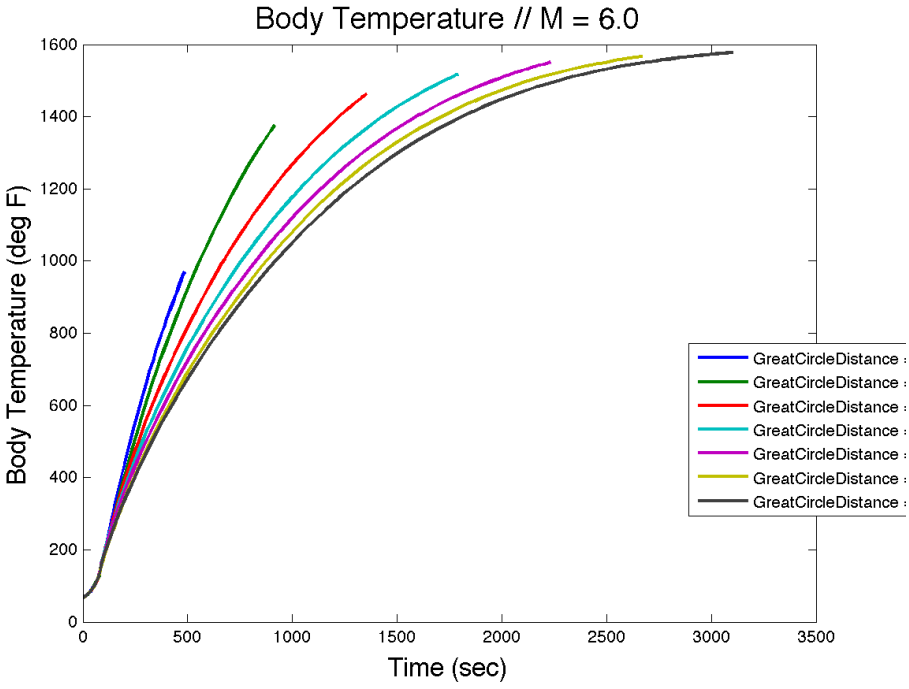

Body Temperature // GreatCircleDistance = 2503 m
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9
Body Temperature (deg F)
Time (sec)
0
500
1000
1500
2000
2500
3000

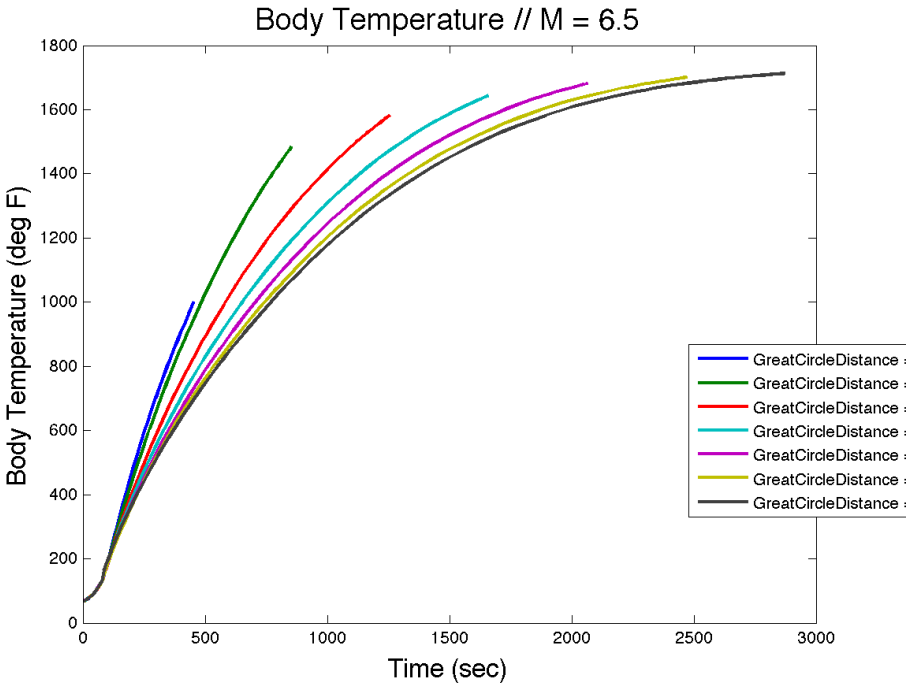

Body Temperature // GreatCircleDistance = 3003 m
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9
Body Temperature (deg F)
Time (sec)
0
500
1000
1500
2000
2500
3000
3500

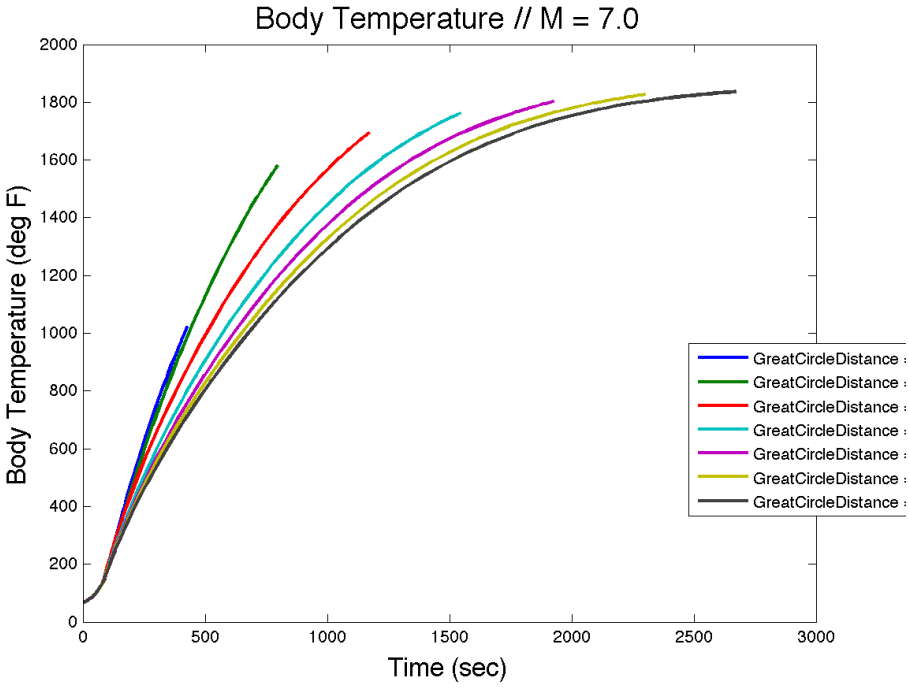

Body Temperature // GreatCircleDistance = 3498 m
Body Temperature (deg F)
Time (sec)
M = 6
M = 6.5
M = 7
M = 7.5
M = 8
M = 8.5
M = 9
0
500
1000
1500
2000
2500
3000
3500
4000

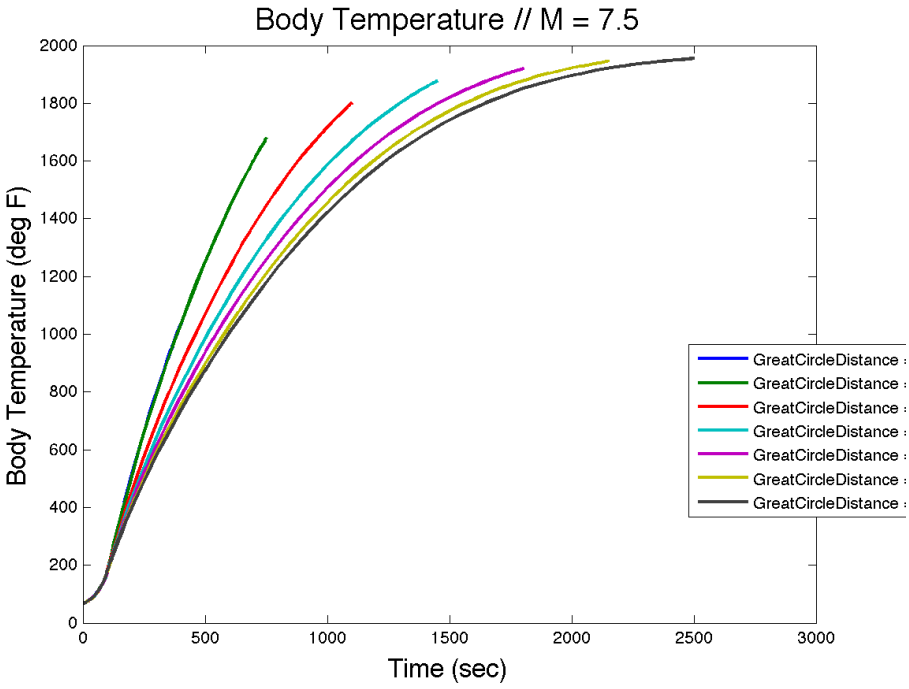

Body Temperature // M = 5.0
Body Temperature (deg F)
Time (sec)
GreatCircleDistance
0
200
400
600
800
1000
1200
1400
500
1500
2000
2500
3000
3500
4000

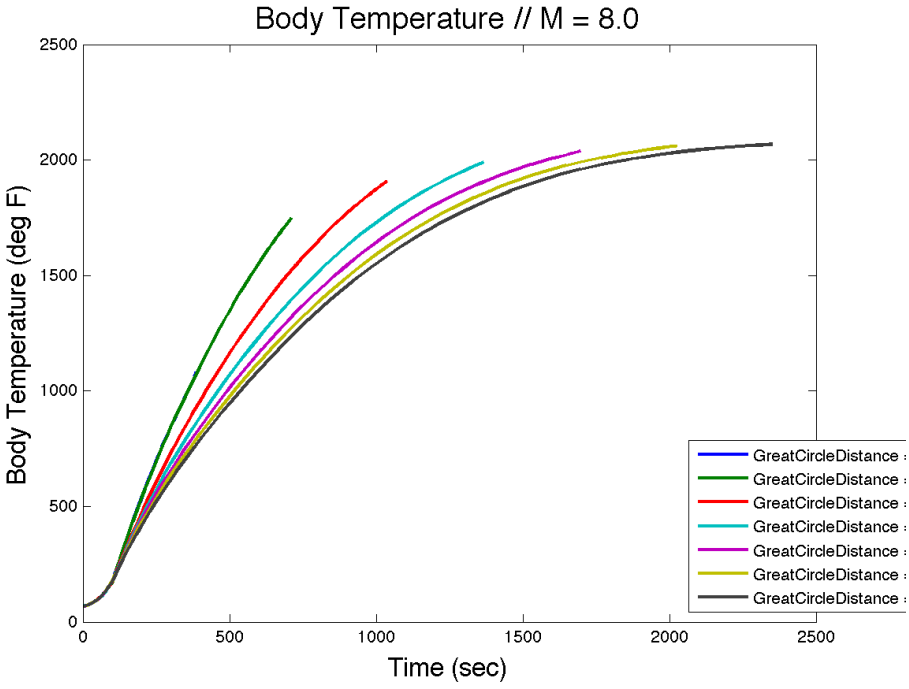

Body Temperature // M = 5.5
Body Temperature (deg F)
Time (sec)
0
500
1000
1500
2000
2500
3000
3500
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance

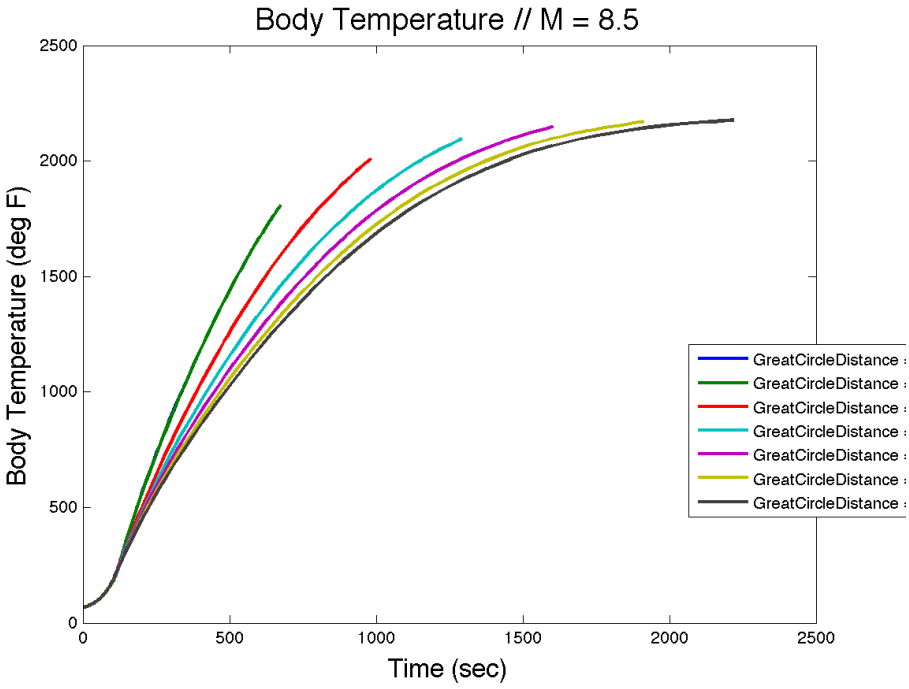

Body Temperature // M = 6.0
Body Temperature (deg F)
Time (sec)
0
200
400
600
800
1000
1200
1400
1600
500
1500
2000
2500
3000
3500
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance

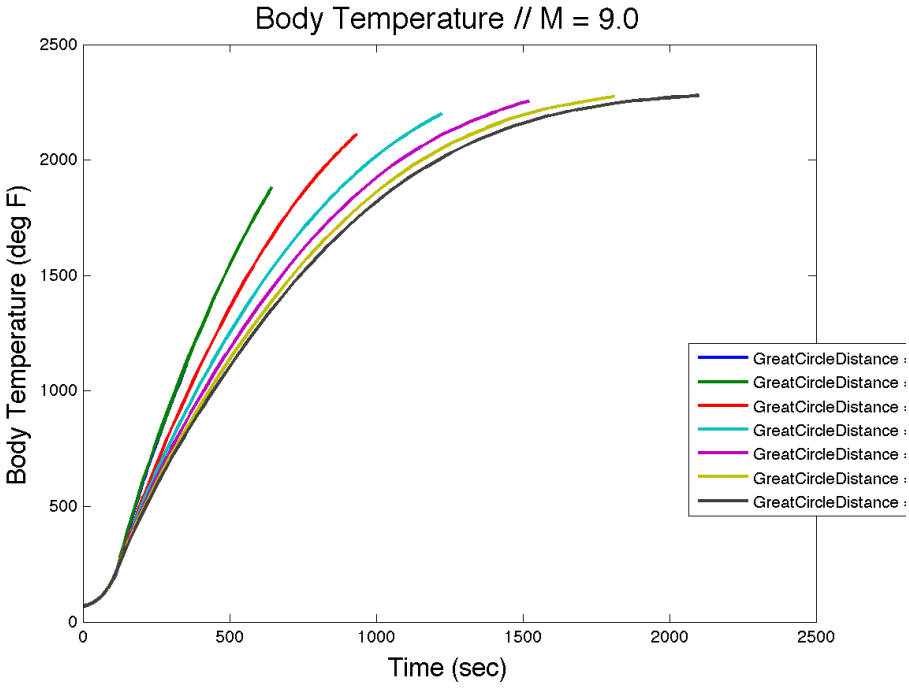

Body Temperature // M = 6.5
Body Temperature (deg F)
Time (sec)
0
200
400
600
800
1000
1200
1400
1600
1800
0
500
1000
1500
2000
2500
3000
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance


Body Temperature // M = 7.0
Body Temperature (deg F)
Time (sec)
0
200
400
600
800
1000
1200
1400
1600
1800
2000
0
500
1000
1500
2000
2500
3000
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance
GreatCircleDistance


Body Temperature // M = 7.5
Body Temperature (deg F)
Time (sec)
0
200
400
600
800
1000
1200
1400
1600
1800
2000
0
500
1000
1500
2000
2500
3000
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :


Body Temperature // M = 8.0
Body Temperature (deg F)
Time (sec)
0
500
1000
1500
2000
2500
0
500
1000
1500
2000
2500
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :


Body Temperature // M = 8.5
Body Temperature (deg F)
Time (sec)
0
500
1000
1500
2000
2500
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :


Body Temperature // M = 9.0
Body Temperature (deg F)
Time (sec)
0
500
1000
1500
2000
2500
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :
GreatCircleDistance :

Appendix E. Draft Conference Article

This appendix contains a draft conference article that was submitted to the AIAA Atmospheric Flight Mechanics Conference at SciTech 2015. The paper was not accepted. It will be updated and submitted as a journal article after this dissertation is completed.

