OPTIMAL PATH PLANNING OF AN UNMANNED COMBAT AERIAL VEHICLE WITH OBSTACLE AVOIDANCE

A Thesis

presented to

the Faculty of the Graduate School

at the University of Missouri-Columbia

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

---

by

IBRAHIM HALIL CIHAN

Dr. Craig Kluever, Thesis Supervisor

JULY 2016

The undersigned, appointed by the dean of the Graduate School, have examined the thesis entitled

OPTIMAL PATH PLANNING OF AN UNMANNED COMBAT

AERIAL VEHICLE WITH OBSTACLE AVOIDANCE

presented by Ibrahim Halil Cihan,

a candidate for the degree of Master of Science,

and hereby certify that, in their opinion, it is worthy of acceptance.

Professor Craig Kluever

Professor Ming Xin

Professor Mahmoud Almasri

ACKNOWLEDGEMENTS

I would sincerely like to thank my advisor, Dr. Kluever, for giving me a huge amount of guidance through the research. I appreciate him always helping me when the need arises. I feel much honored to have a mentor who is one of the wonderful professors in academia. I would like to thank my co-advisor, Dr. Xin, for contributing and his support on my research. I would also like to thank Dr. Almasri for being on my committee.

I am grateful to Fawaz Al-Bakri for collaborative works on aircraft flight mechanics and aerospace propulsion projects at the University of Missouri-Columbia. I am also grateful to Miles Barnhart for his assistance with an optimization algorithm in MATLAB.

I need to thank my Turkish Government for their financial support during my Master's degree. They have given me a great opportunity to study abroad for a Master's and PhD degree. Finally, I appreciate everyone who made a contribution in my educational life.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS ii

LIST OF FIGURES v

LIST OF TABLES vii

SYMBOLS viii

ABBREVIATIONS x

ABSTRACT xi

1 INTRODUCTION 1

1.1 Background and Motivation 1

1.2 Prior Work 3

1.3 Thesis Outline 6

2 SYSTEM MODELS 8

2.1 Aircraft Description 8

2.2 Aircraft Flight Mechanics 9

2.2.1 Assumptions 9

2.2.2 Flight Dynamics 11

2.2.2.1 Climbing Flight 12

2.2.2.2 Cruise Dynamics 17

2.2.2.2.1 Constant Cruise Altitude 17

2.2.2.2.2 Variable Cruise Altitude 19

2.3 Propulsion System 20

2.4 Obstacle Avoidance Model 24

3 OPTIMIZATION 26

3.1 Problem Definition 26

3.2 Fmincon Algorithm 27

3.2.1 Objective Function 28

3.2.1.1 Method 1 – Initial Value Problem (IVP) 28

3.2.1.2 Method 2 – Inverse Dynamics 31

3.2.1.2.1 Shortest Path 34

3.2.1.2.2 Geometric Model 35

4 NUMERICAL RESULTS 37

4.1 Variable Cruise Altitude vs. Constant Cruise Altitude 37

4.2 Optimal Path without Obstacles 40

4.3 Optimal Paths with Obstacle Avoidance 46

4.3.1 Scenario 1 – One-Way Path with One Destination Point 46

4.3.1.1 Varying Cruise Altitude 52

4.3.2 Scenario 2 – Closed-Circuit Path with Three Destination Points 53

5 SUMMARY AND CONCLUSION 59

REFERENCES 61

APPENDIX A 64

LIST OF FIGURES

2.1: X-47A Model [22] 8

2.2: Four Forces Acting on X-47A for SLUF 11

2.3: Force Vectors during Climbing Flight 12

2.4: Power Curves for X-47A at 30,000 ft 15

2.5: Absolute Ceiling for X-47A at 47,906 ft 16

2.6: Level Left Turn for X-47A 17

2.7: Low BPR JT15D-5 Turbofan [25] 21

2.8: Thrust Variation with Mach Number 22

2.9: TSFC Variation with Altitude 23

2.10: Radar for Detection Airplanes [32] 25

3.1: System Dynamics for IVP 28

3.2: Parameters of System Dynamics for IVP 29

3.3: Throttle vs Time throughout Cruise 30

3.4: Bank Angle vs Time throughout Cruise 31

3.5: System Dynamics for Inverse-Dynamics 31

3.6: Velocity vs Time throughout Cruise 32

3.7: Heading Angle vs Time throughout Cruise 32

3.8: X-Coordinate Horizontal Velocity vs Time throughout Cruise 32

3.9: Y-Coordinate Horizontal Velocity vs Time throughout Cruise 33

3.10: Shortest Path by Avoiding Obstacles without Flight Mechanics 34

3.11: Geometric Approach without Flight Mechanics 36

4.1: Altitude vs. Range for Case 1 and Case 2 38

4.2: Aircraft Weight vs. Range for Cases 1 and Case 2 39
4.3: Optimal Path Planning without Obstacles at 30,000 ft 40
4.4: Optimal Throttle Profile Using IVP Method (No obstacles) 42
4.5: Optimal Bank Angle Profile Using IVP Method (No obstacles) 42
4.6: Optimal Velocity Profile Using Inverse Dynamics Method (No obstacles) 44
4.7: Optimal Heading Angle Profile Using Inverse Dynamics Method (No obstacles) ... 44
4.8: Optimal Mass Profile Using Inverse Dynamics Method (No obstacles) 45
4.9: Shortest Path by Avoiding 16 PSRs without Flight Mechanics 48
4.10: Geometric Model without Flight Mechanics 49
4.11: Optimal Path Planning for Scenario 1 for Cruise at 30,000 ft 50
4.12: Optimal Velocity Values by Avoiding 16 PSRs at Varying Cruise Altitudes 52
4.13: Minimized Fuel Consumption of the Aircraft at Different Cruise Altitudes 53
4.14: Optimal Path Planning for Scenario 2 with Cruise at 30,000 ft 55
4.15: Optimal Velocity Profile for Scenario 2 with Cruise at 30,000 ft 56
4.16: Optimal Bank Angle Profile for Scenario 2 with Cruise at 30,000 ft 57
4.17: Optimal Heading Angle Profile for Scenario 2 with Cruise at 30,000 ft 58

LIST OF TABLES

1.1: Size Categories of UAVs according to the US Department of Defense (DoD) 1

2.1: General Specifications of a X-47A 8

2.2: Performance Specifications of a X-47A 9

2.3: Wing Specifications of a X-47A 9

2.4: Typical Values of *CD0* and *e* for Subsonic Aircraft 9

2.5: Iterations of *CD0* and *e* 10

2.6: Engine Specifications of X-47A [19], [24], [25] 20

4.1: Total Ranges of Case1 and Case 2 37

4.2: Fuel Consumption at Segments 45

4.3: Optimization Results of IVP and Inverse Dynamics Methods 46

4.4: Coordinates and Detection Ranges of 16 PSRs for Scenario 1 47

4.5: Optimized Slope from the Geometric Model Solution 49

4.6: Optimization Results of Velocity, Heading Angle, and Mass Variables 51

4.7: Optimization Results for 10 and 20 Discrete Stages at 30,000 ft 52

4.8: Coordinates and Detection Ranges of 16 PSRs for Scenario 2 54

SYMBOLS

a	Acceleration (ft/s^2)
$a_{heading\_angle}$	Heading angle scaling factor
a_{flight_time}	Flight-time scaling factor
$a_{velocity}$	Velocity scaling factor
A_1	Inlet area (ft^2)
AR	Aspect ratio
b	Wingspan (ft)
c	Inequality constraint
c_{eq}	Equality constraint
c_t	Thrust specific fuel consumption (1/hr)
C_D	Drag coefficient
C_{D0}	Zero-lift drag coefficient
C_L	Lift coefficient
C_{Lmax}	Maximum lift coefficient
C_{root}	Root chord (ft)
C_{tip}	Tip chord (ft)
D	Drag (lb_f)
e	Oswald coefficient
G_r	Receive antenna gain
G_t	Transmit antenna gain
h	Altitude (ft)
K	Lift-induced drag coefficient
L	Lift (lb_f)
$(L/D)_{max}$	Maximum lift-over-drag ratio
m	Mass (slug)
M	Mach number
N	Number of stages
P_A	Power available (lb_f-ft/s)
P_R	Power required (lb_f-ft/s)
P_r	Received power (W)
P_t	Transmit power (W)
r_{radar}	Radius of radar (ft)
R	Turn radius (ft)
RC_{max}	Maximum rate-of-climb (ft/s)

R_{max}	Maximum detectable range (m)
s	Horizontal position (ft)
S	Wing area (ft^2)
t_{climb}	Climbing time (s)
t_{flight}	Flight time (s)
T	Thrust (lb_f)
V	Velocity (ft/s)
W	Weight (lb_f)
X	Position in direction of X axis (ft)
Y	Position in direction of Y axis (ft)
γ	Flight-path angle (deg)
δ	Throttle (-)
λ	Radar operating frequency wavelength (m)
ρ	Density of air ($slug/ft^3$)
σ	Radar cross section (m^2)
ϕ	Bank angle (deg)
Ψ	Heading angle (deg)
$\bar{q}$	Dynamic pressure (lb_f/ft^2)

ABBREVIATIONS

BPR	Bypass Ratio
DCNLP	Direct Collocation with Nonlinear Programming
DoD	US Department of Defense
FPA	Flight-Path Angle
ISR	Intelligence Surveillance Reconnaissance
IVP	Initial Value Problem
MILP	Mixed Integer Linear Programming
NM	Nautical Mile
PPR	Pulse Repetition Rate
PSR	Primary Surveillance Radar
RCS	Radar Cross Section
RPA	Remotely Piloted Aircraft
RRT	Rapidly Exploring Random Tree
SDE	State Differential Equation
SLUF	Steady Level Unaccelerated Flight
SSR	Secondary Surveillance Radar
TSFC	Thrust Specific Fuel Consumption
UAS	Unmanned Aircraft System
UAV	Unmanned Aerial Vehicle
UCAS-D	Unmanned Combat Air System Demonstrator
UCAV	Unmanned Combat Aerial Vehicle

ABSTRACT

The objective of this study is to optimize path planning for an unmanned combat aerial vehicle (UCAV) with obstacle avoidance by minimizing fuel consumption. A nonlinear constrained optimization solver, Fmincon, is used to find the optimal path. Two separate methods are implemented to obtain an excellent path and reduce the computation time. In the first method, an initial value problem (IVP) is used where discrete values of throttle and bank angle are the optimum control variables. In the second method, an inverse-dynamics approach is used where discrete values of velocity, heading angle, and mass are the optimization variables. Flight mechanics principles are applied to obtain thrust, bank angle, and position time histories. The geometrical path segments are minimized in order to estimate initial states for the aircraft flight mechanics model. Accuracy of the optimization is directly proportional to the number of variables in Fmincon. On the other hand, as the number of variables increases, so does the computational time to obtain the optimal path. Numerical results are presented for various scenarios.