Trajectory Optimization for Hypersonic Air-Breathing Vehicles Using Variable-Order Gauss-Radau Quadrature Collocation Methods

Tadeusz J. Masternak[1], David R. Jacques[2], Richard G. Cobb[3] and William P. Baker[4]
Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio, 45433, USA

In this paper, we consider determining optimal trajectories for a scramjet-based air-breathing hypersonic vehicle by developing an optimal control formulation and solving it using a variable-order Gauss-Radau quadrature collocation method with a nonlinear programming solver. The vehicle is assumed to be a reusable aircraft that has specified takeoff/landing locations, air-to-air refueling location constraints, and pre-determined no-fly zones. A common three degree-of-freedom dynamics model is implemented with aerodynamics and propulsion models adapted from the commonly used Generic Hypersonic Aerodynamic Model Example data set. Optimal trajectories are developed using several different performance metrics in the optimal control formulation–minimum time, minimum time with control penalty, and maximum range. Several scenarios are considered, building up to a nominal mission planning profile that includes: runway takeoff, climb to cruise altitude, aerial refueling, avoidance of a no-fly zone, descent from cruise altitude, and runway landing. The resulting analysis demonstrates that optimal trajectories that meet specified mission parameters and constraints can be quickly determined and used for mission and fleet planning. As a result of this work, a designer can develop optimal trajectories for a remote sensing mission for a hypersonic aircraft, including vehicle-driven constraints, geographic no-fly zones, and aerial refueling.

[1] Ph.D. Candidate, Department of Systems Engineering and Management, AIAA Senior Member, SMART Scholar.
[2] Associate Professor, Department of Systems Engineering and Management, AIAA Associate Fellow.
[3] Associate Professor, Department of Aeronautics and Astronautics, AIAA Associate Fellow.
[4] Associate Professor, Department of Mathematics and Statistics.

NOMENCLATURE

$\mathbf{U}$	Set of admissible controls
$(x)t$	x-coordinate of candidate state vector at time t, m
$(y)t$	y-coordinate of candidate state vector at time t, m
$(z)t$	z-coordinate of candidate state vector at time t, m
$2-D$	Two-dimensional
$3-D$	Three-dimensional
α	angle of attack, deg
$\mathbf{C}$	optimal control formulation constraints
$\mathbf{x}$	state vector
$\mathbf{x}^*$	optimal state vector
$\mathcal{L}$	Optimal control running cost (Lagrangian)
$\mathcal{M}$	Optimal control running cost (Mayer)
$\dot{m}_a$	air mass flow rate, kg/s
$\dot{m}_f$	propellant mass flow rate, kg/s
ϵ	convex combination coefficient
γ	flight path angle, deg
μ	standard gravitational parameter of earth, kg^3/s^2
ω	angular velocity of earth, rad
Φ	fuel-air ratio

ϕ latitude, deg

Φ_{st} stoichiometric fuel-air ratio

Ψ optimal control formulation boundary conditions

ψ heading angle, deg

ρ atmospheric density, kg/m^3

σ bank angle, deg

θ longitude, deg

φ fuel-air equivalence ratio

A Aerodynamic axial force

a speed of sound, m/s

A_0/A_C inlet capture area ratio

A_0/A_{C0} inlet capture ratio at zero alpha

$A_0/A_{C\alpha}$ inlet capture ratio coefficient based on alpha

A_{ic} inlet capture area, m^2

A_{ref} aerodynamic reference area, m^2

C_A force coefficient along velocity vector

C_D drag coefficient

C_L lift coefficient

C_N force coefficient orthogonal to velocity vector in lift-drag plane

C_{D_0} zero-lift drag coefficient

C_{L_0} zero-lift lift coefficient

$C_{L_{\alpha=0}}$ lift coefficient at zero angle of attack

C_{L_α} lift coefficient curve slope

D drag, N

d_n Minimum keep-away distance from nth no-fly zone center

g gravitational acceleration, m/s^2

g_0 gravitational acceleration at earth's surface, m/s^2

h vehicle height above earth surface, m

J Performance measure or objective functional or cost function

K induced drag coefficient

L lift, N

M Mach number

m mass, kg

N Aerodynamic normal force, N

q dynamic pressure, Pa

r radial distance from vehicle to earth center, m

r_e radius of (spherical) earth, m

T thrust, N

t time, sec

t_0 final time, sec

t_0 initial time, sec

V speed, m/s

x_n x-coordinate of center of nth no-fly zone, m

y_n y-coordinate of center of nth no-fly zone, m

z_n z-coordinate of center of nth no-fly zone, m

$\mathbf{u}$ control vector

$\mathbf{u}^*$ optimal control vector

I_{sp} specific impulse, sec

I. Introduction

In the last several years, hypersonic vehicles have again grabbed the interest of researchers and governments due to recent advancements in technology. The unique capabilities of a hypersonic air-breathing vehicle to provide a reusable and extremely fast means of transportation as well as significantly increasing the survivability of a military vehicle, has driven several near-term hypersonic-related technology efforts as well as conceptual designs for hypersonic air-breathing vehicles.

In addition to developing the materials, propulsion systems, and other hypersonic-enabling technologies, design and planning tools need to be developed as hypersonic technologies transition to operational vehicles. While current trajectory optimization techniques are well developed for subsonic and supersonic vehicles [1], they need to be expanded to incorporate the unique modeling for the aerodynamic, propulsion, and aerothermal aspects of a hypersonic vehicle.

This paper has several contributions. First, it develops a framework to quickly develop optimal trajectories for a hypersonic air-breathing vehicle based on existing system and experimental data, and uses an optimal control formulation, solved using a variable-order Gauss-Radau quadrature collocation method with a Nonlinear Programming (NLP) solver, that could include waypoints and no-fly zones. Second, this paper describes how to properly adapt and use a table lookup model in a gradient-based NLP solver. Although the Generic Hypersonic Aerodynamic Model Example (GHAME) table lookup data is continuous, the derivative of the data is not continuous, which is

usually incompatible for use with gradient-based NLP solvers. Using spline curve fits for interpolated values, the lookup tables can be used directly by the NLP when evaluating the gradients.

A. Overview

This paper consists of five sections. The first section provides an introduction to the problem and reviews the current literature. The second section describes the vehicle and dynamics models used in this paper. The next section describes the methodology to develop optimal trajectories. The fourth section provides example problems and corresponding results using the developed methodology. Finally, the fifth section summarizes the results and conclusions of the work.

B. Literature Review

Trajectory optimization techniques have steadily been developed since Robert Goddard posed the first aerospace optimal control problem in 1919 and its first significant use in the Gemini and Apollo space programs. [2] With the advances in optimal control and numerical computations, trajectory optimization techniques have advanced and been refined to the point that there are many dedicated trajectory optimization software packages, such as Optimal Trajectories by Implicit Simulation (OTIS), developed by the National Aeronautics and Space Administration (NASA) and Boeing, and Program to Optimize Simulated Trajectories (POST-II), developed by NASA and Lockheed-Martin. [1] Typical dedicated trajectory optimization software packages are generalized point mass, discrete parameter targeting and optimization programs. Since these program can break up the problem into multiple phases, they can analyze complex events during trajectories, such as powered and unpowered periods, staging events, and even vehicle reconfigurations or separation into several vehicles. [1]

While OTIS and POST-II are utilized throughout the aerospace industry, they both have significant limitations. POST-II uses direct shooting methods which can limit its applicability. While OTIS can be used with either direct shooting methods or low-order direct collocation methods, the software does not automatically check to see if all the specified constraints are met at the end of the optimization. [1] Other researchers have refined and applied direct shooting methods to hypersonic optimal trajectories, but retain limitations such as simple lookup tables for aerodynamic

forces or just model unpowered flight. [3] Other researchers have simplified the problem, such as assuming a constant airmass flow and limiting the number of independent variables. [4] Finally, other researchers have applied genetic algorithms to develop hypersonic optimal trajectories, albeit for simplified problems. [5]

A trajectory optimization approach, or a general optimal control software approach, that is computationally efficient and versatile, while based on a robust mathematical foundation, would provide significant advantages over many other existing trajectory optimization tools. [6] Several researchers such as Murillo [7], Shi [8], Wu [9] and Song [10] have used a direct collocation approach using a Gauss-Radau methodology to develop optimal trajectories for hypersonic vehicles, but with significant simplifications or fairly narrow conditions, such as looking only at unpowered reentry trajectories or just the cruise phase of hypersonic flight profile.

II. Modeling

Although hypersonic vehicle models have been developed using many different approaches, such as Bolender and Doman [11] and also Frendreis, Skujins, and Cesnik [12], the GHAME model is used in this paper since it is well established, available in the open literature, and in use for over twenty years. This GHAME model has frequently been used in hypersonic vehicle trajectory optimization since it includes the aerodynamic, propulsion, and aerothermodynamic models for a hypersonic air-breathing Single Stage To Orbit (SSTO) vehicle, as documented by Bowers [13] and White [14]. Murillo used a GHAME data set to develop minimum fuel ascent trajectories for SSTO hypersonic air-breathing vehicles, using both open-loop and closed-loop guidance approaches with finite difference methods. [7] Araki used the GHAME data set to investigate reentry dynamics and handling qualities of hypersonic vehicles using perturbation techniques. [15]

The GHAME model simulates a SSTO Turbine Based Combined Cycle (TBCC) vehicle that takes off horizontally, accelerates to near-hypersonic velocities using a turbine engine, transitions to a scramjet for hypersonic velocities, cruises at hypersonic velocities or accelerates to near orbital velocities (Mach 24), and then returns to earth via a horizontal landing. [13] While this paper is only going to consider operations at lower hypersonic velocities (up to Mach 8) for the mission

considered, this model will work well within this flight regime. The model consists of a hypersonic vehicle traveling over a spherical, rotating earth, and uses derived equations of motion and empirical data for both the aerodynamic force and the engine thrust models.

The model assumes the vehicle is a point mass so there are no moment terms in the equation of motion. The model also uses empirically-based tabular data for aerodynamic and propulsive forces to compute equations of motion. [16] When implemented in MATLAB®, these data sets are interpolated using spline or cubic interpolation methods to provide smooth data points when evaluating the dynamic constraints (equations of motion).

A three Degree of Freedom (3-DOF) model is used for this research since this tool is intended to do responsive trajectory optimization analysis for mission planners and system engineers.

A three Degree of Freedom (3-DOF) model is used for this research since it provides sufficient fidelity to meet research objectives, allowing for responsive trajectory optimization analysis. This approach is common in controls and trajectory research and is commonly called "inertialess" control. [17]

A. Vehicle Modeling

For this paper, a remote sensing mission is used. A nominal mission planning profile is considered and includes: runway takeoff, climb to cruise altitude, aerial refueling, avoidance of no-fly zones, descent from cruise altitude, and runway landing.

To perform this mission, a hypersonic air-breathing vehicle is considered. As stated previously, the modeling of a hypersonic vehicle is different from a subsonic or supersonic vehicles because the hypersonic vehicle models need to be enhanced to incorporate the unique modeling of aerodynamic, propulsion, and aerothermodynamic aspects.

The GHAME hypersonic vehicle model used in this paper is taken from Bowers [13] and White. [14] GHAME is a hypothetical aircraft to provide simulation models for design activities, such as trajectory optimization. It was developed to provide aerodynamic, aerothermodynamic and propulsion data representative of a SSTO-class air-breathing hypersonic vehicle using hydrogen fuel. The shape of the vehicle was based on a composite of simple geometric shapes that resemble an

aircraft. Table 1 lists the aerodynamic and propulsion reference data for the GHAME vehicle.

Table 1 GHAME reference data

GHAME Attribute	Values
aerodynamic reference area, A_{ref}	$557\,m^2$
length	$71\,m$
reference chord	$23\,m^2$
reference span	$24\,m$
take-off gross mass, m_{max}	$136078\,kg$
zero fuel mass, m_{min}	$54431\,kg$
inlet capture area, A_{ic}	$28\,m^2$
minimum fuel flow rate, $\dot{m}_{f_{\min}}$	$18\,kg/s$
maximum fuel flow rate, $\dot{m}_{f_{\max}}$	$163\,kg/s$

For this research, the aerodynamic forces and engine thrust models are adapted from the GHAME data sets, which are based on empirical data from previous operational systems and wind tunnel testing. [7] The lift, drag, and propulsion forces are computed from force and thrust coefficients in GHAME data sets, which vary based on vehicle flight conditions. For this paper, coefficients will vary with Mach number from Mach 0.4 to Mach 8 (GHAME data goes to Mach 24), angle of attack $-3^\circ \leq \alpha \leq 21^\circ$, and fuel-air equivalence ratio (φ) from 0 to 2.

The atmospheric model used in this paper is a MATLAB function for standard atmosphere function based on the 1976 Standard Atmosphere. [18] For a given altitude, it returns several values including atmospheric density and speed of sound. The model is limited to altitudes up to approximately 53 miles (86km) but that is much higher than the expected operational altitudes of the vehicle in this paper.