CHAPTER 1 - INTRODUCTION

1.1 Background and Motivation

Unmanned Aerial Vehicles (UAVs) are aircraft without a human operator onboard. They can be controlled by an operator outside of the vehicle or by pre-programmed flight plans. The UAVs are also named as “drone”, “crone”, “remotely piloted aircraft (RPA)”, “unmanned aircraft system (UAS)”, or “unmanned combat aerial vehicle (UCAV)” [1]. The UAVs date back to World War I (WWI). The first unmanned aircraft in WWI was the aerial torpedo aircraft [2]. Since that time, UAVs have been developed all over the world for various military and civilian applications. UAVs can be classified according to size (Table 1.1), user and mission purpose [3], [4]:

Table 1.1: Size Categories of UAVs according to the US Department of Defense (DoD)

Size	Maximum Takeoff Weight (lb)	Operating Altitude (ft)	Airspeed (knots)
Small	0-20	<1,200	<100
Medium	21-55	<3,500	<250
Large	<1,320	<18,000	<250
Larger	>1,320	<18,000	Any airspeed
Largest	>1,320	>18,000	Any airspeed

User Categories:

- Military
- Public
- Commercial
- Private

Mission Purpose Categories:

- Intelligence, Surveillance, and Reconnaissance (ISR)
- Combat
- Payload Delivery
- Environmental
- Search and Rescue
- Training
- Communications
- Recreation

The UAVs are designed by taking into consideration their usage areas. While performance is most crucial for a fighter aircraft, the fuel efficiency is most important for a commercial aircraft. For instance, Lufthansa Airline Company sets goals to improve fuel efficiency by 1.5 percent per year up to 2020 and decrease CO_2 emissions in the atmosphere at the same time [5].

In this study, the largest size military aircraft, X-47A, is investigated for mainly ISR mission purpose at cruise. X-47A is known as a UCAV. The number of UCAVs is increasing day by day around the world. UCAVs usually carry bombs and/or missiles similar to the manned combat aircrafts. UCAVs can be used for environments that are dangerous to human life. Therefore, in recent years, developed countries have preferred to use UCAVs instead of manned combat aircraft. Another advantage of using UCAVs instead of manned combat aircraft is that UCAVs are smaller and lighter than the equivalent manned combat aircrafts since they are without pilot and cockpit.

As previously mentioned, the primary duty of X-47A is to fulfill ISR at cruise. While X-47A is doing ISR, it should avoid obstacles, which are radar systems on the aerial maps. As fuel efficiency increases, the aircraft can fly longer range. This provides more ISR.

There are several parameters that influence on the fuel consumption of an aircraft. Some of the parameters are related to the specifications of the aircraft, such as empty weight, engine types and numbers, wingspan, aspect ratio, etc. These specifications of X-47A are constant during flight.

On the other hand, some of the parameters are directly proportional to the fuel consumption during flight, such as flight speed, operating altitude, throttle, lift-to-drag ratio (*L/D*), etc. These parameters can be set in order to improve fuel efficiency during flight; hence, optimal path planning techniques are used to find the optimum parameters in this thesis. The previous works related to path planning are represented in the following section.

1.2 Prior Work

In recent decades, optimal path planning for UAVs is one of the most popular subjects among aircraft flight mechanics research. In the literature, different methods have been developed for solving the path-planning problem.

Mixed Integer Linear Programming (MILP) method was implemented to solve trajectory optimization for single or multiple vehicles in [6]–[9]. The MILP makes the dynamic model into a linearized form by using discrete state and control matrix. In [6], Schouwenaars et al. investigated the fuel-optimal path planning of multiple vehicles by avoiding obstacles and collision with each other. They used A Mathematical

Programming Language (AMPL) and CPLEX optimization algorithms. In [7], [8], Richards and How applied the MILP for spacecraft and aircraft applications. They minimized fuel consumption and flight time in separate cases. A large number of decision binary variables (0 or 1) were used in [7], [8]. The authors described square and polygon obstacle constraints instead of actual circular obstacle constraints so that nonlinear constraints convert into linear constraints. In [9], Forsmo implemented MILP method to minimize fuel consumption or flight time, or a combination of these in his thesis. The author preferred to utilize YALMIP and Gurobi optimization algorithms differently from [6]–[8]. The major disadvantage of using the MILP is the restriction to linear problems.

Kabamba, Meerkov, and Zeitz [10] presented optimal path planning for an UCAV under missile threats. They proposed the minimax optimal control method in order to minimize the maximum value of the probability of the aircraft being shot down. The authors emphasized that radar cross section (RCS) of the aircraft has a direct impact on probability of detection.

Bortoff's path-planning study consisted of two steps in order to minimize the total length penalized [11]. In the first step, the author generated the Voronoi diagram, which included the discrete vertices and edges. For the next step, virtual forces approach was applied by using the diagram solution as an initial estimation.

Cruise flight optimizations were investigated in [12], [13]. Ardema and Asuncion proposed to maximize the range of a commercial aircraft at a constant cruise altitude [12]. The authors performed the Brequet range equation (constant altitude and velocity), and a singular optimal control problem, and compared their results. Green's theorem was utilized to deal with the singular optimal control, which contained velocity and mass state

variables and a throttle-control variable. In [13], Ansberry found optimal velocity, cruise altitude, and throttle values in order to minimize fuel cost. In his thesis, the cruise-step approach was used rather than cruise-climb or constant cruise since it is more realistic. The author compared the results of the optimization and three Brequet range equations: case 1 (constant altitude and velocity), case 2 (constant velocity and lift coefficient), and case 3 (constant altitude and lift coefficient). Additionally, the author looked at the effects of jet streams over flight time and fuel consumption.

Geiger et al. maximized the viewing time of moving or stationary targets for the surveillance purpose in [14]. They used a direct collocation with nonlinear programming (DCNLP) method to solve optimal control problems. The states and controls were approximated as cubic polynomials. The authors converted an optimal control problem into a nonlinear problem and then integrated the state equations using numerical integration methods. The Fmincon algorithm in MATLAB was used as a nonlinear optimization program in their paper. In [15], Liu and his colleagues developed Gauss pseudospectral method, which is a kind of DCNLP, for stealth UAVs. The problems of minimum fuel consumption, minimum flight time and minimum probability of detection were examined separately when the aircraft performed the reconnaissance mission in this paper. In [16], Mohan investigated the multi-aircraft landing problem. The pseudospectral method was applied for optimal control problem in her thesis. In addition to that, a rapidly exploring random tree (RRT) algorithm was used in order to determine the shortest path between initial and destination coordinates. The RRT algorithm enables the author to obtain initial estimations for optimal control program.

Call presented the path planning for small UAVs in [17]. The author examined the path planning into two phases: pre-mission path planning (for avoiding known obstacles) and reactive path planning (for avoiding unknown obstacles). In the pre-mission path planning, RRT and genetic algorithms were utilized and their results were compared. The RRT method found an acceptable path quickly but not necessarily the shortest path for obstacle-rich environments. On the other hand, the genetic algorithm came up with a better path, but the computation time is longer than RRT. In [18], Yang and Wenjie created a new method to enhance traditional genetic algorithm through reasonable coding scheme and prior knowledge. This gave the authors the opportunity to decrease the computation time and converge to the solution more effectively.

The main objective of this thesis is to determine optimal paths for different scenarios in order to minimize fuel consumption. The secondary goal is to reduce the computation time of the optimization algorithm. Therefore, Initial Value Problem and Inverse Dynamics methods are used in Fmincon optimization solver to compare their results. The computer used in this research has following system specifications:

- Windows 7 Home Premium
- Intel® Core™ i5 CPU @ 2.67GHz
- 4 GB RAM
- 64-bit Operating System

1.3 Thesis Outline

In chapter 2 of this thesis, the flight mechanics and propulsion system of X-47A are described. This chapter also gives information about primary and secondary surveillance

radars. In chapter 3, the optimization problem is presented. The Fmincon interior-point algorithm is used as an optimization solver in MATLAB. Initial Value Problem (method 1) and Inverse Dynamics (method 2) are implemented to integrate the state differential equations. The shortest path is determined by avoiding obstacles in order to minimize range and fuel consumption. Then a geometric approach is applied in order to provide a good estimate for the discrete heading angles. Chapter 4 presents numerical results of the optimization problem. The cases for variable cruise altitude and constant cruise altitude are compared for fuel efficiency. Optimal paths are found for different scenarios in this chapter. Chapter 5 gives a summary and conclusion of the thesis.

CHAPTER 2 - SYSTEM MODELS

2.1 Aircraft Description

The Northrop Grumman X-47A Pegasus is used as an unmanned combat aerial vehicle (UCAV) model. It is a first variant of the X-47 series which is X-47A, X-47B and X-47C. The aircraft was built for Defense Advanced Research Project Agency (DARPA) as a proof-of-concept model by Boeing and Northrop Grumman in 2001. Later it joined US Navy's Unmanned Combat Air System Demonstrator (UCAS-D) program. After it was developed by Northrop Grumman, the first flight occurred on February 23, 2003. Figure 2.1 and Tables 2.1-2.3 represent the specifications of the aircraft [19]–[21].

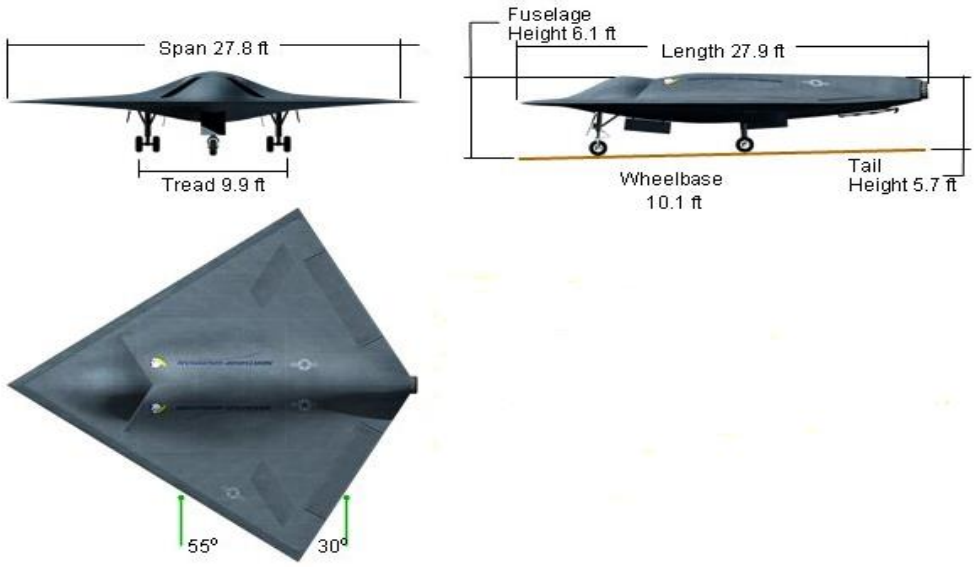

Fig. 2.1 X-47A Model [22]

Table 2.1: General Specifications of a X-47A

Length (ft)	Height (ft)	Wingspan (ft)	Empty weight (lb_f)	Fuel weight (lb_f)	Max takeoff weight (lb_f)	Payload (lb_f)
27.9	6.1	27.8	3836	1580	5903	2x500

Table 2.2: Performance Specifications of a X-47A

Max speed @35,000 ft	Max range (nmi)	Service ceiling (ft)	Thrust/Weight ratio	Max *L*/*D* ratio	Engine
Mach 0.76	1,500+	47,669	0.59	10.3373	PW-JT15D-5C

Table 2.3: Wing Specifications of a X-47A

C_{root} (ft)	C_{tip} (ft)	C_{Lmax}	Wing area (ft^2)	Leading edge sweep (deg)	Trailing edge sweep (deg)	Aspect ratio
27.9	0	1.46	387.29	55	30	1.9955

Aerodynamic and engine specifications of the X-47A are presented in detail in Sections 2.2.1 and 2.3, respectively.