1. Aerodynamics Model

The lift force equation in the aerodynamics model for this vehicle is a standard linear equation for the lift coefficient with A_{ref} given in the GHAME data set and modified by Zipfel[5], defined as:

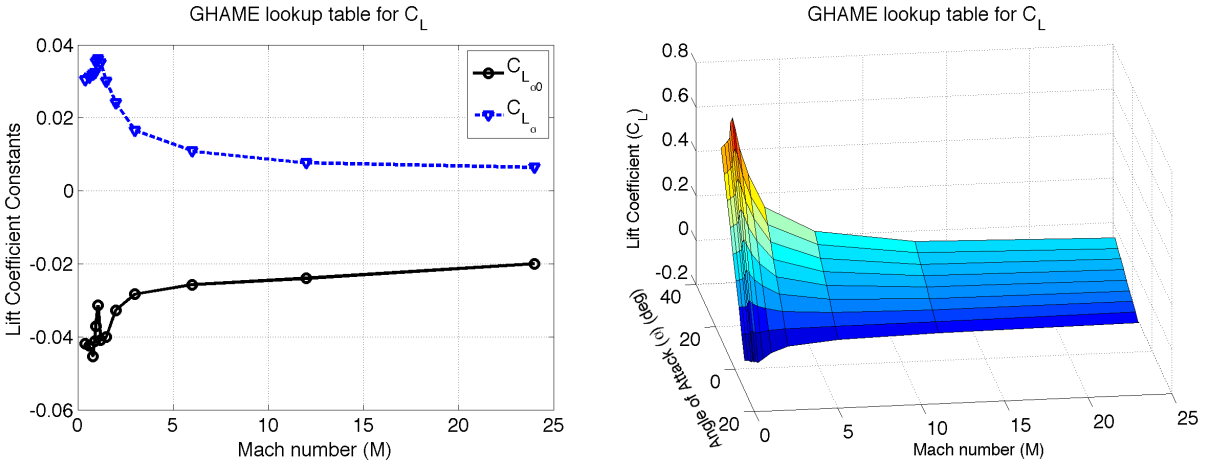
[5] The lift coefficient formulation used here is given by Zipfel [19], simplifying the GHAME lift coefficient equation by integrating the elevator deflection and pitch rate dependencies into the remaining terms.

$$L = C_L q A_{ref} \tag{1}$$

$$q = \frac{\rho V^2}{2} \tag{2}$$

$$C_L = C_{L_{\alpha_0}} + C_{L_\alpha}\alpha \tag{3}$$

The terms $C_{L_{\alpha_0}}$ and C_{L_α} are functions of Mach number and are defined in the GHAME data tables. In Figure 1 below, both the 3-D plot of C_L and 2-D plot for C_L equation terms ($C_{L_{\alpha_0}}$ and C_{L_α}) are shown.



(a) 2-D Equation terms (slope and intercept coefficients)

(b) 3-D Curve as a function of M and *alpha*

Fig. 1 GHAME Lift Coefficient Curves expressed as either (a) a linear function with slope and intercept coefficients as a function of M and (b) as a 3-D function of M and *alpha*

While C_D and C_L are vehicle frame force coefficients, it is useful to define the resulting stability frame forces, C_A and C_N. These force coefficients are useful to visualize vehicle forces as well as factors used in additional path constraints.

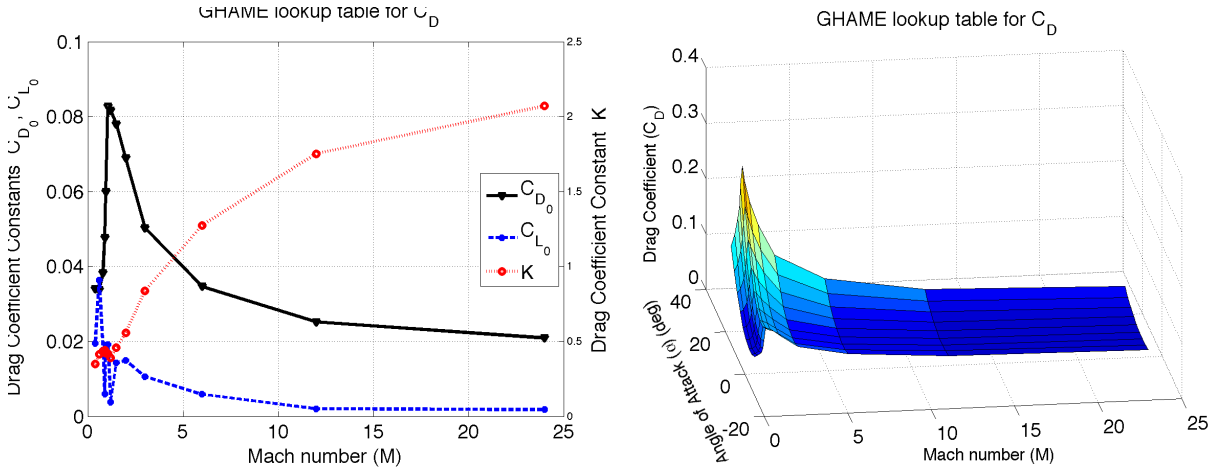
$$\begin{bmatrix} C_A \\ C_N \end{bmatrix} = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} C_D \\ C_L \end{bmatrix} \tag{4}$$

The drag force equation in the aerodynamics model for this vehicle is a standard offset parabolic drag polar equation for the drag coefficient[6] with A_{ref} given in the GHAME data set, defined as:

$$D = C_D q A_{ref} \tag{5}$$

$$C_D = C_{D_0} + k(C_L - C_{L_0})^2 \tag{6}$$

The terms C_{D_0}, k, and C_{L_0} are functions of Mach number and are defined in the GHAME data tables. In Figure 2 below, both the 3-D plot of C_D and 2-D plot for C_D equation terms (C_{D_0}, k, and C_{L_0}) are shown.



(a) 2-D Equation terms (slope and intercept coefficients)

(b) 3-D Curve as a function of M and *alpha*

Fig. 2 GHAME Drag Coefficient Curves expressed as either (a) a linear function with slope and intercept coefficients as a function of M and (b) as a 3-D function of M and *alpha*

2. Propulsion Model

The GHAME thrust equation in the propulsion model for this vehicle is a standard thrust equation, defined as:

[6] The drag coefficient formulation used here is given by Zipfel [19], modifying the GHAME drag coefficient equation by transforming it into a parabolic equation.

$$T = g_0 \dot{m}_f I_{sp} \tag{7}$$

The specific impulse term, I_{sp}, is provided in the GHAME data set and is also a function of Mach number and fuel-air equivalence ratio, where Φ_{st} is the fuel stoichiometric fuel-air ratio[7] and φ is defined as:

$$\varphi = \frac{\Phi}{\Phi_{st}} = \frac{\frac{\dot{m}_f}{\dot{m}_a}}{\Phi_{st}} = \frac{\frac{\dot{m}_f}{\rho V A_0 / A_c A_{ic}}}{\Phi_{st}} \tag{8}$$

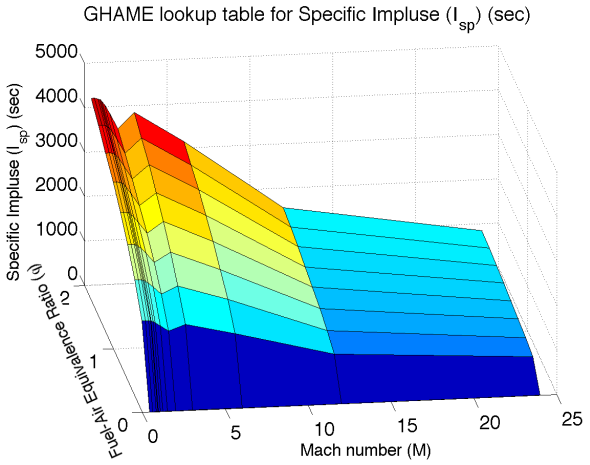
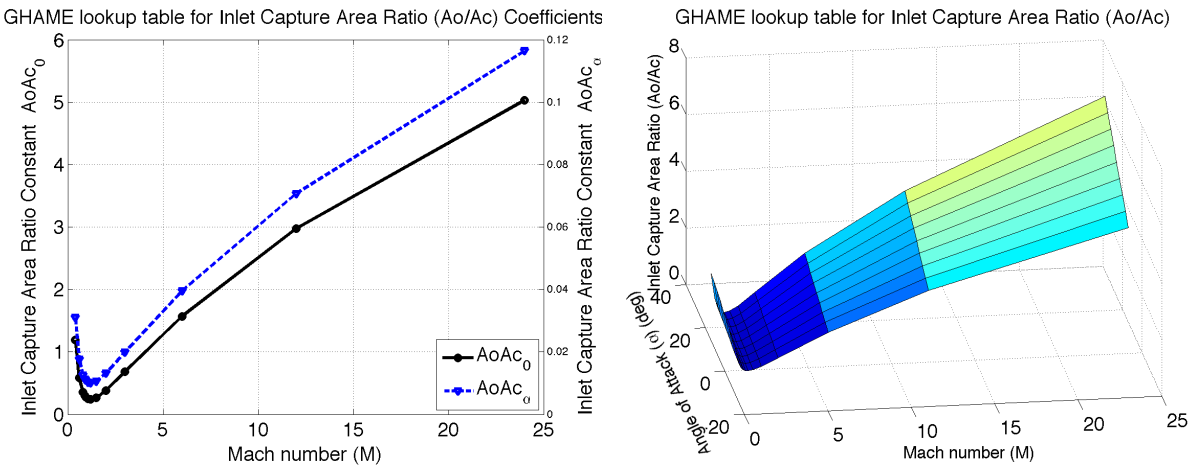


Fig. 3 GHAME Specific Impulse (I_{sp})

A_0/A_c is the inlet area capture ratio and is a function of Mach number and α. In the referenced GHAME source [14], A_0/A_c is provided as a lookup table based on Mach number and α, but it can be reduced to a linear equation that varies with α where each of the coefficients is only based on Mach number.

$$A_0/A_c = A_0/A_{c_0} + A_0/A_{c_\alpha}\alpha \tag{9}$$

[7] For hydrogen fuel, $\Phi_{st} = .0292$



(a) 2-D Equation terms (slope and intercept coefficients)

(b) 3-D Curve as a function of M and *alpha*

Fig. 4 GHAME Inlet Area Capture Ratio Curves expressed as either (a) a linear function with slope and intercept coefficients as a function of M and (b) as a 3-D function of M and *alpha*

B. Reference Frames

The coordinate systems used in this paper are standard and shown in many sources, such as Vinh. [20] Figure 5 summarizes the rotations and translations between the reference frames.

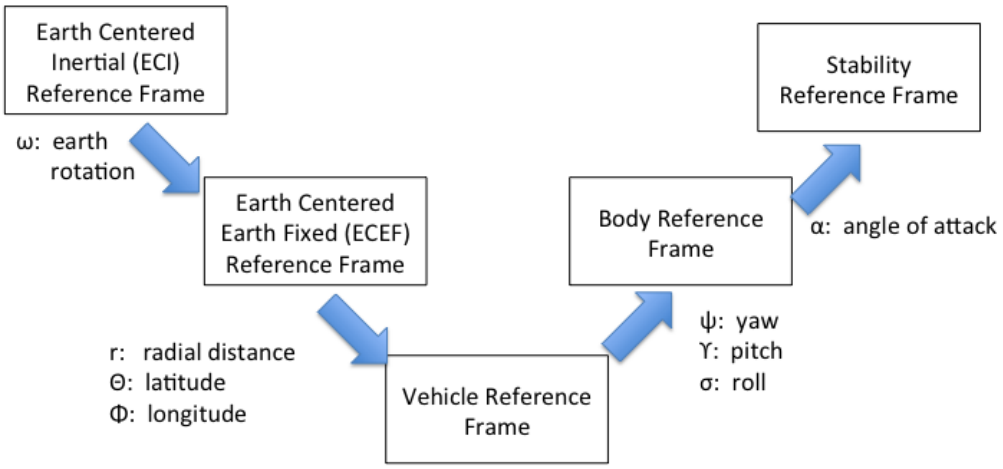


Fig. 5 Reference Frames

C. Dynamics Modeling

This paper uses the standard 3-DOF equations of motion as derived by Vinh for a point mass. [20] Symmetric flight is assumed, thus sideslip angle β is zero and not included in any equations.

The following set of equations are the equations of state (or dynamic constraints). The first three equations are kinematic equations, the second three equations are force equations and the last equation is the mass flow rate equation. As previously stated, these equations of motion assume flight over a spherical, rotating earth and coordinated turns with no sideslip angle. These equations are also considered "inertialess" since they do not include command delays, from using control surface deflections, and body moments. [17]

$$\frac{dr}{dt} = V \sin\gamma \tag{10}$$

$$\frac{d\theta}{dt} = \frac{V \cos\gamma \cos\psi}{r \cos\phi} \tag{11}$$

$$\frac{d\phi}{dt} = \frac{V \cos\gamma \sin\psi}{r} \tag{12}$$

$$\frac{dV}{dt} = \frac{F_T}{m} - g \sin\gamma + \omega^2 r \cos\phi \left(\sin\gamma \cos\phi - \cos\gamma \sin\psi \sin\phi\right) \tag{13}$$

$$\frac{d\gamma}{dt} = \frac{F_N \cos\sigma}{mV} - \cos\gamma \left(\frac{V}{r} - \frac{g}{V}\right) + 2\omega \cos\psi \cos\phi + \frac{\omega^2 r \cos\phi}{V} \left(\cos\gamma \cos\phi + \sin\gamma \sin\psi \sin\phi\right) \tag{14}$$

$$\frac{d\psi}{dt} = \frac{F_N \sin\sigma}{mV \cos\gamma} - \frac{V}{r} \cos\gamma \cos\psi \tan\phi + 2\omega \left(\tan\gamma \sin\psi \cos\phi - \sin\phi\right) - \frac{\omega^2 r}{V \cos\gamma} \cos\psi \sin\phi \cos\phi \tag{15}$$

$$\frac{dm}{dt} = -\frac{T}{I_{sp} g_0} \tag{16}$$

For the equations of motion in Equations 10 thru 16, the state and control vector are given as:

$$\mathbf{x} = \begin{bmatrix} h(t) \\ \theta(t) \\ \phi(t) \\ V(t) \\ \gamma(t) \\ \psi(t) \\ m(t) \end{bmatrix} \tag{17}$$

$$\mathbf{u} = \begin{bmatrix} \alpha(t) \\ \sigma(t) \\ \dot{m}_f(t) \end{bmatrix} \tag{18}$$