2.2 Aircraft Flight Mechanics

2.2.1 Assumptions

All specifications of the UCAV are not available because of its classified nature. Some reasonable assumptions are required for flight mechanics calculations. Aerodynamic coefficients, zero-lift drag (C_{D0}), and Oswald (e), are unknown. Typical values of Oswald efficiency factor e and zero-lift drag coefficients C_{D0} for subsonic jets are shown in Table 2.4, according to [23].

Table 2.4: Typical Values of *CD0* and *e* for Subsonic Aircraft

C_{D0} range	e range
0.014 – 0.02	0.75-0.85

X-47A is a tailless delta-wing aircraft like the Avro Vulcan jet bomber and its values for e and C_{D0} 0.7 and 0.009, respectively [24]. Some iterations were made for e and C_{D0} in

Table 2.5 to match the published specifications for service ceiling (min 40,000 ft), maximum speed (high subsonic), and range (min 1,500 nmi).

Table 2.5: Iterations of *CD0* and *e*

e	*K*	C_{D0}	$(L/D)_{max}$	Altitude (ft)	Max speed (Mach)	Max range (nmi)
0.70	0.2279	0.014	8.8523	sea level	0.451	1,216 at 44,000 ft
				30,000	0.63	
				35,000	0.644	
				40,000	0.615	
				44,672	0.482	
0.75	0.2127	0.014	9.1630	sea level	0.452	1,243 at 44,000 ft
				30,000	0.632	
				35,000	0.647	
				40,000	0.623	
				45,389	0.482	
0.75	0.2127	0.009	11.4283	sea level	0.529	1,732 at 46,000 ft
				30,000	0.794	
				35,000	0.827	
				40,000	0.817	
				49,621	0.604	
0.75	0.2127	0.011	10.3373	sea level	0.493	1,516 at 46,000 ft
				30,000	0.716	
				35,000	0.74	
				40,000	0.723	
				47,669	0.545	

Oswald's coefficient is in the range of 0.70-0.75 for delta wings. C_{D0} values were taken as between 0.014 and 0.009. Although all iterations provide minimum service ceiling,

some of them fail to satisfy maximum speed and range. The value of C_{D0} is 0.014 is too high for maximum range and 0.009 is too small for maximum speed because Mach 0.827 is transonic speed. As e increases, maximum speed and range also increase. Therefore, optimum aerodynamic coefficients are chosen as C_{D0} = 0.011 and e =7.5.

2.2.2 Flight Dynamics

The four forces of flight, thrust (T), lift (L), drag (D), and weight (W), affect the aircraft as seen Figure 2.2. Thrust is produced by a low bypass ratio (BPR) turbofan engine in the direction of motion. Lift is created by differences in air pressure perpendicular to the oncoming flow direction. Drag is a kind of negative force caused by friction along the oncoming flow direction. The last force is weight, which is the gravity force toward the center of the Earth. Opposing forces are equal in steady, level, unaccelerated flight (SLUF). Newton's second law of motion, $F = ma$, is applied for balance of forces.

Fig. 2.2 Four Forces Acting on X-47A for SLUF

In this study, mainly cruise flight conditions are examined. Cruise occurs between ascent and descent phases, so climbing flight must be evaluated before cruise.

2.2.2.1 Climbing Flight

The aircraft climbs from sea level to the desired cruise altitude. Turning flight is not considered during climbing flight. Figure 2.3 shows climbing flight with the positive flight-path angle (FPA).

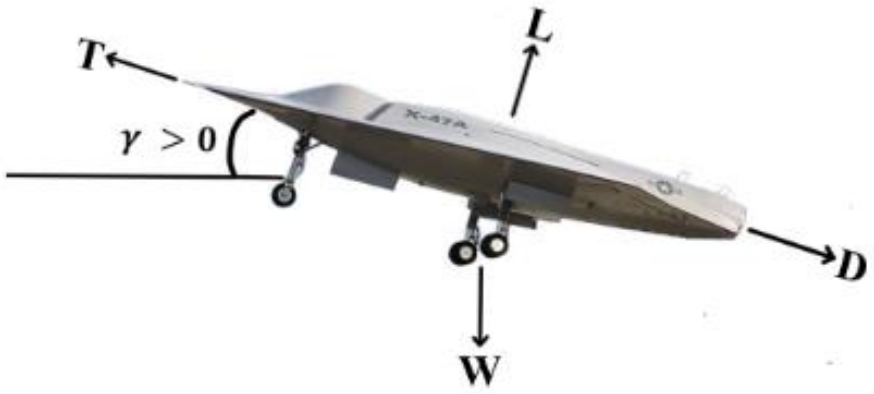

Fig. 2.3 Force Vectors during Climbing Flight

Equations (1) and (2) are derived by balancing of forces along and normal to flight path, respectively.

$$m\dot{V}_\infty = T - D - W\sin\gamma \tag{1}$$

$$mV_\infty\dot{\gamma} = L - W\cos\gamma \tag{2}$$

where *m* is mass of the aircraft. Kinematic equations (3) and (4) are the horizontal and vertical velocity components, respectively. Flight-path angle indicates aircraft flight conditions: ascent ($\gamma > 0$), descent ($\gamma < 0$), and level flight ($\gamma = 0$).

$$\dot{s} = V_\infty\cos\gamma \tag{3}$$

$$\dot{h} = V_\infty\sin\gamma \tag{4}$$

After solving equation (1) for the sine of FPA with constant velocity, it is inserted into equation (4) in order to get rate-of-climb (RC):

$$\dot{h} = V_\infty \sin\gamma = \frac{TV_\infty - DV_\infty}{W} \tag{5}$$

There are two terms in the numerator of the right-hand side of equation (5): power available ($P_A = TV_\infty$) and power required ($P_R = DV_\infty$). An equivalent RC equation can be written as:

$$\dot{h} = V_\infty \sin\gamma = \frac{TV_\infty - DV_\infty}{W} = \frac{P_A - P_R}{W} \tag{6}$$

The right side of equation (6) is defined as excess power divided by weight. In this study, maximum RC ($\mathrm{RC_{max}}$), which occurs at maximum excess power, is used for climbing flight calculation. Thrust *T* is obtained from an engine model and changes with altitude and velocity (see Section 2.2). Weight *W* decreases with time because of fuel consumption. Aerodynamic lift *L* and drag *D* are determined by equations (7) and (8).

$$L = \bar{q}_\infty S C_L \tag{7}$$

$$D = \bar{q}_\infty S C_D \tag{8}$$

where dynamic pressure $\bar{q}_\infty$ is

$$\bar{q}_\infty = \frac{1}{2}\rho_\infty V_\infty^2 \tag{9}$$

where ρ_∞ is air density, *S* is the wing area of the aircraft, C_L is the lift coefficient, and C_D is the drag coefficient. The lift coefficient for SLUF ($\dot{\gamma} = 0$) is computed from equations (2) and (1).

$$C_L = \frac{W \cos\gamma}{\bar{q}_\infty S} \tag{10}$$

The aerodynamic drag coefficient is computed using the standard drag polar:

$$C_D = C_{D0} + KC_L^2 \tag{11}$$

In this study, C_L equation can be simplified by taking $\cos\gamma \sim 1$ because FPA is small. Once C_L is determined, C_D can be calculated using the constant C_{D0} (chosen in the Section 2.2.1) and induced-drag coefficient, which is computed using:

$$K = \frac{1}{\pi e AR} \tag{12}$$

where *e* is Oswald's efficiency coefficient (selected in Section 2.2.1), and AR is the aspect ratio of the aircraft:

$$AR = \frac{b^2}{S} \tag{13}$$

In equation (13), *b* is the aircraft's wingspan. Finally, *D* can be calculated by using equations (8) - (13), and maximum RC can be determined by equation (6) with power required, power available, and weight. Figure 2.4 represents $P_A$and P_R vs. Mach number at 30,000 ft. The aircraft cannot exceed Mach 0.716 at 30,000 ft because power required is greater than power available in this flight regime.

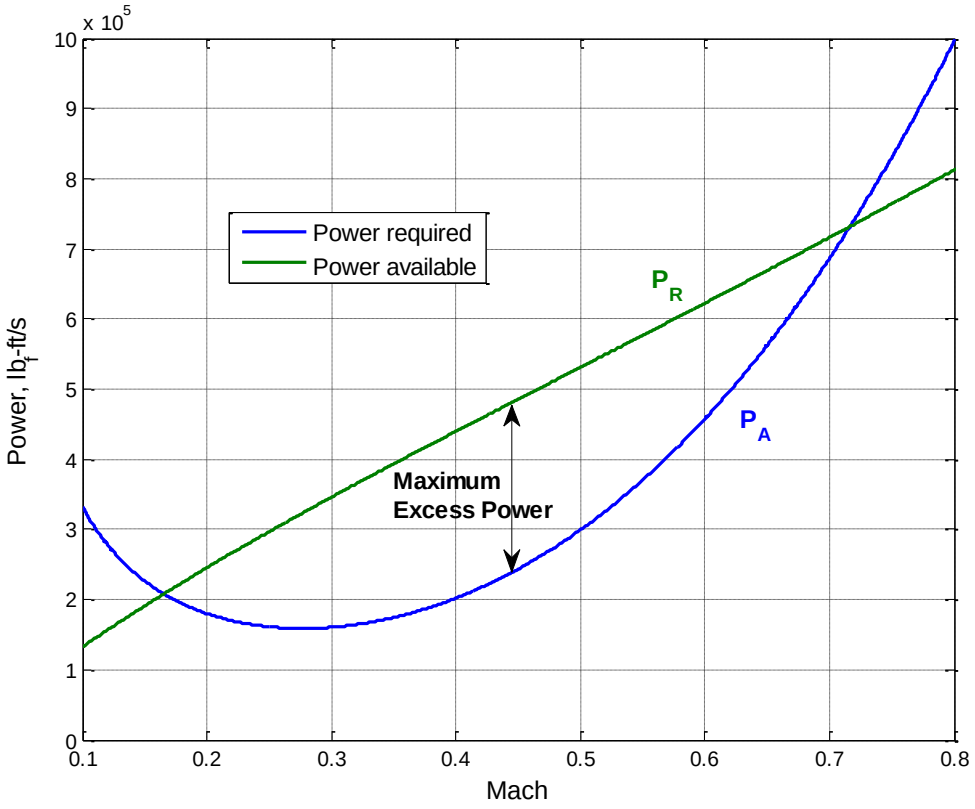

Fig. 2.4 Power Curves for X-47A at 30,000 ft

Also, RC provides the opportunity to determine the absolute ceiling of the aircraft. Absolute ceiling is the highest altitude where maximum excess power is equal to zero. In other words, the aircraft cannot fly above this altitude because it requires more power to climb to higher altitudes. However, power available is less than power required above absolute ceiling. Figure 2.5 shows absolute ceiling of the X-47A at 0.545 Mach number at 47,906 ft.