D. Path Constraints

For this paper, there are three default path constraints: M, φ, and q. In the GHAME data set, φ values are given between 0 and 2, which constrains the admissible set of solutions. Vehicles also have structural limits, which can be directly related to the dynamic pressure, q. A representative limit for q for this class of vehicle is used as a path constraint. [21]

Many mission scenarios of interest for this class of vehicle includes potential flight profiles near restricted airspace where the vehicle is constrained to stay out of a no-fly zone. This restriction is incorporated as an additional path constraint. Equation 19 is used to calculate the distance from every no-fly center to each element in a candidate state vector. This distance calculation is used with the user-defined no-fly size as a path constraint.

$$\sqrt{(x(t) - x_n)^2 + (x(t) - y_n)^2 + (x(t) - z_n)^2} \geq d_n \tag{19}$$

III. Methodology

A. Optimal Control Problem Formulation

Optimal control theory deals with the problem of finding a set of control inputs for a nonlinear dynamic system which result in system states that minimize (or maximize) a specified performance measure while meeting specified system dynamics and staying within specified constraints and boundary conditions. A more formal definition is given as:

> Find an admissible control $\mathbf{u}^* \in U$ that causes the system state dynamics $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t)$ to follow an admissible trajectory $\mathbf{x}^* \in X$ that minimizes the performance measure J. $\mathbf{x}^*$ is called the optimal trajectory and $\mathbf{u}^*$ is called the optimal control. [22]

Specifically, an optimal control problem can be formulated as:

$$\text{Minimize: } J = \mathcal{M}\left(\mathbf{x}_0, \mathbf{x}_f, t_0, t_f\right) + \int_{t_o}^{t_f} \mathcal{L}\left(\mathbf{x}(t), \mathbf{u}(t), t\right) dt \tag{20}$$

Subject to:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t)$$

$$\Psi\left(\mathbf{x}(t_0), \mathbf{x}(t_f), t_0, t_f\right) = 0 \tag{21}$$

$$\mathbf{C}\left(\mathbf{x}(t), \mathbf{u}(t), t\right) \leq 0$$

In these equations, $\mathcal{M}$ represents the terminal cost for the endpoints (also called the Mayer term) [22], while $\mathcal{L}$ represents the running cost (also called the Lagrangian term). In addition, $\mathbf{x}$ represents the state, $\dot{\mathbf{x}}$ the state dynamics, $\mathbf{u}$ the control, Ψ the boundary conditions, and $\mathbf{C}$ the path constraints.

This is the formulation for a single-phase optimal control problem, where the constraints and dynamics are consistent across the admissible solution space. For control problems that have dynamics and/or controls that have different definitions across the admissible solution space, the optimal control problem will have to be broken down into "smaller" optimal control problems that still span the entire problem being considered. This division of the optimal control problem is called phasing.

In this paper, three common types of optimal control problems will be considered–minimum time. minimum time with control penalty, and maximum range. Each of these problems are common design and mission planning scenarios.

B. Minimum Time Problem

Minimum time problems are one of the simplest applications of the general optimal control problem. This formulation is useful for computing the fastest possible time to transit between two fixed points. For this research, this formulation will be used to determine the fastest possible time to execute a given mission profile, with prescribed mission parameters and constraints. The minimum time formulation of the cost functional is:

$$\min_{u^* \in U} J = \int_{t_o}^{t_f} dt = t_f - t_0 \tag{22}$$

C. Minimum Time with Control Penalty Problem

Minimum time with control penalty problems are a variation to minimum time problems. In addition to the minimum time contribution to the objective function, there is also a component for the control usage. The term ϵ is a convex combination coefficient to give the user control on the relative weighting of the time and control contributions.

$$\min_{u^* \in U} J = \epsilon(t_f - t_0) + (1 - \epsilon) \int_{t_o}^{t_f} \mathbf{u}^T(t)\, \mathbf{u}(t)\, dt \tag{23}$$

D. Maximum Range Problem

Maximum range problems are another simple applications of the general optimal control problem. This formulation is useful for computing the maximum range of the vehicle for a given fuel load. The maximum range formulation is:

$$\min_{u^* \in U} J = -\int_{t_o}^{t_f} V dt \tag{24}$$

V is the vehicle speed, given by $\mathbf{V} = \|\dot{\mathbf{x}}\| = \|\mathbf{f}\|$

E. Pseudospectral Method

Pseudospectral (PS) techniques are a class of numerical methods developed in part to solve partial differential equations. They have been increasing in popularity as tools to solve increasingly complex optimal control problems. [23] These methods emerged from the development of spectral methods in the 1960s and 1970. Spectral methods are themselves a class of Method of Weighted Residuals (MWR) techniques, which are used in applied mathematics and scientific computing to numerically solve ordinary/partial differential equations and eigenvalue problems involving differential equations. Since spectral methods use basis functions (called trial functions) that are globally smooth over the entire problem interval, spectral method errors decay exponentially as the number of approximation nodes increases, which is significantly faster than the polynomial rates for Finite Difference and Finite Element methods, which have errors that decay no faster than quadratically. [24]

In this research, the MATLAB-based Variable-Order Gaussian Quadrature Collocation Method software called General Pseudospectral Optimal Control Software (GPOPS-II) is used to solve for optimal trajectories. [23] It iuses a PS technique that is a direct method and uses orthogonal trial functions and collocation points (i.e. direct orthogonal collocation). This approach approximates the states and controls using a basis of Lagrange polynomials and collocates them with the state equation dynamics at roots of Legendre polynomials (i.e. Legendre-Gauss-Radau (LGR) points). Since Legendre polynomials are orthogonal, they can be easily integrated or differentiated, using numerical methods, to determine objective function integrals or the state dynamics. The continuous-time optimal control problem is then transcribed to a finite-dimensional NLP and the NLP is solved using off-the-shelf NLP solvers to satisfy the specified NLP error tolerance. GPOPS-II also contains an adaptive mesh refinement method that determines the number of mesh intervals and the degree of the approximating polynomial for each mesh interval in order to satisfy the specified mesh error tolerance. This approach solves the optimal control problem using a given mesh (number of collocation points in each (sub)interval of the mesh). Once the NLP solver returns an optimal solution, there is a check to see if the mesh tolerance error is met. [23] If not, the mesh is refined by adding additional intervals and/or collocation points and then the NLP solver is invoked again to

find the optimal solution for this mesh. The GPOPS-II operational flow is shown in Figure 6. [23]

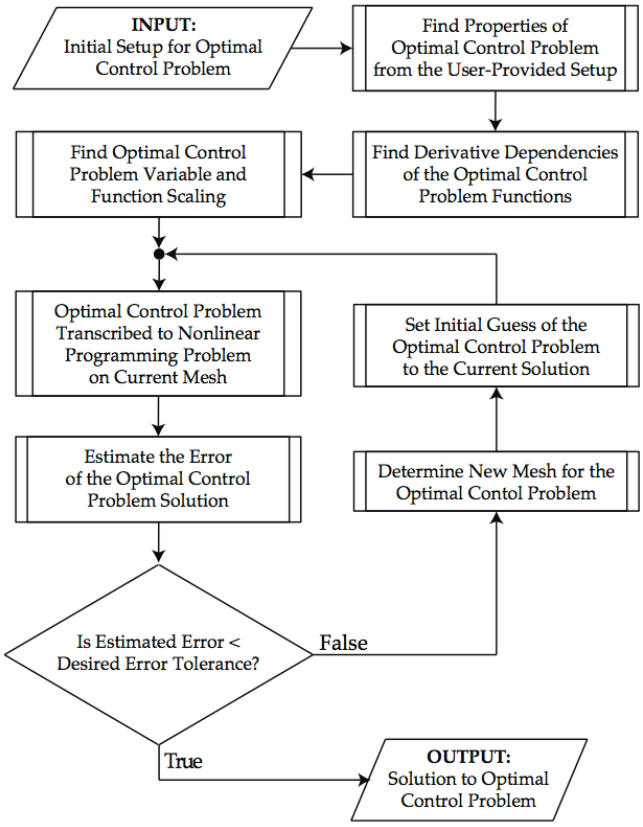


Fig. 6 GPOPS Operational Flowchart [23]

F. GHAME model implementation

As previously described, the GHAME data used in the paper is a set of lookup tables for aerodynamic and propulsion coefficients with Mach number (and controls) as the independent variable for each table; while φ is an additional variable for the I_{sp} (propulsion) lookup table, it is calculated from A_0/A_c, which is a function of Mach number. Each of these tables is based on empirical and analytical data, with low Mach number data points closer together than data points at larger Mach numbers. Since collocation methods use a mesh of independent variables that are not specified a priori, the framework has to interpolate between data points when using the lookup tables. Since the selected NLP solver (IPOPT (Interior Point OPTimizer)) requires objective and constraint op-

timization functions that are at least once continuously differentiable ($\mathbb{C}^1$) and preferably twice continuously differentiable ($\mathbb{C}^2$) [25], the lookup table interpolation method is limited to approaches that result in interpolated data that is at least $\mathbb{C}^1$.[8] In this paper, the MATLAB function griddedInterpolant[9] is used to interpolate lookup table data, using the spline interpolation method resulting in interpolated data that is a cubic interpolation using not-a-knot end conditions.[10]

IV. Results

This section provides the results of running the hypersonic trajectory optimization model against representative scenarios for different optimal control formulations.

The scenarios in this paper were run using in Matlab® version 2014a on a Mac Pro (MacPro5,1) desktop computer (24GB of memory, 3.2GHz quad-core Intel® Xeon W3530 processor) running OS X 10.9.2. GPOPS-II version 1.0 (hp-adaptive Radau Pseudospectral method) was run within Matlab®. Unless stated otherwise, the following GPOPS-II settings were used to run the scenarios in this paper.

Table 2 GPOPS-II settings

GPOPS-II option	Setting
setup.nlp.solver	IPOPT (v3.11.0)
setup.nlp.options.ipopt.linear_solver	ma57
setup.derivatives.supplier	sparseCD (sparse Center Difference)
setup.derivatives.derivativelevel	second
setup.derivatives.dependencies	sparseNaN (sparse Not a Number)
item setup.method	RPMIntegration
setup.nlp.options.tolerance	1e-6
setup.nlp.options.maxiterations	1000
setup.mesh.method	hp1
setup.mesh.tolerance	1e-3

[8] Both the objective and constraint functions can be "linear or nonlinear and convex or non-convex (but should be twice continuously differentiable)". [25]
[9] For I_{sp} interpolation, MATLAB function ndgrid is also used
[10] http://www.mathworks.com/help/matlab/ref/griddedinterpolant-class.html

The following results are given for the optimal control formulations described above. All the solutions provided met both the NLP and the mesh error tolerances specified in Table 2. Unless otherwise specified, the results for each scenario are presented in 2 sets of 4 figures each, with some subfigures containing two figures. In the first set, the results shown are (clockwise from the upper left): optimal trajectory plotted on a Mercator projection, optimal controls, and optimal states (2 subfigures). In the second set, the results shown are (clockwise from the upper left): 2-D flight profile, path constraints and flow rates, mesh interval and collocation point profile, and body/stability frame forces profile. Most of the plots have time as the independent variable, unless otherwise shown in the figures. Unless otherwise stated, the only path constraints included in each scenario are for q_{max}, M_{max}, and φ_{max}.

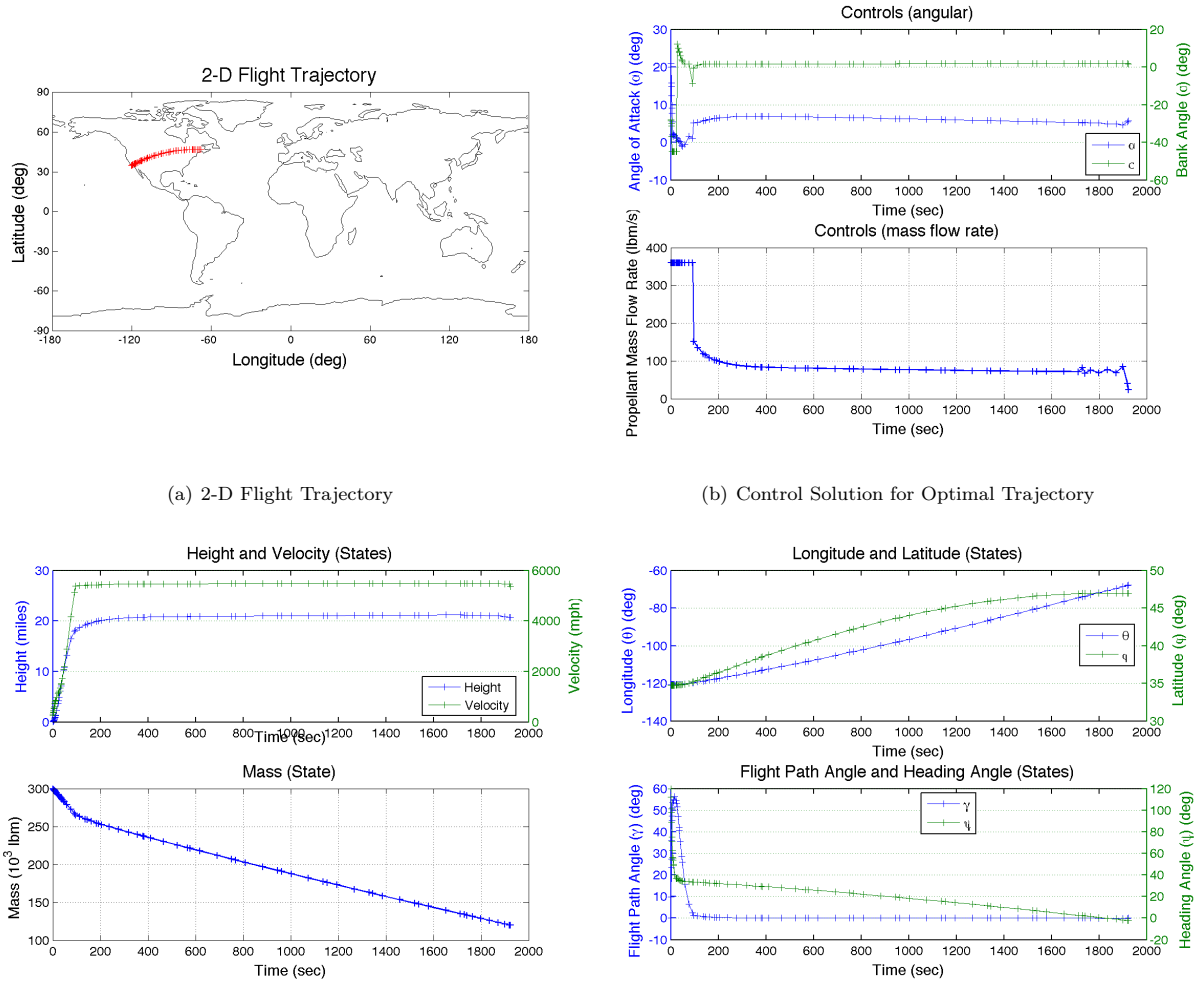
In each of the figures, the results are plotted against relevant independent variables. The vertical hash marks in each of the plot curves represent the set of collocation points in the optimal solution. Most of the figures are self explanatory except for the Mesh Tolerance and Maximum Relative Error plot, which shows how maximum relative error of the NLP solution for each mesh iteration. The trend over mesh iterations shows how quickly the ultimate solution converges to meet the user-specified mesh tolerance.

In each of the scenarios, there are transients in the optimal trajectory solutions, especially evident in the control subfigures. These transients occur when the vehicle is around Mach 2 and again at around Mach 6. This corresponds to the propulsion system transition from turbine to ramjet modes and then from ramjet to scramjet modes. At these Mach numbers, the lookup tables are relatively sparse, and even though spline fits are used with the lookup tables, this sparsity results in quickly varying trajectory parameters. If the GHAME lookup tables had more refined data points in these regimes, this phenomena would be reduced.

A. Minimum time scenario

This scenario is for a minimum time trajectory from California to Maine. The vehicle takes off from a California runway, climbs and turns to its destination, then cruises until it overflies the Maine runway. Since this is a minimum time problem, Figure 7 shows that after the vehicle leaves the

runway, it quickly gains altitude and turns towards the destination. Once it reaches cruise altitude, it cruises at its maximum Mach number until it overflies the destination (unlike the takeoff from a California runway, there is no enforced altitude at which the vehicle overflies the Maine runway). Figure 8 shows that for a majority of the trajectory, except during the initial climb, the only active path constraint is the maximum Mach number. During the rapid ascent, the active path constraint is φ due to the rapidly decreasing air mass flow, as well as the lookup table limit on φ. For this scenario, the flight duration is 1923 seconds. The trend of the Maximum Relative Error illustrates the exponential convergence quality of a spectral method solution.



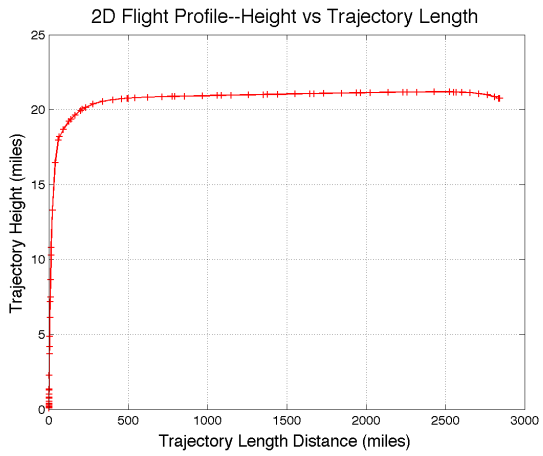
(a) 2-D Flight Trajectory

(b) Control Solution for Optimal Trajectory

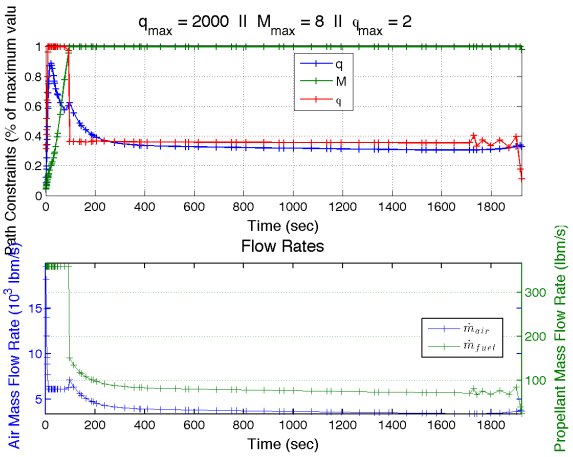
(c) State Solution for Optimal Trajectory (1 of 2)

(d) State Solution for Optimal Trajectory (2 of 2)

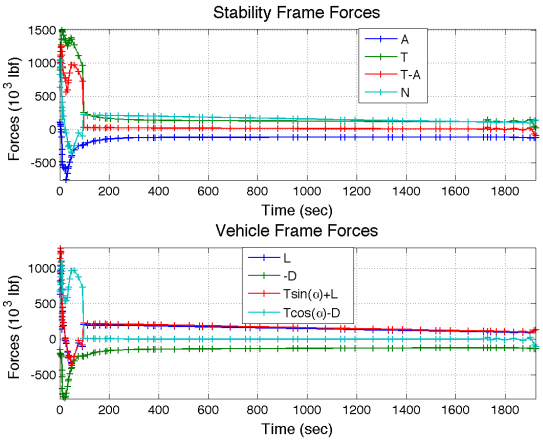
Fig. 7 Optimal trajectory solution for minimum time problem



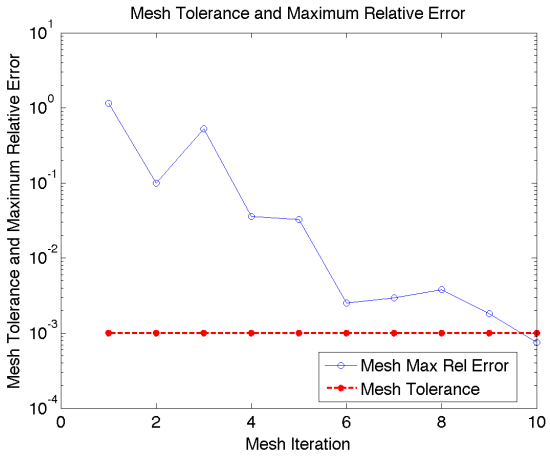
(a) 2-D Height Profile



(b) Path Constraints and Flow Rates for Optimal Trajectory



(c) Body and Stability Frame Forces Profile



(d) Mesh Interval and Collocation Point Profile

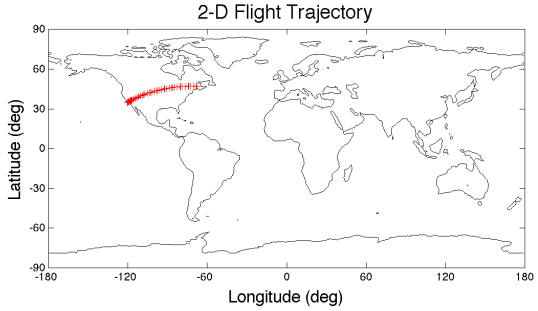
Fig. 8 Path parameters for minimum time optimal trajectory solution

B. Minimum time with control penalty scenario

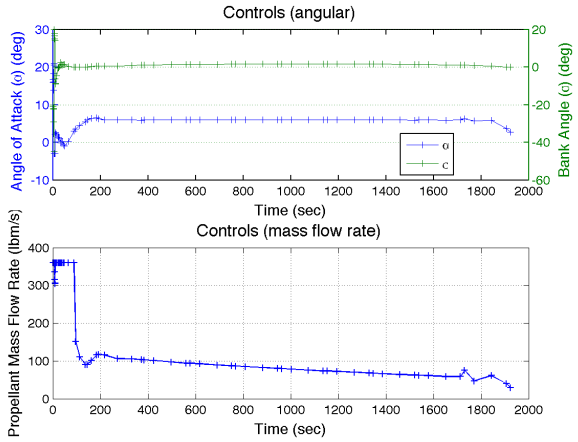
This scenario is very similar to the previous minimum time scenario except that a control usage penalty is added and the mesh tolerance is set to 1e-2 (this scenario does not converge to the default mesh tolerance within 20 iterations). As shown in Section III C, ϵ is the convex combination coefficient that is used to specify the contributions of both the (scaled) time and (scaled) control usage for objective function. In this scenario, all the controls have an equal contribution to the control penalty and ϵ is set to 0.9, significantly favoring the control penalty over the flight time.

Like the previous scenario, the vehicle takes off from a California runway, climbs and turns to its

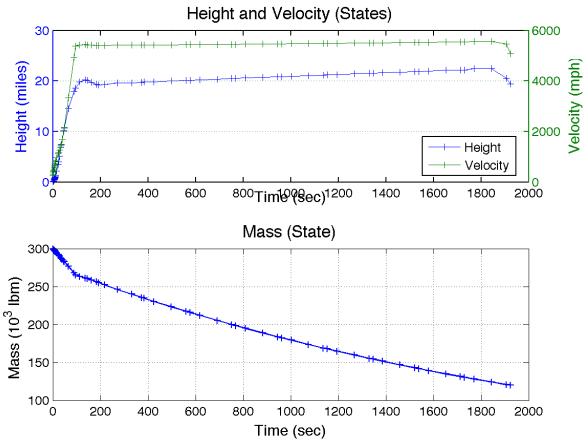
destination, then cruises until it overflies the Maine runway. Since this is a minimum time problem, Figure 9 shows that after the vehicle leaves the runway, it quickly gains altitude and turns towards the destination, albeit at a sightly slower rate than the minimum time scenario. This scenario has a lower initial cruise altitude but slowly climbs throughout the flight profile, while flying at the maximum Mach number. Figure 10 again shows that for a majority of the trajectory, except during the initial climb, the only active path constraint is the maximum Mach number. During the rapid ascent, the active path constraint is φ due to the rapidly decreasing air mass flow, as well as the physical limit on φ. For this scenario, the flight duration is 1924 seconds.



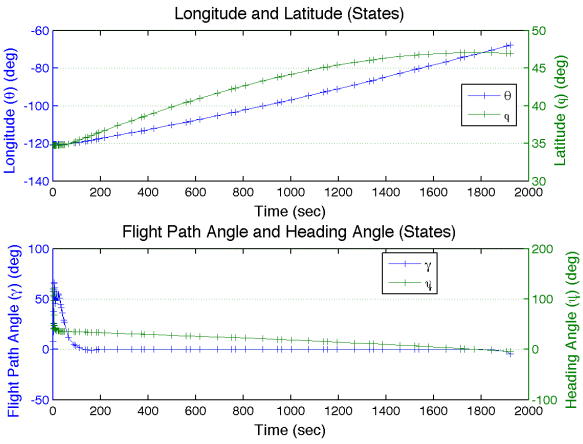
(a) 2-D Flight Trajectory



(b) Control Solution for Optimal Trajectory

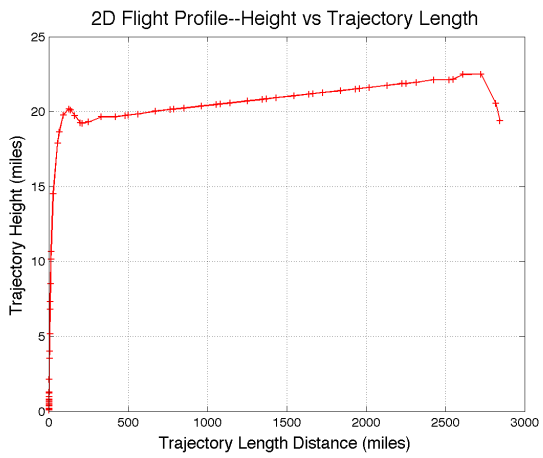


(c) State Solution for Optimal Trajectory (1 of 2)

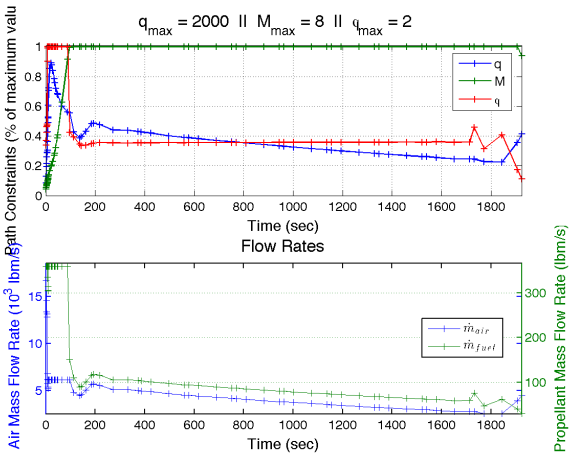


(d) State Solution for Optimal Trajectory (2 of 2)

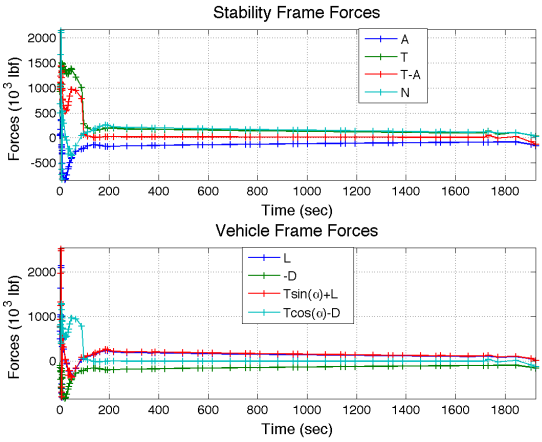
Fig. 9 Optimal trajectory solution for minimum time with control penalty problem



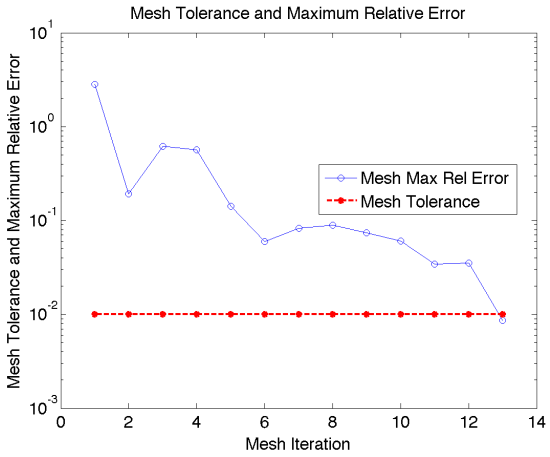
(a) 2-D Height Profile



(b) Path Constraints and Flow Rates for Optimal Trajectory



(c) Body and Stability Frame Forces Profile



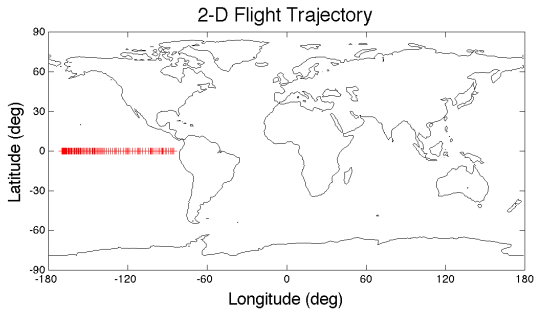
(d) Mesh Interval and Collocation Point Profile

Fig. 10 Path parameters for minimum time (with control penalty) optimal trajectory solution

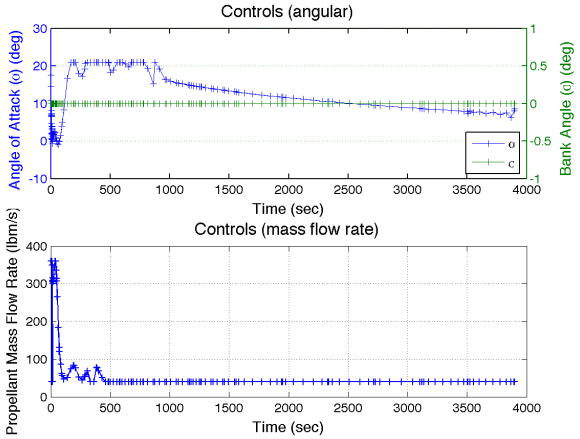
C. Maximum range scenario

This scenario is used to determine the nominal maximum range for this vehicle under specific conditions using the optimal control formulation defined by Equation 24. It assumes that the entire flight will occur along a latitude line, so that measuring the ground path distance is a function of the initial and final longitudes. The vehicle is flown starting at the equator and given an initial heading along the equator so that the vehicle does not have to turn. As a result, the limits for the bank angle control (σ) were set to 0, as well as the heading angle (ψ). The vehicle flew over the equator until all the fuel was spent.