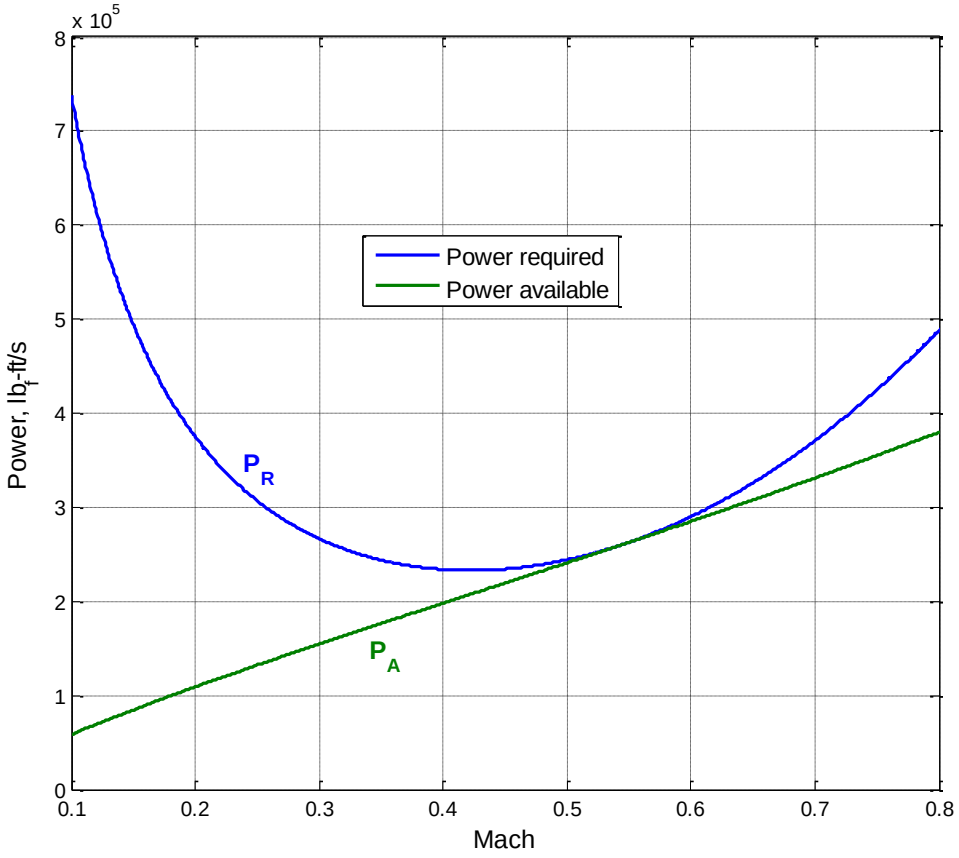

Fig. 2.5 Absolute Ceiling for X-47A at 47,906 ft

The last performance metric for climbing flight is the time-to-climb (t_{climb}), which can be calculated by equation (14).

$$t_{climb} = \int_{h_1}^{h_2} \frac{1}{RC_{max}} dh \tag{14}$$

In this thesis, equation (14) is solved by the trapezoidal numerical integration (M-file trapz) method in MATLAB. Integration limits h_1 and h_2 are the initial and final altitudes of climbing flight.

2.2.2.2 Cruise Dynamics

2.2.2.2.1 Constant Cruise Altitude

Cruise altitude is constant with zero FPA. Therefore, equation (1) should be rewritten in order to neglect $Wsin\gamma$. The result is

$$\dot{V}_\infty = \frac{T - D}{m} \tag{15}$$

There are three conditions that can be applied to equation (15). The first condition is SLUF, which means that *T* is equal to *D* so the aircraft has constant velocity. The second is the acceleration case where *T* is greater than *D* and velocity of the aircraft increases with time. The last condition is deceleration where *T* is less than *D*. Therefore, the velocity decreases with time.

In this study, the X-47A should be turning in order to avoid obstacles. These obstacles are radars placed on known locations. Therefore, another dynamical equation, heading rate or turn rate ($\dot{\Psi}$), must be taken into account for cruise. Figure 2.6 shows level turn of X-47A. The flight direction of the aircraft is out of the page in Figure 2.6.

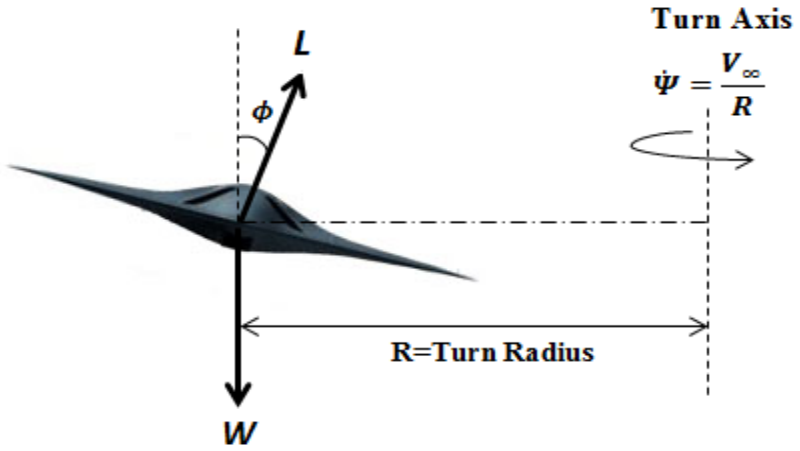

Fig. 2.6 Level Left Turn for X-47A

The heading rate is

$$\dot{\Psi} = \frac{V_\infty}{R} = \frac{L \sin \phi}{m V_\infty} \tag{16}$$

where R is the turn radius and ϕ is the bank angle as shown in Figure 2.6. According to the balance of forces along vertical and radial directions, equations (17) and (18) are derived from Figure 2.6.

$$L \cos \phi = W \tag{17}$$

$$L \sin \phi = m \frac{V_\infty^2}{R} \tag{18}$$

The aircraft has a maximum bank angle to provide a minimum turn radius. It cannot exceed the limit on bank angle because of equation (17). As ϕ increases in equation (17), L must increase because W is constant (Load factor

$$n = \frac{L}{W} = \frac{1}{\cos \phi} \tag{19}$$

is limited by structural limits). Then C_L, C_D, and D go up respectively due to equations (7), (11), and (8). Drag D can be equal to T_{max} in order to sustain a maximum turn rate. In addition, the aircraft should fly with the velocity that is greater than the stall speed for level turn. As velocity decreases, lift coefficient increases, but it has a limit on lift coefficient (C_{Lmax}). The stall speed also depends on load factor (n), and it can be computed by equation (20):

$$V_{stall} = \sqrt{\frac{2nW}{\rho S C_{Lmax}}} \tag{20}$$

There are two kinematic equations derived from equation (3) for constant cruise altitude:

$$\dot{X} = V_\infty \cos\gamma \cos\Psi = V_\infty \cos\Psi \tag{21}$$

$$\dot{Y} = V_\infty \cos\gamma \sin\Psi = V_\infty \sin\Psi \tag{22}$$

In equations (21) and (22), $\dot{X}$ and $\dot{Y}$ are the horizontal velocity components along X and Y axis, respectively. Cosine of FPA is equal to one in the both cases because FPA is zero during level cruise.

2.2.2.2.2 Variable Cruise Altitude

Next, consider cruise where the aircraft ascends ($\gamma > 0$) or descends ($\gamma < 0$). Ascent (climbing) flight has been investigated in Section 2.2.2.1. For descent flight, gliding flight is taken into consideration at cruise. The aircraft does not consume any fuel during gliding flight. Equations (1) – (4) are integrated to find the gliding range. It is important to get the maximum gliding range between the given initial and final altitudes. When the aircraft glides with the narrowest angle, the longest gliding range can be obtained. Straight and unaccelerated glide is assumed in order to determine the narrowest angle. Equations (1) and (2) are solved for $\dot{V}_\infty = 0$ and $\dot{\gamma} = 0$:

$$m\dot{V}_\infty = T - D - W\sin\gamma = 0 \Rightarrow D = -Wsin\gamma \tag{23}$$

$$mV_\infty\dot{\gamma} = L - W\cos\gamma = 0 \Rightarrow L = Wcos\gamma \tag{24}$$

Equation (23) and (24) are derived by dividing the modified equations (1) and (2). As the *L*/*D* ratio increases, the gliding angle decrease in equation (25). The narrowest glide angle takes place at maximum *L*/*D* $(L/D)_{max}$.

$$tan\gamma = \frac{-1}{L/D} \tag{25}$$

Zero lift drag coefficient (C_{D0}) is equal to induced drag coefficient (KC_L^2) at LD_{max}. Therefore, lift coefficient (C_L^*) and drag coefficient (C_D^*) for $(L/D)_{max}$ are defined in equations (26) and (27).

$$C_L^* = \sqrt{C_{D0}/K} \tag{26}$$

$$C_D^* = 2C_{D0} \tag{27}$$

Maximum L/D ratio can be calculated in equation (28) by divided equations (26) and (27).

$$(L/D)_{max} = \frac{C_L^*}{C_D^*} = \frac{\sqrt{C_{D0}/K}}{2C_{D0}} = \frac{1}{2\sqrt{C_{D0}/K}} \tag{28}$$

2.3 Propulsion System

X-47A has one Pratt & Whitney Canada JT15D-5C turbofan engine. It is a small turbofan developed in 1989. It is a low BPR turbofan where BPR is two. The JT15D-5C is used for other two aircraft: DASA Ranger 2000 and Agusta S211/S211A. Its specifications are summarized in Table 2.6.

Table 2.6: Engine Specifications of X-47A [19], [25], [26]

Thrust at takeoff (lb_f)	BPR	Turbine inlet temperature (°R)	Inlet diameter (in)	Length (ft)	Dry weight (lb_f)
3,190	2	1,751.7	27	5.25	665.8

The JT15D-5 has a single-stage high-pressure centrifugal compressor that is driven by a single-stage high-pressure turbine. The fan, which has wide chord blades, and the axial

boost stage are driven by single-stage low pressure turbine. Figure 2.7 shows the JT15D-5 engine model.

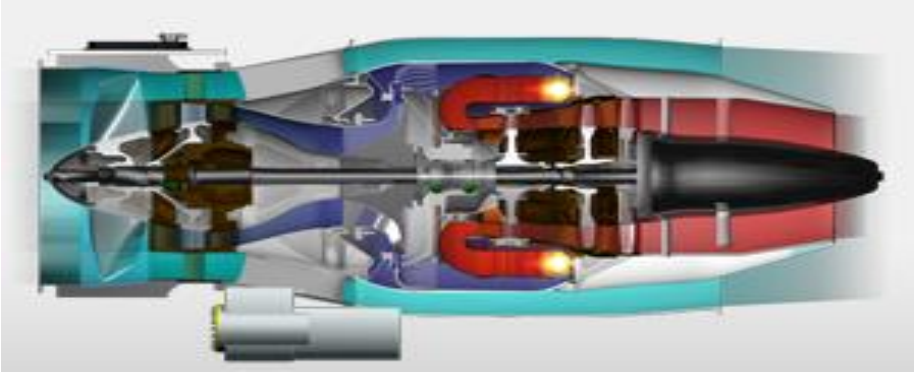

Fig. 2.7 Low BPR JT15D-5 Turbofan [25]

Compressor and fan pressure ratios are not available for JT15D-5C. They can be estimated from other variants of JT15D-5 because each engine in JT15D-5 differs slightly. The pressure ratio for the compressor, which is in the range of 12.3-13.1 for other variants of JT15D-5, is chosen as 12.3 [27]. The fan pressure ratio is taken as 1.5 [28]. After all pressure loss ratios and efficiencies are determined, function m-file is created in MATLAB; see Appendix. The function has two outputs: thrust (T) and thrust specific fuel consumption ($TSFC$ or c_t), and two inputs, altitude (h) and velocity (V). The function $[T, c_t] = turbofan\_JT15D5C(h, V)$ is able to calculate T and c_t for each altitude and velocity. Figure 2.8 presents thrust vs. Mach number at different altitudes.