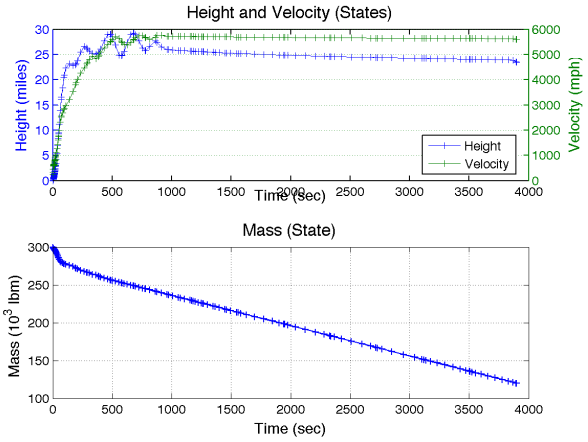
As seen in Figures 11 and 12, there is an oscillation in the altitude during the initial climb resulting from oscillations in the α control, due the φ path constraint becoming active. This is a similar behavior to that shown by Zipfel in his vehicle simulation using GHAME data sets. [19] The resulting range was 5851 miles.



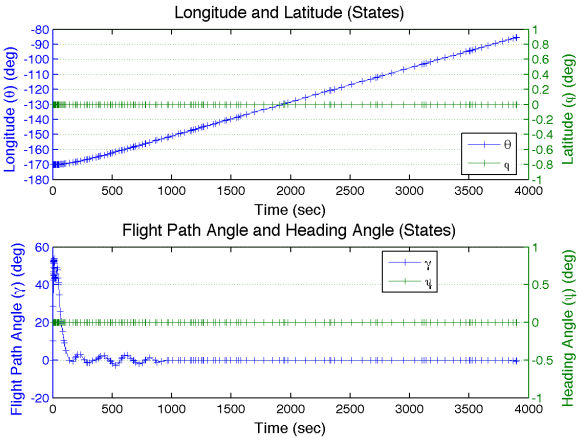
(a) 2-D Flight Trajectory



(b) Control Solution for Optimal Trajectory



(c) State Solution for Optimal Trajectory (1 of 2)

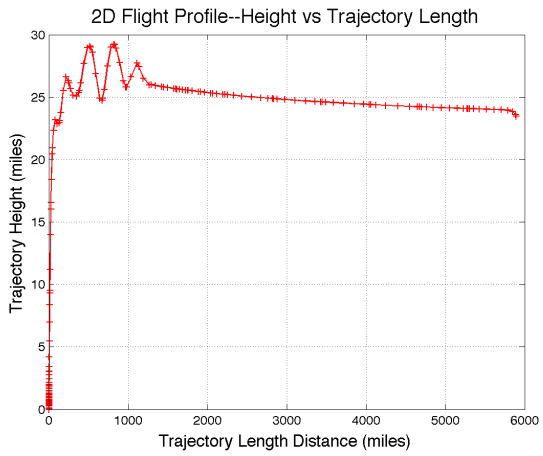


(d) State Solution for Optimal Trajectory (2 of 2)

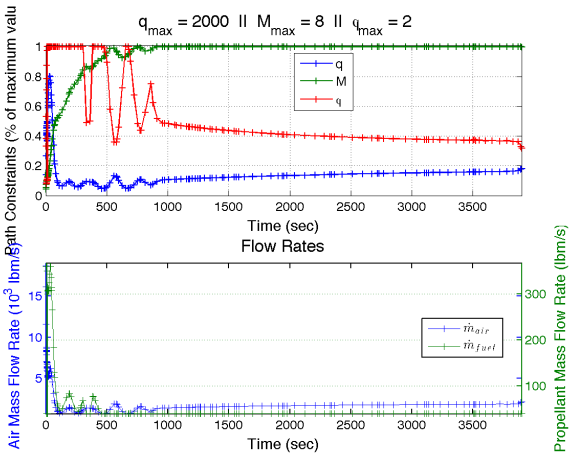
Fig. 11 Optimal trajectory solution for maximum range problem

D. Minimum time with tanking and no-fly zone scenario

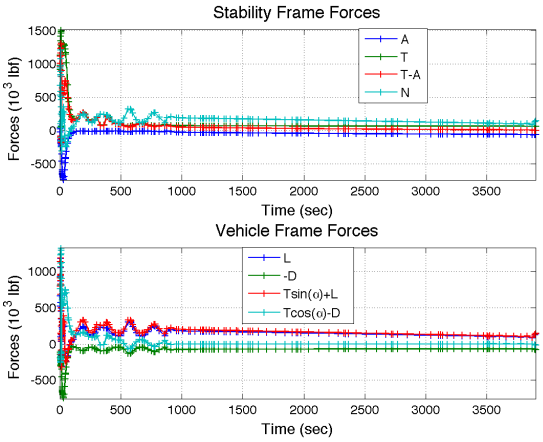
This scenario is similar to the first minimum time scenario in Section IV A except that a tanking event along the trajectory was added. The distance between the initial and final locations is greater



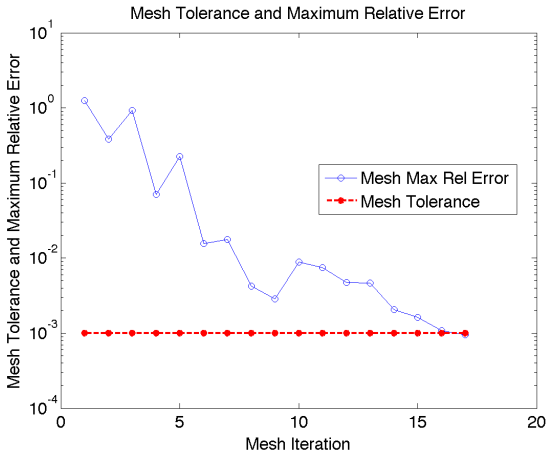
(a) 2-D Height Profile



(b) Path Constraints and Flow Rates for Optimal Trajectory



(c) Body and Stability Frame Forces Profile



(d) Mesh Interval and Collocation Point Profile

Fig. 12 Path parameters for maximum range optimal trajectory solution

than the maximum unrefueled range. In addition, a no-fly zone was added.

Similar to the previous scenario, the vehicle takes off from a Hawaii runway, climbs and turns to its destination, then continues until it overflies the United Kingdom runway. This scenario has a lower initial cruise altitude than the nominal minimum time scenario but this lower altitude is only maintained for a short duration until the vehicle descends to rendezvous with a tanker. The refueling event is modeled as the vehicle instantaneously taking on additional fuel at $M = 0.8$ and $h = 30,000ft$. After being refueled, the vehicle climbs to a cruising altitude until it reaches its destination. The refueling location is not predetermined; however, it is limited to occurring at least

100 miles from the initial or final destinations.

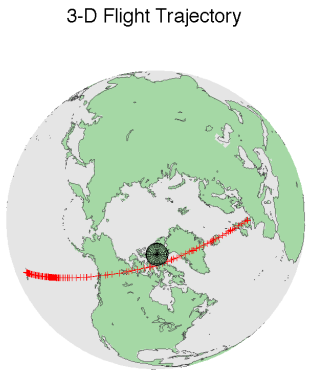
Given the minimum time formulation, the vehicle quickly tries to descend to rendezvous with the tanker. Since the GHAME lookup tables have a limited range for both α and φ, the vehicle does not have an effective means of shedding energy from its initial cruise profile of 90,000ft and Mach 8 (unlike the Space Shuttle that flew several high-α S-curves). Thus, the vehicle undergoes several "bang-bang" maneuvers to shed energy to rendezvous with the tanker at the lower altitude/speed. During this descent, both φ and q path constraints become active.

In the optimal control construct, this refueling event drives the scenario to become two separate optimal control problems with state continuity conditions. This reformulation is called phasing or pseudospectral knots. [26] This results in both optimal control problems being solved simultaneously while meeting the state continuity conditions at the phasing boundary between the two problems. This scenario also includes a no-fly zone, which is a 300 mile hemisphere centered on the earth's surface on a point that would have been on the vehicle's ground track if there was not a no-fly zone. This zone is shown in Figure 13a.

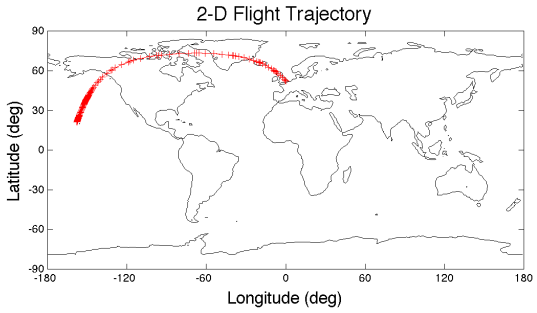
Figure 14 shows that for a majority of the trajectory, except during the climb and descent maneuvers, the only active path constraint is the maximum Mach number. During the rapid ascent and descents, the active path constraint is either φ or q due to the rapidly decreasing air mass flow (and physical limit on φ) or high speeds in the lower atmosphere, respectively. For this scenario, the flight duration is 4851 seconds.

V. Conclusions

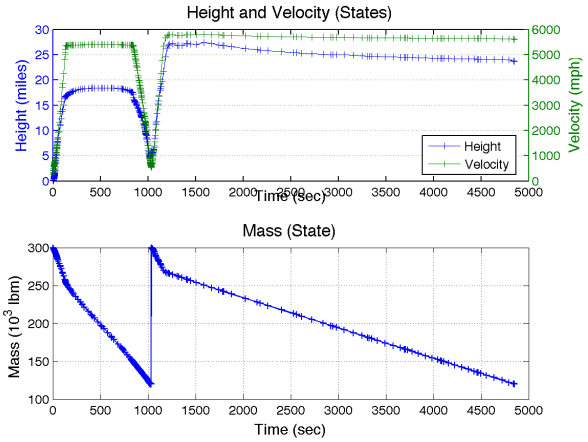
In this paper, an optimal flight path methodology for an air-breathing hypersonic vehicle was demonstrated incorporating pseudospectral methods. This paper developed and demonstrated a framework to calculate optimal trajectories for several different objective functions with the resulting optimal trajectories satisfying the specified constraints and error tolerances. The advantage of the methodology is that a priori definition of guidance commands do not have to be developed to generate optimal trajectories for several different types of flight profiles. Future work includes the incorporation of aerothermodynamic models as a path constraint, incorporating vehicle thermal



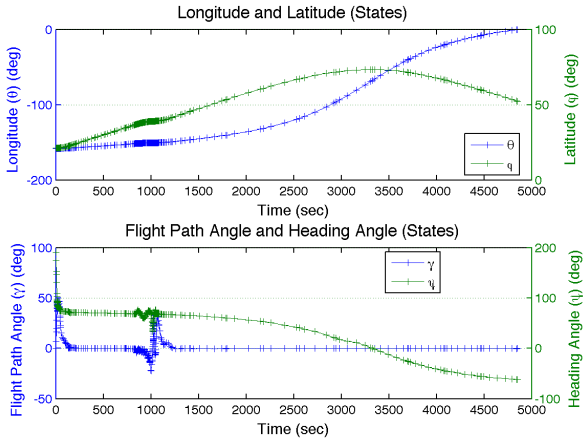
(a) 3-D Flight Trajectory



(b) 2-D Flight Trajectory



(c) State Solution for Optimal Trajectory (1 of 2)

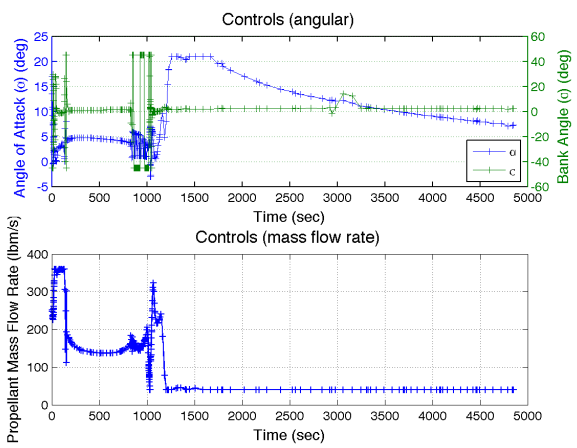


(d) State Solution for Optimal Trajectory (2 of 2)

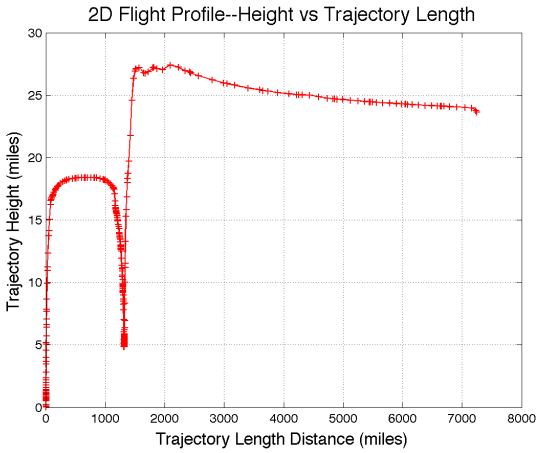
Fig. 13 Optimal trajectory solution for minimum time problem with tanking and no-fly zone

protection system limitations. As a result of this work, a designer can quickly develop optimal trajectories for a hypersonic aircraft on a remote sensing mission, including vehicle-driven constraints, geographic no-fly zones, and aerial refueling. Several different objective functional can be considered, in addition to varied path constraints.

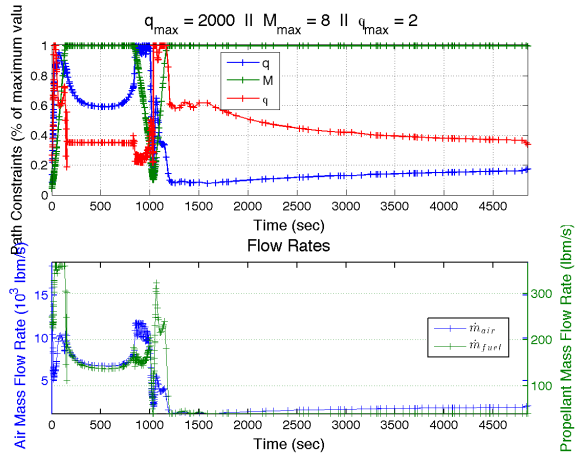
The use of the open-source GHAME model allowed the results of this effort to be distributed without restrictions, but there were limitations in the applicability of the data. The α and φ data limitations bounded the vehicle's flight regime and the limited number of data points in the lookup tables resulted in transients in the optimal trajectory solutions, even with using spline-based



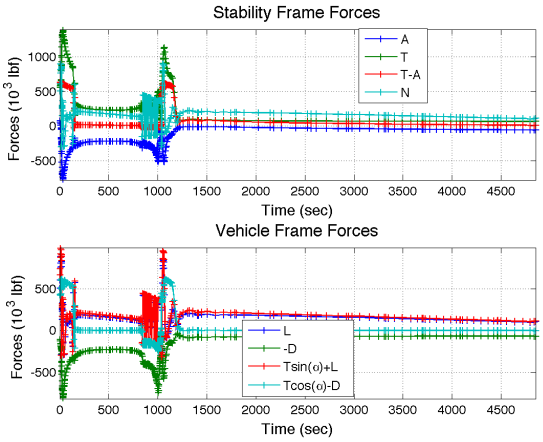
(a) Control Solution for Optimal Trajectory



(b) 2-D Height Profile



(c) Path Constraints and Flow Rates for Optimal Trajectory



(d) Body and Stability Frame Forces Profile

Fig. 14 Path parameters for minimum time problem with tanking and no-fly zone optimal trajectory solution

interpolation methods. If the GHAME lookup tables had more refined modeling in the turbine-to-ramjet (Mach 2) and ramjet-to-scramjet (Mach 6) regimes, this phenomena would be reduced.

VI. Acknowledgements

The authors would like to thank the United States Air Force Research Laboratory Aerospace Systems Directorate (AFRL/RQ) and the Science, Mathematics, and Research for Transformation (SMART) Defense Education Program for sponsoring this research.

VII. Disclaimer

The views expressed in this paper are those of the authors and do not reflect the official policy or position of the United States Air Force, the Department of Defense, or the U.S. Government.

References

[1] Nelson, D., "Qualitative and Quantitative Assessment of OTIS and POST," AE 8900 Individual Research Project, Georgia Institute of Technology, Apr 2001.

[2] Bryson, A., "Optimal Control-1950 to 1985," *Control Systems, IEEE*, Vol. 16, No. 3, 1996, pp. 26–33.

[3] Dong, C., Chao, T., yan Wang, S., and Yang, M., "Rapid Three-Dimensional Constrained Trajectory Generation for Near Space Hypersonic Vehicles," *18th AIAA/3AF International Space Planes and Hypersonic Systems and Technologies Conference*, 2012.

[4] Keshmiri, S., Colgren, R., and Mirmirani, M., "Trajectory Optimization for a Generic Hypersonic Vehicle," *AIAA/AHI Space Planes and Hypersonic Systems and Technologies Conferennce*, 2006, pp. 1–8.

[5] Zhou, H., Chen, W., and Yin, X., "Hypersonic Vehicle Trajectory Design Based on Optimal Control Theory," *Sixth International Symposium on Instrumentation and Control Technology: Sensors, Automatic Measurement, Control, and Computer Simulation*, International Society for Optics and Photonics, 2006, pp. 63582L–63582L–8.

[6] Rao, A. V., "Variable-Order Gaussian Quadrature Collocation Methods for Optimal Control and Software GPOPS-II Seminar," May 2013.

[7] Murillo, O. J., "A Fast Ascent Trajectory Optimization Method for Hypersonic Air-breathing Vehicles," 2010.

[8] Shi, Y., Xu, M., and Wang, Z., "Constrained trajectory optimization for hypersonic unpowered glider in near space using gauss pseudospectral method," *Electronic and Mechanical Engineering and Information Technology (EMEIT), 2011 International Conference on*, Vol. 7, IEEE, 2011, pp. 3804–3807.

[9] Shi, Y., Wan, Z., and Xu, M., "Multiphase Trajectory Optimization for Hypersonic Vehicle Using Gauss Pseudospectral Method," *International Journal of Aerospace and Lightweight Structures*, Vol. 3, No. 1, 2013, pp. 153–165.

[10] Song, C., Zhao, G., and Zhi, Y., "Boost phase trajectory optimization for hypersonic vehicle based on GPM," *Computer Research and Development (ICCRD), 2011 3rd International Conference on*, Vol. 4, IEEE, 2011, pp. 166–170.

[11] Bolender, M. A. and Doman, D. B., "Flight Path Angle Dynamics of Air-Breathing Hypersonic Vehi-

cles," Tech. Rep. AFRL-VA-WP-TP-2006-301, Air Force Research Laboratory Air Vehicles Directorate, 2005.

[12] Frendreis, S. G., Skujins, T., and Cesnik, C. E., "Six-Degree-of-Freedom Simulation of Hypersonic Vehicles," *AIAA Atmospheric Flight Mechanics Conference*, 2009.

[13] Bowers, A. and Iliff, K., "Generic Hypersonic Aerodynamic Model Example (GHAME) for Computer Simulation," *NASA TM, NASA Ames-Dryden, Edwards, CA*, 1987.

[14] White, D., Bowers, A., Iliff, K., and Menousek, J., *Flight, Propulsion, and Thermal Control of Advanced Aircraft and Hypersonic Vehicles*, Handbook of Intelligent Control: Neural, Fuzzy, and Adaptative Approaches, Van Nostrand Reinhold Company, New York, 1992, pp. 357–465, chapter: Flight, Propulsion, and Thermal Control of Advanced Aircraft and Hypersonic Vehicles.

[15] Araki, J. J., "Reentry Dynamics and Handling Qualities of a Generic Hypersonic Vehicle," 1992.

[16] Watson, K. and Liston, G., "DOF36 Formulation Report," August 2008.

[17] Bollino, K. P., Ross, I. M., and Doman, D. D., "Optimal Nonlinear Feedback Guidance for Reentry Vehicles," *Proceeding of AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2006, pp. 563–582.

[18] Sartorius, S., "STDATMO: Standard Atmosphere Function," 2013.

[19] Zipfel, P. H., *Modeling and Simulation of Aerospace Vehicle Dynamics*, AIAA, Reston, VA, 2nd ed., 2000.

[20] Vinh, N., *Optimal Trajectories in Atmospheric Flight*, Elsevier, 1981.

[21] Liston, G. and Adamczak, D., "Personal Communication," July 2013.

[22] Kirk, D. E., *Optimal Control Theory: An Introduction*, Dover, Mineola, NY, 2012.

[23] Patterson, M. A. and Rao, A. V., "GPOPS-II A MATLAB Software for Solving Multiple-Phase Optimal Control Problems Using hp-Adaptive Gaussian Quadrature Collocation Methods and Sparse Nonlinear Programming," *ACM Transactions on Mathematical Software (TOMS)*, Vol. 39, No. 3, 2013, pp. 1.

[24] Gourgoulhon, E., "Introduction to Spectral Methods," *4th EU Network meeting, Palma de Mallorca*, 2002.

[25] Kawajir, Y., Laird, C., and Wachter, A., "Introduction to IPOPT: A Tutorial for Downloading, Installing, and Using IPOPT," 2006.

[26] Ross, I. M. and Fahroo, F., "Pseudospectral Knotting Methods for Solving Optimal Control Problems," *Journal of Guidance, Control, and Dynamics*, Vol. 27, No. 3, 2004, pp. 397–405.

Bibliography

1. "Speed is the New Stealth". *Economist, The*, 31 July 2013. URL http://www.economist.com/news/technology-quarterly/21578522-hypersonic-weapons-building-vehicles-fly-five-times-speed-sound.

2. "Air Force Mission, Vision, Core Competencies, and Distinctive Capabilities", 6 August 2013. URL http://www.airforce.com/learn-about/our-mission/.

3. Aeronautics, National, Space Administration, National Oceanic, Atmospheric Administration, and U.S. Air Force. *US Standard Atmosphere, 1976*. NASA/NOAA/USAF Technical Report NOAA-SIT 76-1562 or NASA-TM-X-74335, US Government Printing Office, 1976.

4. Al-Garni, AZ and JB Barlow. "Aerospace Plane Ascending Trajectories with Heat Consideration". *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 210(3):231–245, 1996.

5. Allwine, Daniel A., Joseph E. Fisher, Jeremy A. Strahler, Douglas A. Lawrence, Michael W. Oppenheimer, and David B. Doman. "On-line Trajectory Generation for Hypersonic Vehicles". *Proceedings of the 2005 AIAA Guidance, Navigation, and Control Conference*. 2005.

6. Araki, John Jun. *Reentry Dynamics and Handling Qualities of a Generic Hypersonic Vehicle*. MS Thesis, Massachusetts Institute of Technology, 1992.

7. Benson, David A. *A Gauss Pseudospectral Transcription for Optimal Control*. PhD Dissertation, Massachusetts Institute of Technology, 2005.

8. Benson, David A., Geoffrey T. Huntington, Tom P. Thorvaldsen, and Anil V. Rao. "Direct Trajectory Optimization and Costate Estimation via an Orthogonal Collocation Method". *Journal of Guidance, Control, and Dynamics*, 29(6):1435–1440, 2006.

9. Betts, John T. *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2nd edition, 2010.

10. Bolender, Michael A. and David B. Doman. *Flight Path Angle Dynamics of Air-Breathing Hypersonic Vehicles*. Technical Report AFRL-VA-WP-TP-2006-301, Air Force Research Laboratory Air Vehicles Directorate, 2005.

11. Bollino, Kevin P. *High-Fidelity Real-time Trajectory Optimization for Reusable Launch Vehicles*. PhD Dissertation, Naval Postgraduate School, 2006.

12. Bollino, Kevin P., I. Michael Ross, and David D. Doman. "Optimal Nonlinear Feedback Guidance for Reentry Vehicles". *Proceeding of AIAA Guidance, Navigation, and Control Conference and Exhibit*, 563–582. 2006.

13. Bowers, AH and KW Iliff. "Generic Hypersonic Aerodynamic Model Example (GHAME) for Computer Simulation". *NASA TM, NASA Ames-Dryden, Edwards, CA*, 1987.

14. Bryson, Arthur. "Optimal Control-1950 to 1985". *Control Systems, IEEE*, 16(3):26–33, 1996.

15. Bryson, Arthur. *Dynamic Optimization*. Addison Wesley Longman, Menlo Park, 1999.

16. Buren, Mark A. Van and Kenneth D. Mease. "Minimum-fuel Ascent to Orbit Using Air-Breathing propulsion". *American Control Conference, 1989*, 2482–2488. IEEE, 1989.

17. Chapman, Dean R. "An Approximate Analytical Method for Studying Entry into Planetary Atmospheres". 1958.

18. Chudej, Kurt, Hans Josef Pesch, Markus Wchter, Gottfried Sachs, and Florent Le Bras. *Instationary Heat-Constrained Trajectory Optimization of a Hypersonic Space Vehicle by ODEPDE-Constrained Optimal Control*, 127–144. Variational Analysis and Aerospace Engineering. Springer, New York, NY, 2009.

19. Coello, Carlos A. Coello. "A Comprehensive Survey of Evolutionary-based Multiobjective Optimization Techniques". *Knowledge and Information Systems*, 1(3):129–156, 1999.

20. Darby, Christopher. *hp-Pseudospectral Method for Solving Continuous-Time Non-linear Optimal Control Problems*. PhD Dissertation, University of Florida, 2011.

21. DeJarnette, F., F. Cheatwood, H. Hamilton, and K. Weilmuenster. "A Review of Some Approximate Methods Used in Aerodynamic Heating Analyses". *Journal of Thermophysics and Heat Transfer*, 1(1):5–12, 1987.

22. Department of the Air Force. *Air Force Basic Doctrine, Organization and Command*. Air Force Doctrine Document (AFDD) 1, October 2011.

23. Dewell, Larry D., Jason L. Speyer, Michael Dinkelmann, Gottfried Sachs, Larry Dewell, Jason Speyer, Michael Dinkelmann, and Gottfried Sachs. "Fuel-Optimal Periodic Solutions to Hypersonic Cruise wwith Active Engine Cooling". *Guidance, Navigation, and Control Conference*, 1997.

24. Dinkelmann, M., M. Wchter, and G. Sachs. "Modelling and Simulation of Unsteady Heat Transfer for Aerospacecraft Trajectory Optimization". *Mathematics and Computers in Simulation*, 53(4):389–394, 2000.

25. Dong, Chen, Tao Chao, Song yan Wang, and Ming Yang. "Rapid Three-Dimensional Constrained Trajectory Generation for Near Space Hypersonic Vehicles". *18th AIAA/3AF International Space Planes and Hypersonic Systems and Technologies Conference*. 2012.

26. Ernstmeyer, J. *Hypersonic Aerospace Sensor Technology (HAST) Program Summary*. In-House Technical Report AFRL-SN-HS-TR-2005-027, AFRL/SN, July 2005.

27. Fay, James A. and F. R. Riddell. "Theory of Stagnation Point Heat Transfer in Dissociated Air". *Journal of the Aerospace Sciences*, 25(2):73–85, 1958.

28. Fornberg, Bengt. *A Practical Guide to Pseudospectral Methods*. Cambridge University Press, Cambridge, 1998.

29. Frendreis, Scott GV, Torstens Skujins, and Carlos ES Cesnik. "Six-Degree-of-Freedom Simulation of Hypersonic Vehicles". *AIAA Atmospheric Flight Mechanics Conference*. 2009.

30. Fry, Ronald S. "A Century of Ramjet Propulsion Technology Evolution". *Journal of Propulsion and Power*, 20(1):27–58, 2004.

31. Gambier, Adrian and Essameddin Badreddin. "Multi-objective Optimal Control: An Overview". *Control Applications, 2007. CCA 2007. IEEE International Conference on*, 170–175. IEEE, 2007.