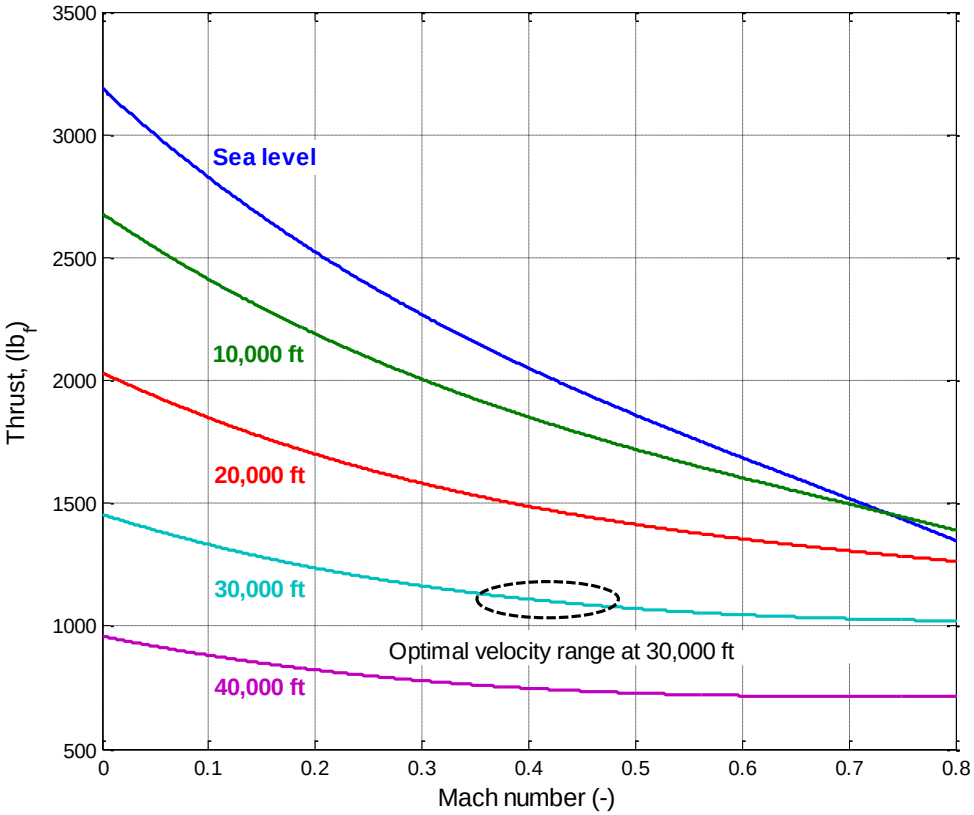

Fig. 2.8 Thrust Variation with Mach Number

As Mach number increases, the thrust decreases sharply at low altitudes and slightly at high altitudes. There are large differences in thrust at the low speeds as seen in Figure 2.8. As altitude increases, thrust goes down because the mass-flow rate ($\dot{m} = \rho_i A_i V_i$) reduces with decreasing air density (see Appendix). Cruise altitude of X-47A is 30,000 ft and optimal velocity range is between Mach 0.35 and 0.48 in this study. Therefore, approximately 1150 (lb_f) of thrust is produced by the JT15D-5C turbofan. Figure 2.9 shows *TSFC* of the engine vs altitude.

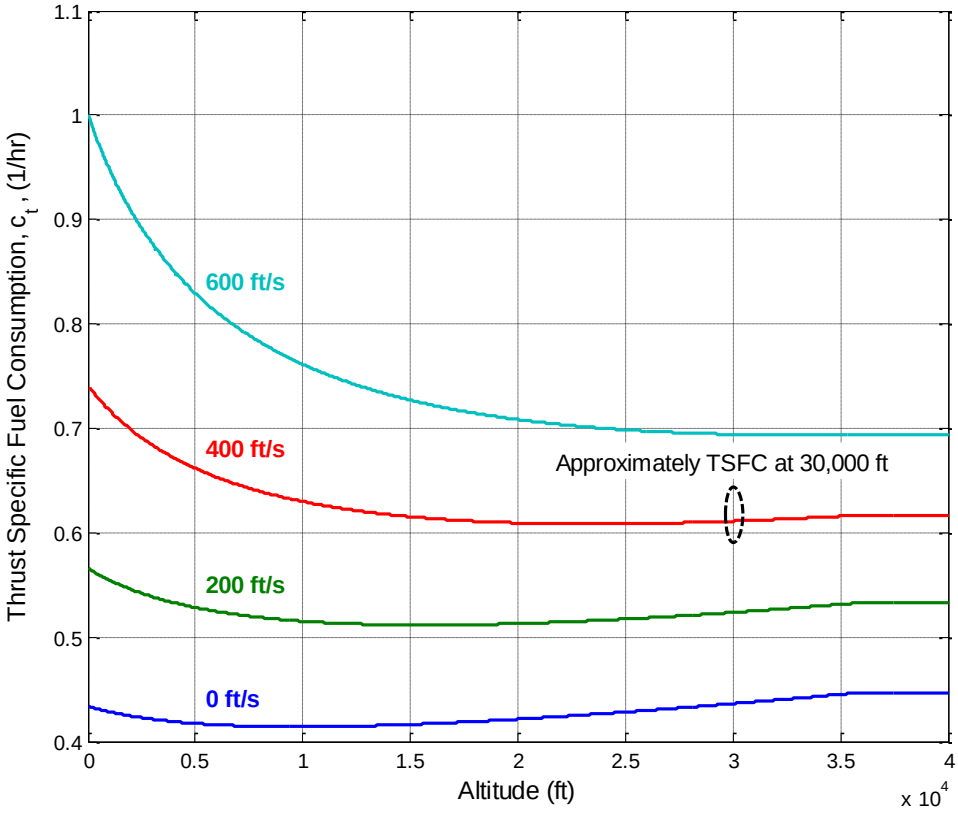

Fig. 2.9 TSFC Variation with Altitude

TSFC is poor at low altitudes with high speeds. As altitude increases for high speeds, fuel efficiency becomes much better. Although fuel efficiency of low speeds is better than high speeds as seen in Figure 2.9, thrust decreases with velocity in Figure 2.8. Velocity should be optimized in order to minimize fuel mass-flow rate ($\dot{m}_{fuel}$) as defined in equation (29). Therefore, typical TSFC range is between 0.59 and 0.64 1/hr at 30,000 ft.

$$\dot{m}_{fuel} = -c_t T \tag{29}$$

2.4 Obstacle Avoidance Model

In this study, the obstacle avoidance model is a radar system that was developed around the Second World War for air traffic control [29]. Radar is an acronym of "Radio Detection and Ranging." Radars can be categorized mainly as Primary and Secondary Surveillance Radars (SSRs). Primary surveillance radars (PSRs) are used to determine the distances of objects from radar stations. After objects are seen in the detection ranges of radars, their direction can be determined. PSRs can scan up to 220 nautical miles (nmi) in the horizontal direction and up to 80,000 ft in the vertical direction [30]. They need much more power compared to SSRs, but detection accuracy of PSRs is better than SSRs. SSRs detect location, direction and even identification of a friend aircraft thanks to the transponder. Enemy aircraft cannot be identified by SSRs because they are not cooperative. Therefore, SSRs are used commonly for civilian air traffic control.

The distance between an aircraft and a radar station can be detected by the help of electromagnetic radio signals. The radar station transmits radio waves periodically. Once a radio wave hits the aircraft, it returns to the radar station with the velocity of light. The distance can be calculated by multiplying the velocity of light times half of round trip time. Round-trip time means the time the radio signal gets to the aircraft and returns to the radio station. Hence, round-trip time should be divided by two in order to compute one-way time. Maximum detectable range (R_{max}) can be found with specifications of the radar system and the radar cross section (RCS) of the aircraft in equation (30) in Ref [31].

$$R_{max} = \left(\frac{P_t G_t G_r \sigma \lambda^2}{(4\pi)^3 P_r} \right)^{1/4} \tag{30}$$

where P_t is transmit power, P_r is received power, G_t and G_r are transmit and receive antenna gains, λ is wavelength, and σ is *RCS* which depends on material, size, and shape of the aircraft. A high *RCS* is more easily detectable than a low one; see Ref [15] for detailed information. Figure 2.10 shows a radar for detecting aircraft. Its antenna rotates steadily to scan a particular area.

Fig. 2.10 Radar for Detection Airplanes [32]

Antenna rotation rate can be tempered to the detection range of the radar due to pulse repetition rate (PRR). As PRR increases, the time between pulses goes up. That gives the opportunity to scan greater range. Therefore, the antenna should rotate slowly in order to raise the detection range [30].

CHAPTER 3 - OPTIMIZATION

3.1 Problem Definition

The objective of the optimization problem is to minimize the fuel consumption of the aircraft for cruise as seen in equation (31). Fuel consumption can be calculated by integrating the mass-flow rate, which is equal to the product of thrust and TSFC.

The problem statement is

$$\text{Minimize:} \quad F(x) = \int_{t_0}^{t_f} \dot{m}\, dt \tag{31}$$

$$\text{Subject to:} \quad \left\{ \begin{array}{ll} \dot{y}_1 = \dot{X} = V cos\Psi, & X(0) = 0 \Rightarrow \text{fixed} \\ \dot{y}_2 = \dot{Y} = V sin\Psi, & Y(0) = 0 \Rightarrow \text{fixed} \\ \dot{y}_3 = \dot{V} = \dfrac{T - D}{m}, & V(0) = V_0 \Rightarrow \text{free} \\ \dot{y}_4 = \dot{\Psi} = \dfrac{L sin\phi}{mV}, & \Psi(0) = \Psi_0 \Rightarrow \text{free} \\ \dot{y}_5 = \dot{m} = -c_t T, & m(0) = m_0 \Rightarrow \text{fixed} \end{array} \right\} \tag{32}$$

There are five state differential equations (SDEs) comprised of equations (15), (16), (21), (22), and (29) from chapter 2. The first and second SDEs are the horizontal velocities in the direction of *X* and *Y* axes. Initial position of the aircraft is fixed to the origin (0,0). The third SDE is acceleration with free initial velocity. The forth SDE is heading rate with free initial heading angle (between the direction the aircraft's velocity vector and the reference direction). The last equation is mass-flow rate. The mass of the aircraft is known at the beginning of the cruise, so initial mass is fixed.

The system has two equality equations (33) and one inequality equation (34). For equality equations, the aircraft's coordinates must be equal to the target's coordinates at the end of the cruise. For inequality equation (34), the aircraft must be outside of the radar detection range during cruise.

$$\text{Equality constraints:} \quad \begin{Bmatrix} c_{eq}(1) = X(t_f) - X_{target} = 0 \\ c_{eq}(2) = Y(t_f) - Y_{target} = 0 \end{Bmatrix} \tag{33}$$

$$\text{Inequality constraint:} \quad \left\{ c = \sqrt{(X(t) - X_{radar})^2 + (Y(t) - Y_{radar})^2} \geq r_{radar}^2 \right\} \tag{34}$$

3.2 Fmincon Algorithm

Fmincon is one of the MATLAB optimization solvers. It finds the local minimum value of a constrained nonlinear multivariable function [33]. The syntax of the Fmincon algorithm is

```
x = fmincon (fun, x0, A, b, Aeq, Beq, lb, ub, nonlcon)
```

where `x` is the solution of the optimization; `fun` is a function M-file which assesses objective function value; `x0` consists of initial estimations of the variables to start the optimization; `A` and `b` are linear inequality constraints matrix and vector, respectively; A_{eq} and B_{eq} are linear equality constraints matrix and vector, respectively; lb and ub are lower and upper bounds for the variables, and `nonlcon` is a nonlinear constraint function M-file which includes constraint equations.

In this study, there are no linear constraints, so `A`, `b`, A_{eq}, and B_{eq} are empty brackets, [ ]. The system has lower and upper bounds for throttle, bank angle, and velocity. Therefore, the syntax should be modified as:

```
x = fmincon (fun, x0, [], [], [], [], lb, ub, nonlcon)
```

The goal of the fmincon algorithm is to minimize the objective function value. The objective function needs to have an integration method to solve SDEs. Two different methods are used to integrate SDEs, which are “Initial Value Problem” and “Inverse dynamics.”

3.2.1 Objective Function

3.2.1.1 Method 1 – Initial Value Problem (IVP)

The ode45 solver in MATLAB is employed to integrate the SDEs. It is based on the Runge-Kutta method [34]. The syntax of the ode45 solver is defined as:

```
[t, y] = ode45(odefun, tspan, y0)
```

where `t` is evaluation points on time span, `y` is solution of state variables, `odefun` is a function M-file which determines the right-hand sides of the SDEs, `tspan` is interval of integration `[t0 tf]`, `y0` is initial condition of state variables.

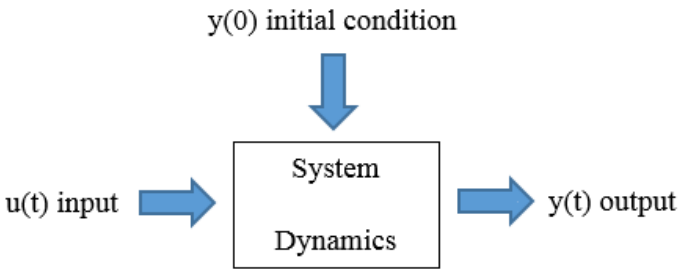

Fig. 3.1 System Dynamics for IVP

The initial conditions and inputs of the dynamic system are known. The system has two inputs (throttle and bank angle) and five outputs (position coordinates X and Y, velocity,

heading angle, and mass) as shown in Figure 3.2. Their initial conditions are fixed except initial velocity and initial heading angle.

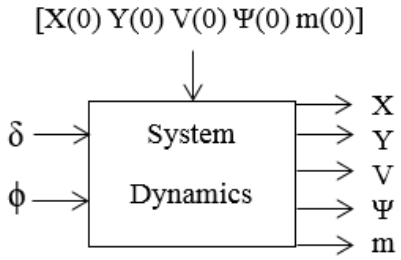

Fig. 3.2 Parameters of System Dynamics for IVP

Each control variable has *N* discrete values along the trajectory as shown in Figures 3.3 and 3.4. Also, the system needs the scaling factors in order to determine initial free variables (velocity and heading angle) and flight time. Initial velocity, initial heading angle, and flight time are computed by multiplying scaling factors by estimated values in equations (35) – (37)

$$V_{initial} = a_{velocity} V_{estimated} \tag{35}$$

$$\Psi_{initial} = a_{heading\_angle} \Psi_{estimated} \tag{36}$$

$$t_{flight} = a_{flight\_time} t_{estimated} \tag{37}$$

The total number of variable is

$$N(\delta) + N(\Phi) + a_{velocity} + a_{heading\_angle} + a_{flight\_time} = 2N+3 \text{ variables}$$

There are upper and lower bounds for two control variables. In Figure 3.3, the upper bound is one which corresponds to 100% throttle and maximum engine thrust at a given altitude and velocity. The lower bound is zero, which corresponds to 0% throttle. In Figure 3.4, absolute values of the lower and upper bounds for bank angle are equal to

each other. The bounds depend on maximum thrust available, weight of the aircraft, and maximum lift-over-drag ratio. As weight decreases, bank-angle boundary range increases with stall speed for level turn. Maximum bank (ϕ_{max}) is derived from equation (19):

$$\phi_{max} = \cos^{-1}\left(\frac{1}{n_{max}}\right) \tag{38}$$

where n_{max} is maximum load factor at $(L/D)_{max}$ that is computed by using equation (39).

$$n_{max} = \left(\frac{L}{D}\right)_{max} \frac{T_{max}}{W} \tag{39}$$

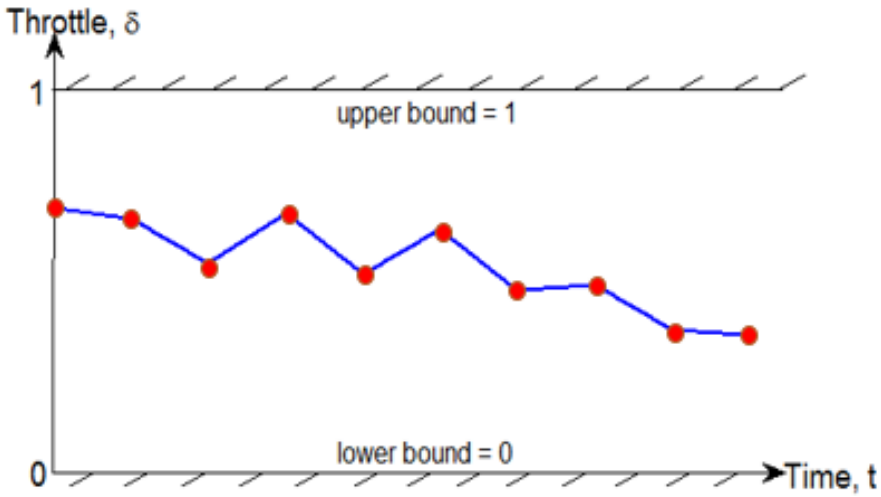

Fig. 3.3 Throttle vs Time throughout Cruise

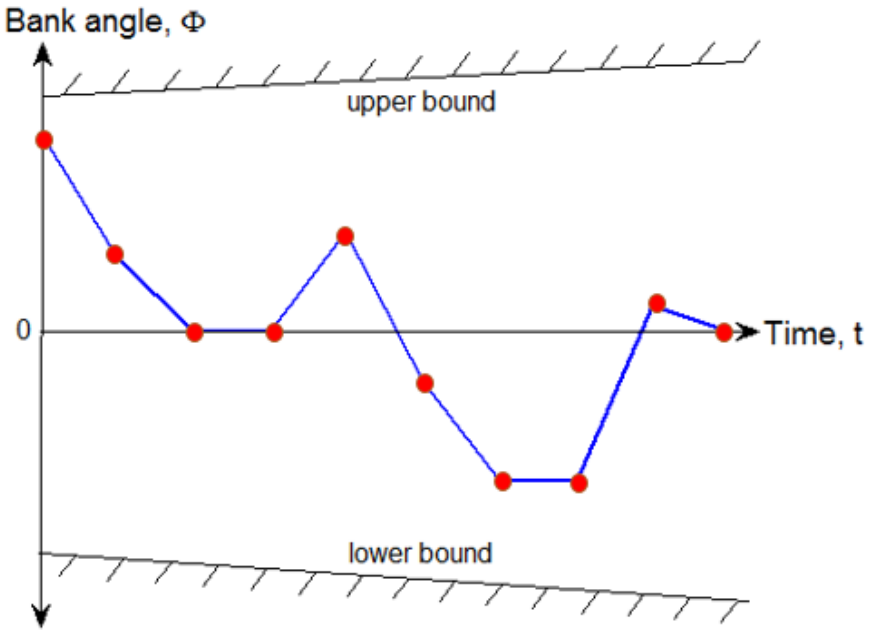

Fig. 3.4 Bank Angle vs Time throughout Cruise

3.2.1.2 Method 2 – Inverse Dynamics

Discrete state variables are the inputs for an inverse-dynamics approach as shown in Figure 3.5. SDEs can be obtained by using required state variables, and then control, u(t), can be determined from the state rates.

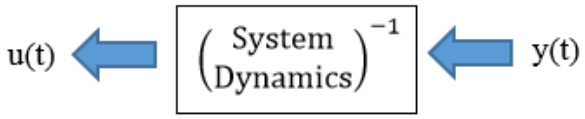

Fig. 3.5 System Dynamics for Inverse-Dynamics

For velocity and heading angle, *N* discrete values defined the trajectory as shown in Figures 3.6 and 3.7. The slopes of the segments between the stages in Figures 3.6 and 3.7 indicate acceleration ($\dot{V}$) and heading rate ($\dot{\Psi}$), respectively. Kinematic equations, ($\dot{X}$) and ($\dot{Y}$), can be obtained using the velocity and heading angle values. Cumulative trapezoidal numerical integration (cumtrapz) method in MATLAB is implemented to integrate the

first and second SDEs to determine the X and Y coordinates. The method calculates the area under the horizontal velocity curves as shown in Figures 3.8 and 3.9.