32. Garg, Divya, Michael A. Patterson, Camila Francolin, Christopher L. Darby, Geoffrey T. Huntington, William W. Hager, and Anil V. Rao. "Direct Trajectory Optimization and Costate Estimation of Finite-horizon and Infinite-horizon Optimal Control Problems using a Radau Pseudospectral Method". *Computational Optimization and Applications*, 49(2):335–358, 2011.

33. Gill, Philip E., Walter Murray, and Michael A. Saunders. "SNOPT: An SQP Algorithm for Large-scale Constrained Optimization". *SIAM Review*, 47(1):99–131, 2005.

34. Golub, Gene H. and Charles F van Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, 3rd edition, 1996.

35. Gourgoulhon, Eric. "Introduction to Spectral Methods". *4th EU Network meeting, Palma de Mallorca*. 2002.

36. Grant, Michael J., Ian G. Clark, and Robert D. Braun. "Rapid Simultaneous Hypersonic Aerodynamic and Trajectory Optimization Using Variational Methods". *AIAA Atmospheric Flight Mechanics Conference and Exhibit, Portland, OR*, 8–11. 2011.

37. Gundlach, Jay. *Designing Unmanned Aircraft Systems: A Comprehensive Approach*. American Institute of Aeronautics and Astronautics, Reston VA, 2012.

38. Hankey, Wilbur L. *Reentry Aerodynamics*. AIAA, Washington, DC, 1988.

39. Heiser, William H. and David T. Pratt. *Hypersonic Airbreathing Propulsion*. AIAA, Washington, DC, 1994.

40. Janicki, William Daniel. *Asymptotic Analysis of Hypersonic Vehicle Dynamics Along Entry Trajectory*. MS Thesis, Massachusetts Institute of Technology, 1991.

41. Kawajir, Yoshiaki, Carl Laird, and A. Wachter. "Introduction to IPOPT: A Tutorial for Downloading, Installing, and Using IPOPT", 2006.

42. Keshmiri, Shahriar, Richard Colgren, and Maj Mirmirani. "Trajectory Optimization for a Generic Hypersonic Vehicle". *AIAA/AHI Space Planes and Hypersonic Systems and Technologies Conferennce*, 1–8, 2006.

43. Kirk, Donald E. *Optimal Control Theory: An Introduction*. Dover, Mineola, NY, 2012.

44. Kreh, B. *Hypersonic Aerospace Sensor Technology (HAST) Program: EO/IR Sensor Feasibility Study and Technology Roadmap*. In-House Technical Report AFRL-SN-HS-TR-2006-0020, AFRL/SN, August 2006.

45. Liston, Glenn and David Adamczak. "Personal Communication", July 2013.

46. Louderback, Pierce Mathew. *A Software Upgrade of the NASA Aeroheating Code "MINIVER"*. MS Thesis, Florida Institute of Technology, 2013.

47. Marler, R. Timothy and Jasbir S. Arora. "Survey of Multi-objective Optimization Methods for Engineering". *Structural and Multidisciplinary Optimization*, 26(6):369–395, 2004.

48. McClinton, Charles R. "X-43 Scramjet Power Breaks the Hypersonic Barrier: Dryden Lectureship in Research for 2006". *AIAA*, 1:2006, 2006.

49. Meirovitch, Leonard. *Methods of Analytical Dynamics*. McGraw-Hill, New York, 1970.

50. Mercier, Robert A. *Hypersonic Propulsion - Transforming the Future of Flight*. Presentation ASC 03-1846, AIAA 100 Years of Flight, 2003.

51. Mirmirani, Maj, Chivey Wu, Andrew Clark, Sangbum Choi, and Richard Colgren. "Modeling for Control of a Generic Airbreathing Hypersonic Vehicle". *AIAA*, 6256:2005, 2005.

52. Mooij, Erwin and PG Hanninen. "DIstributed Global Trajectory Optimization of a Moderate Lift-to-Drag Re-entry Vehicle". *AIAA Guidance, Navigation, and Control Conference*. 2009.

53. Murillo, Oscar J. *A Fast Ascent Trajectory Optimization Method for Hypersonic Air-breathing Vehicles*. PhD Dissertation, Iowa State University, 2010.

54. Murillo, Oscar J. and Ping Lu. "Fast Ascent Trajectory Optimization for Hypersonic Air-Breathing Vehicles". *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 1–17. 2010.

55. Mutzman, Richard. "X-51 Hypersonic Flight Test,Innovative Aero Transport Systems". *Innovative Aero Transport Systems*. AIAA, Washington DC, Sept 13, 2010 2010.

56. Nagel, AL, LB Doyle, and HD Fitzsimmons. *Analysis of Hypersonic Pressure and Heat Transfer Tests on Delta Wings with Laminar and Turbulent Boundary Layers*. Technical Report 535, National Aeronautics and Space Administration, 1966.

57. Nelson, Doug. *Qualitative and Quantitative Assessment of OTIS and POST*. AE 8900 Individual Research Project, Georgia Institute of Technology, Apr 2001.

58. Nizami, SSAK Javeed. *Control of Hypersonic Flight Trajectories to Minimize Heat Load*. MS Thesis, King Fahd University of Petroleum and Minerals, 1994.

59. Ordaz, Irian. *A Probabilistic and Multi-Objective Conceptual Design Methodology for the Evaluation of Thermal Management Systems on Air-Breathing Hypersonic Vehicles*. PhD Dissertation, Georgia Institute of Technology, 2008.

60. Patterson, Michael A. and Anil V. Rao. "GPOPS-II A MATLAB Software for Solving Multiple-Phase Optimal Control Problems Using hp-Adaptive Gaussian Quadrature Collocation Methods and Sparse Nonlinear Programming". *ACM Transactions on Mathematical Software (TOMS)*, 39(3):1, 2013.

61. Pesch, Hans Josef. "The Princess and Infinite-Dimensional Optimization". *Documenta Mathematica*, 345–356, 2012.

62. Pietz, Jesse. *Pseudospectral Collocation Methods for the Direct Transcription of Optimal Control Problems*. MS Thesis, Rice University, 2003.

63. Rao, Anil V. "Variable-Order Gaussian Quadrature Collocation Methods for Optimal Control and Software GPOPS-II Seminar", May 2013.

64. Rao, Anil V., Michael A. Patterson, and William W. Hager. "A ph Collocation Scheme for Optimal Control". *Automatica*, 2013.

65. Reich, G., J. Hinger, and M. Huchler. "Thermal Protection Systems for Hypersonic Transport Vehicles". *20th Intersociety Conference on Environmental Systems*, 1990.

66. Ross, I. Michael and Fariba Fahroo. "Pseudospectral Knotting Methods for Solving Optimal Control Problems". *Journal of Guidance, Control, and Dynamics*, 27(3):397–405, 2004.

67. Sachs, Gottfried and Michael Dinkelmann. "Heat Input Reduction in Hypersonic Flight by Optimal Trajectory Control". *AIAA Guidance, Navigation and Control Conference*, 1996.

68. Sartorius, S. "STDATMO: Standard Atmosphere Function", 2013. URL http://www.mathworks.com/matlabcentral/fileexchange/28135-stdatmo--standard-atmosphere-function/content/stdatmo.m.

69. Shaffer, Patrick J. *Optimal Trajectory Reconfiguration and Retargeting for the X-33 Reusable Launch Vehicle*. PhD Dissertation, Naval Postgraduate School, 2004.

70. Shaffer, Patrick J., I. Michael Ross, Michael W. Oppenheimer, and David B. Doman. *Optimal Trajectory Reconfiguration and Retargeting for a Reusable Launch Vehicle*. Defense Technical Information Center, 2005.

71. Shen, Jie, Tao Tang, and Li-Lian Wang. *Spectral Methods: Algorithms, Analysis and Applications*, volume 41. Springer, Berlin, 2011.

72. Shi, Yuyang, Zi Wan, and Min Xu. "Multiphase Trajectory Optimization for Hypersonic Vehicle Using Gauss Pseudospectral Method". *International Journal of Aerospace and Lightweight Structures*, 3(1):153–165, 2013.

73. Shi, Yuyang, Min Xu, and Zi Wang. "Constrained trajectory optimization for hypersonic unpowered glider in near space using gauss pseudospectral method". *Electronic and Mechanical Engineering and Information Technology (EMEIT), 2011 International Conference on*, volume 7, 3804–3807. IEEE, 2011.

74. Song, Chao, Guorong Zhao, and Yue Zhi. "Boost phase trajectory optimization for hypersonic vehicle based on GPM". *Computer Research and Development (ICCRD), 2011 3rd International Conference on*, volume 4, 166–170. IEEE, 2011.

75. Suplisson, Angela, Nathan Smith, and Tadeusz Masternak. "AFIT AERO899 Independent Study Course Material under direction of Drs Baker, Cobb, and Jacques", 2012.

76. Sussmann, Hector J. and Jan C. Willems. "300 Years of Optimal Control: from the Brachystochrone to the Maximum Principle". *Control Systems, IEEE*, 17(3):32–44, 1997.

77. Thompson, Loren. "Hypersonic Jet Hits over 3,300 mph, Started at Wright-Patt". *Dayton Daily News*, 18 Aug 2013. URL http://www.mydaytondailynews.com/news/news/local-military/hypersonic-jet-hits-over-3300-mpg-started-at-wrigh/nXmw6/?icmp=daytondaily_internallink_textlink_apr2013_daytondailystubtomydaytondaily_launch.

78. US Air Force Scientific Advisory Board. *Why and Whither Hypersonics Research in the US Air Force*. SAB-TR-00-03, 2000.

79. Vinh, Nguyen. *Optimal Trajectories in Atmospheric Flight*. Elsevier Scientific Software, New York, NY, 1981.

80. Wachter, Andreas. "Short Tutorial: Getting Started with IPOPT in 90 Minutes". *Combinatorial Scientific Computing (U.Naumann, O.Schenk, HD Simon, eds.)*, 2009.

81. Wachter, M. and G. Sachs. "Constraining Heat Input by Trajectory Optimization for Minimum-Fuel Hypersonic Cruise". *Asian Journal of Control*, 8(4):307–313, 2006.

82. Watson, Kenneth and Glenn Liston. "DOF36 Formulation Report", August 2008.

83. Watson, Kenneth and Glenn Liston. "DOF36 Flight Simulation User's Manual", May 2012.

84. White, David, Albion Bowers, Ken Iliff, and John Menousek. *Flight, Propulsion, and Thermal Control of Advanced Aircraft and Hypersonic Vehicles*, 357–465. Handbook of Intelligent Control: Neural, Fuzzy, and Adaptative Approaches. Van Nostrand Reinhold, New York, NY, 1992. Chapter: Flight, Propulsion, and Thermal Control of Advanced Aircraft and Hypersonic Vehicles.

85. Windhorst, Robert, MD Ardema, and JV Bowles. "Minimum Heating Re-entry Trajectories for Advanced Hypersonic Launch Vehicles". *AIAA Guidance, Navigation and Control Conference*, 354, 1997.

86. Windhorst, Robert, MD Ardema, and JV Bowles. "Minimum Heating Entry Trajectories for Reusable Launch Vehicles". *Journal of Spacecraft and Rockets*, 35(5):672, 1998.

87. Wright, Michael J. *Aerothermal Modeling for Entry and Aerocapture*. Technical Report D9P1-NSTC-07-0026, NASA, 2007.

88. Wurster, Kathryn E. "MINIVER User's Guide", October 2000.

89. Yu, Li and Cui Nai-gang. "Optimal Attack Trajectory for Hypersonic Boost-Glide Missile in Maximum Reachable Domain". *Mechatronics and Automation, 2009. ICMA 2009. International Conference on*, 4012–4017. IEEE, 2009.

90. Zhao, Jiang and Rui Zhou. "Reentry Trajectory Optimization for Hypersonic Vehicle Satisfying Complex Constraints". *Chinese Journal of Aeronautics*, 26(6):1544–1553, 2013.

91. Zhou, Hao, Wanchun Chen, and Xingliang Yin. "Hypersonic Vehicle Trajectory Design Based on Optimal Control Theory". *Sixth International Symposium on Instrumentation and Control Technology: Sensors, Automatic Measurement, Control, and Computer Simulation*, 63582L–63582L–8. International Society for Optics and Photonics, 2006.

92. Zipfel, Peter H. *Modeling and Simulation of Aerospace Vehicle Dynamics*. AIAA, Reston, VA, 2nd edition, 2000.

Vita

In August 2011, Tadeusz Masternak arrived at the Air Force Institute of Technology (AFIT) at Wright-Patterson AFB in Ohio to pursue a doctoral degree in systems engineering. Previously, he received his bachelor's degree in aerospace engineering from Syracuse University and his master's degree in astronautical engineering at AFIT. After graduation, he will be assigned to the Air Force Research Laboratory working on system engineering related efforts for upcoming hypersonic systems technology development efforts and demonstration projects.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704–0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202–4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. **PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**

1. REPORT DATE *(DD–MM–YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From — To)*
26–12–2014	Doctoral Dissertation	Sept 2011 — Dec 2014

4. TITLE AND SUBTITLE

Multi-Objective Trajectory Optimization of a Hypersonic Reconnaissance Vehicle with Temperature Constraints

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Masternak, Tadeusz J., Mr., Civ.

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
WPAFB OH 45433–7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT-ENV-DS-14-D-21

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Research Laboratory, Aerospace Systems Directorate
Attn: Lt Col Clark Allred
2130 Eighth Street
WPAFB OH 45433–7542
937–255–9398 (DSN 785–9398) clark.allred@us.af.mil

10. SPONSOR/MONITOR'S ACRONYM(S)

AFRL/RQHV

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

DISTRIBUTION STATEMENT A:
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

14. ABSTRACT

Temperature-constrained optimal trajectories for a scramjet-based hypersonic reconnaissance vehicle were generated by developing an optimal control formulation and solving it using a variable order Gauss-Radau quadrature collocation method. The vehicle was assumed to be an air-breathing reconnaissance aircraft that has specified takeoff/landing locations, airborne refueling constraints, specified no-fly zones, and specified targets for sensor data collections. The aircraft model included flight dynamics, aerodynamics, and thermal constraints. This model was incorporated into an optimal control formulation that includes constraints on both the vehicle as well as mission parameters, such as avoidance of no-fly zones and coverage of high-value targets. Optimal trajectories were be developed using several different performance costs in the optimal control formulation—minimum time, minimum time with control penalties, and maximum range. The resulting analysis demonstrated that optimal trajectories that meet specified mission parameters and constraints can be determined and used for larger-scale operational and campaign planning.

15. SUBJECT TERMS

Optimal Control, Hypersonics, Trajectory Optimization, Direct Orthogonal Collocation, Aerothermal Constraints

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Dr. David R. Jacques, AFIT/ENV
U	U	U	UU	416	19b. TELEPHONE NUMBER *(include area code)* (937) 255–3636, x3329; david.jacques@afit.edu