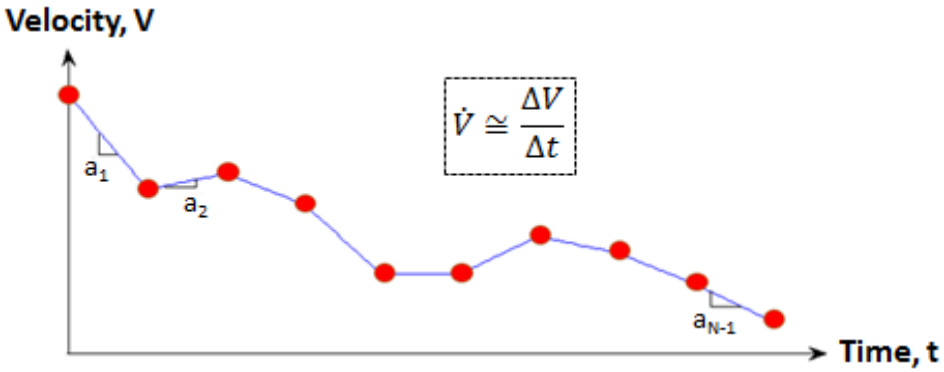

Fig. 3.6 Velocity vs Time throughout Cruise

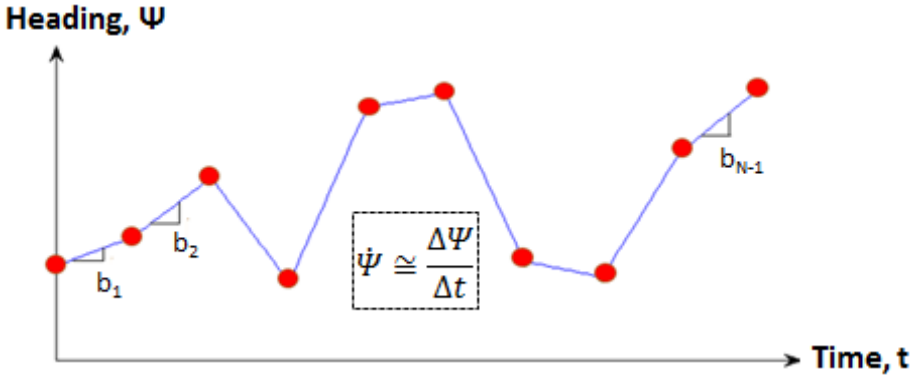

Fig. 3.7 Heading Angle vs Time throughout Cruise

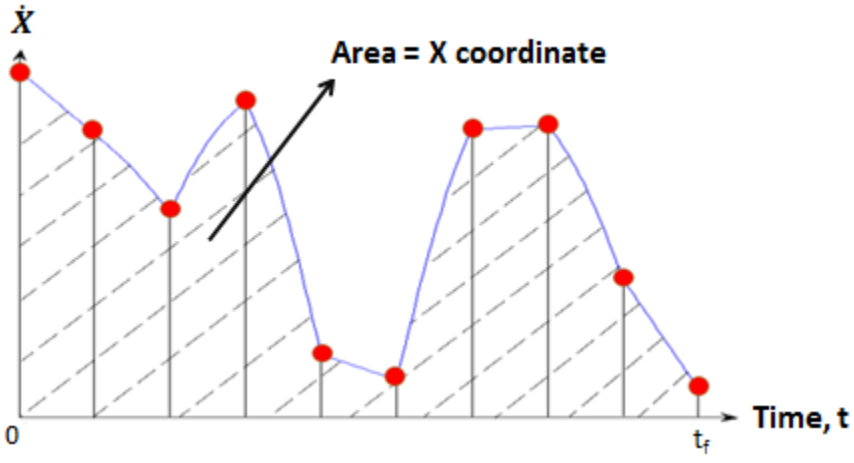

Fig. 3.8 X-Coordinate Horizontal Velocity vs Time throughout Cruise

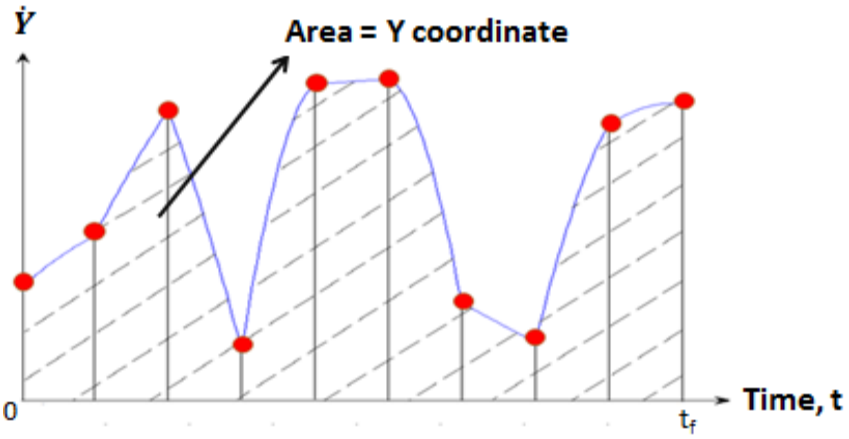

Fig. 3.9 Y-Coordinate Horizontal Velocity vs Time throughout Cruise

The last SDE, mass-flow rate ($\dot{m} = -c_t T$), can be solved with cumtrapz method but the equation needs thrust required. Therefore, thrust required must be calculated in the third SDE by using the mass of the aircraft at each stage:

$$\dot{V} = \frac{\Delta V}{\Delta t} = \frac{T - D}{m} \Rightarrow T = m\frac{\Delta V}{\Delta t} + D \tag{40}$$

Initial mass of the aircraft is known, so there are *N*-1 mass variables and *N*-1 equality constraints as shown in equation (41). Input mass variables at each stage must be equal to the amount of the initial mass of the aircraft minus total fuel consumption, which is until that stage:

$$\begin{aligned} m(2) &= m(1) - m_{fuel}(1) \\ m(3) &= m(1) - (m_{fuel}(1) + m_{fuel}(2)) \\ &\vdots \qquad\quad \vdots \qquad\quad \vdots \qquad\quad \vdots \\ m(N) &= m(1) - \left(m_{fuel}(1) + m_{fuel}(2) + \cdots + m_{fuel}(N-1)\right) \end{aligned} \tag{41}$$

In the inverse-dynamics method, only flight-time scaling factor is used as a variable because initial velocity and initial heading angle are defined by *N* discrete values. The total number of variables is:

$$N(\mathrm{V}) + N(\Psi) + N\text{-}1(\mathrm{m}) + a_{flight\_time} = 3N \text{ variables}$$

When an optimal path-planning problem has obstacles, solving the problem becomes more difficult. It is difficult to determine optimal heading angle. Therefore, first the shortest path should be found to minimize range and fuel consumption. Then a geometric approach should be used to estimate heading angles for the flight mechanics model.

3.2.1.2.1 Shortest Path

Fuel consumption of the aircraft and range are directly proportional. As range between initial and destination points decreases, the aircraft consumes less fuel. In Figure 3.10, there are several paths to go from point A to point B by avoiding three circular obstacles (radars); hence, the shortest path should be determined to obtain the smallest range and least fuel consumption.

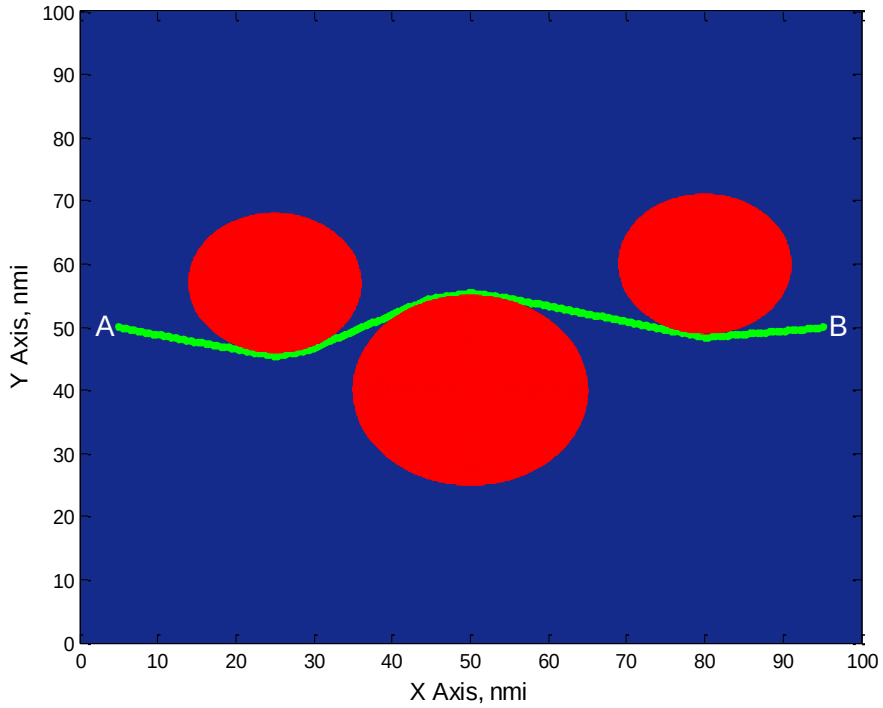

Fig. 3.10 Shortest Path by Avoiding Obstacles without Flight Mechanics

The locations and detection ranges of the three radars are known in advance. Figure 3.10 is composed of a grid matrix where the blue area is represented by zeros and the red area (radar) is represented by ones. The shortest path is found as the green curve by applying 'shpath' code in MATLAB [35]. The code has five inputs: grid matrix (zeros and ones), initial and destination coordinates (X_i, Y_i, X_f, and Y_f). The outputs of the code are the X and Y way-points of the path. Point A and B must be on the blue area to start searching the shortest path. Then the path follows zeros to reach point B.

3.2.1.2.2 Geometric Model

A geometric model is utilized to estimate heading angle at each discrete stage for flight mechanics model. The Fmincon optimization algorithm is used to minimize the length of the total segments by avoiding the radars as shown in equation (42) and Figure 3.11. The optimization parameters are the *X* and *Y* values without regard for flight mechanics equations of motion.

$$\text{Minimize:} \quad F(x) = \sum_{n=1}^{N} \sqrt{(X(n+1) - X(n))^2 + (Y(n+1) - Y(n))^2} \qquad (42)$$

N+1 stages are chosen around the shortest path in Figure 3.10 to get *N* linear segments and *N* slopes of these segments. Each slope is used to develop guesses of the heading angle variables for flight mechanics model. The total number of free variables is:

$$N+1(\text{X}) + N+1(\text{Y}) = 2N+2 \text{ variables}$$

There are *N*-1 equality constraint equations because the lengths of the each segment should be equal in order to gain more accurate results for flight mechanics model. In addition, three inequality constraint equations are used to avoid the three radars.

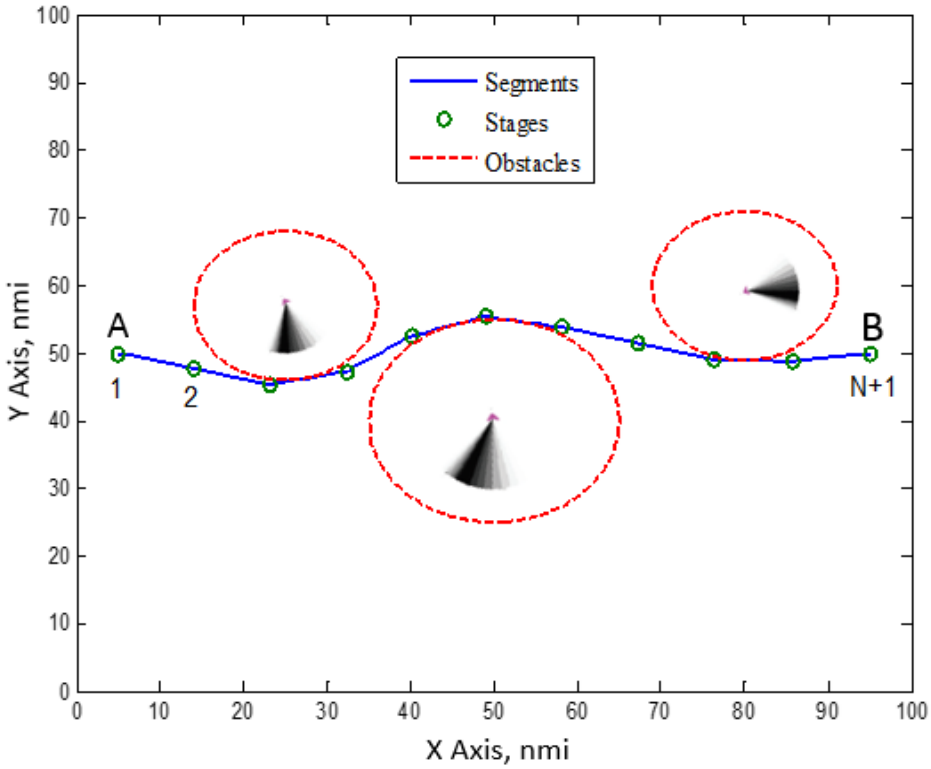

Fig. 3.11 Geometric Approach without Flight Mechanics

CHAPTER 4 – NUMERICAL RESULTS

4.1 Variable Cruise Altitude vs. Constant Cruise Altitude

In this section, the fuel consumption is compared between variable and constant cruise altitudes for the same range. They are investigated separately as case 1 (variable cruise altitude) and case 2 (constant cruise altitude).

For case 1, first the aircraft glides from 30,000 ft to 20,000 ft. Then it flies for a short time at 20,000 ft with 100% throttle to achieve the appropriate velocity for maximum rate-of-climb (RC_{max}). After it reaches the velocity, it climbs from 20,000 ft to 30,000 ft. Case 1 can be separated into three phases: descent, constant-altitude, and ascent phases. The fuel consumption of case 1 is

$$m_{f\_total\_case1} = m_{f\_descent} + m_{f\_constant-altitude} + m_{f\_ascent} \qquad (43)$$

In the descent phase, the fuel consumption is zero because it is gliding flight, so the total fuel consumption of case 1 is the product of the constant-altitude and the ascent phases.

For case 2, the flight altitude is fixed at 30,000 ft. The aircraft flies the same range in order to compare the amount of fuel consumptions of case 1 and case 2. In Table 4.1, the total ranges of case 1 and case 2 are same at the end of cruise. The longest range phase is the descent phase because of glide at $(L/D)_{max}$.

Table 4.1: Total Ranges of Case1 and Case 2

Case	Descent phase (nmi)	Constant-altitude phase (nmi)	Ascent phase (nmi)	Total range (nmi)
1	19.24	0.89	12.15	32.28
2	-	32.28	-	32.28

Figure 4.1 shows cruise altitude vs. range. The initial velocities and the flight-path angles for both cases are 426 ft/s and 0 degree, respectively, in Figure 4.1. When the aircraft begins the unpowered glide for case 1, the velocity is too high for equilibrium gliding flight. Therefore, the aircraft climbs by increasing lift coefficient (C_L). Damped oscillatory motion occurs until the FPA reaches steady state with the shallowest gliding angle.

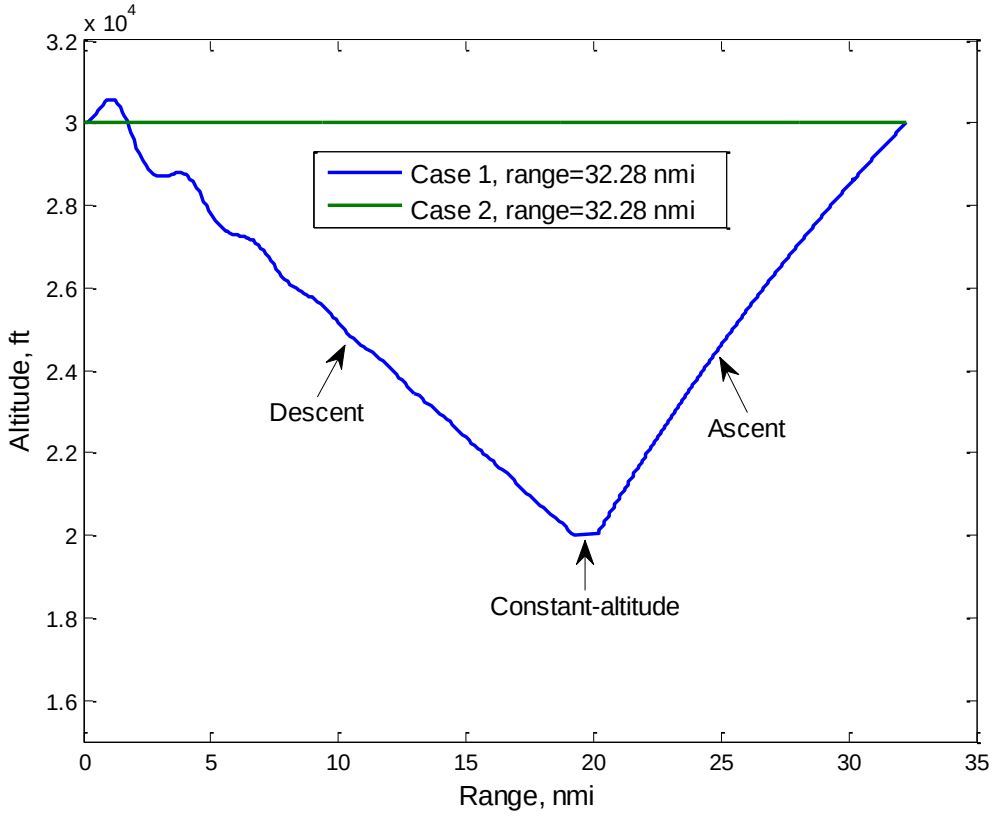

Fig. 4.1 Altitude vs. Range for Case 1 and Case 2

Figure 4.2 presents the weight variance of the aircraft with range. While initial weights of the aircraft for both cases are the same, the weight difference between case 1 and case 2 at the end of cruise is only 2.1 lb_f. For case 1, the weight in the descent phase remains constant because the engine does not produce power from 30,000 ft until 20,000 ft. After

then the weight decreases sharply in the constant phase with maximum throttle. The fuel consumption of the constant-altitude phase is 3.7 lb_f. Most fuel in the case 1 is consumed during the ascent phase. The aircraft burns 38.5 lb_f of fuel while climbing from 20,000 ft to 30,000 ft. Total fuel consumption for case 1 is 42.2 lb_f. For case 2, Fmincon optimization algorithm is used to minimize the fuel consumption for case 2. The inverse-dynamics method is applied to solve SDEs. Total fuel consumptions for case 1 is 40.1 lb_f. The weight profile changes towards the end of case 2 because the last discrete velocity value for case 2 decreases sharply.

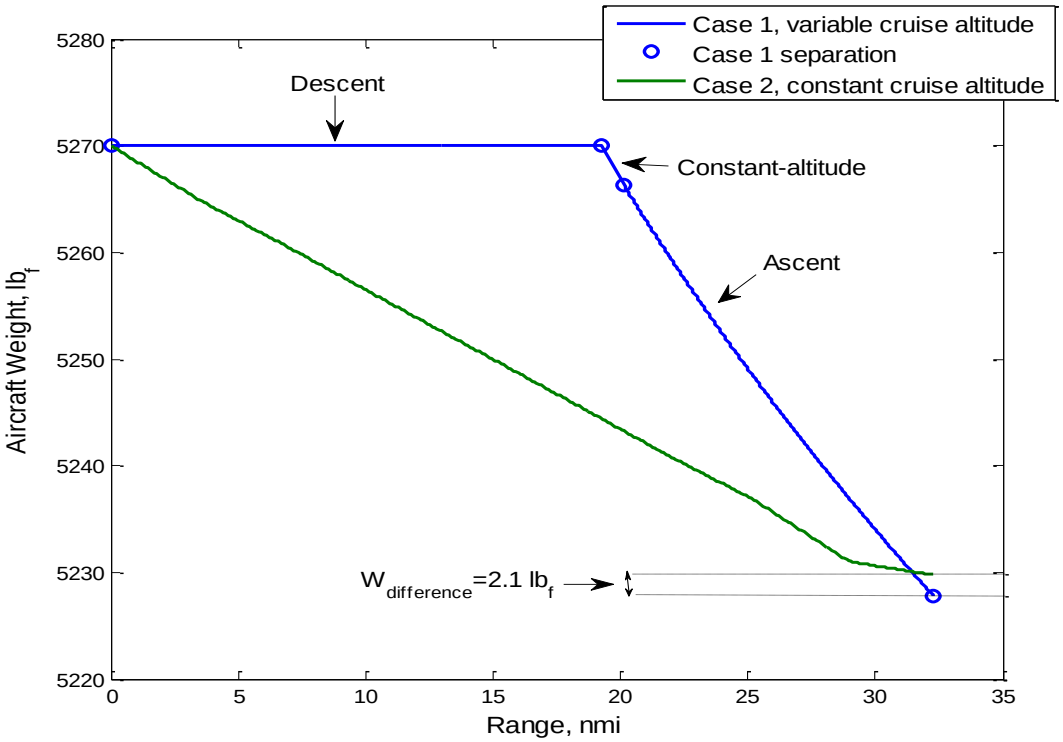

Fig. 4.2 Aircraft Weight vs. Range for Cases 1 and Case 2

The comparison of the fuel consumption for cases 1 and 2 is made to determine if gliding helps fuel consumption. Constant cruise altitude (case 2) is implemented for the remainder of numerical calculations in this thesis because case 2 is more fuel-efficient.

4.2 Optimal Path without Obstacles

In this section, the optimal path is determined by IVP (method 1) and inverse dynamics (method 2). Figure 4.3 shows the optimal path without obstacle avoidance for flight at 30,000 ft. The starting point of the cruise is the origin (0,0) and the target is (700,700 nmi). There is no obstacle, so the path should be a straight line to minimize range and fuel consumption. Ten discrete stages are chosen on the path for methods 1 and 2.

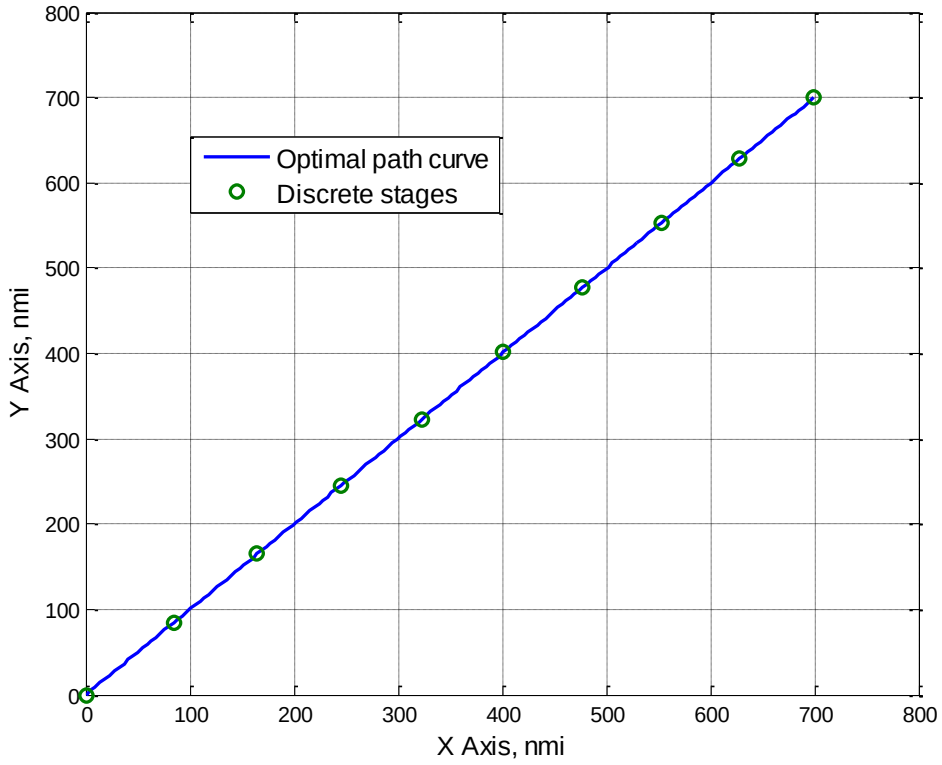

Fig. 4.3 Optimal Path Planning without Obstacles at 30,000 ft

For method 1, there are 10 throttle and 10 bank angle variables. Also, three scaling factors are necessary for initial velocity, initial heading angle, and flight time. The total number of variables is:

$$10(\delta) + 10(\Phi) + a_{velocity} + a_{heading\_angle} + a_{flight\_time} = 23 \text{ variables}$$

The initial guesses for initial velocity, initial heading angle, and flight time are taken as 450 ft/s, 45 deg, and 15,000 sec, respectively. The scaling factors are found as 0.947, 1, and 0.956, respectively. Initial velocity, initial heading angle, and flight time can be computed by using equations (35) - (37):

$$V_{initial} = a_{velocity} V_{estimated} \Leftrightarrow (0.947)(450) = 426.15\ ft/s \quad (35)$$

$$\Psi_{initial} = a_{heading\_angle} \Psi_{estimated} \Leftrightarrow (1)(45) = 45\ deg \quad (36)$$

$$t_{flight} = a_{flight\_time} t_{estimated} \Leftrightarrow (0.956)(15000) = 14{,}340\ sec \quad (37)$$

Figure 4.4 shows optimization results of the throttle variables vs.time. The throttle control variables at each stage are in the range between 0.381 and 0.651. Figure 4.5 shows that the bank angle control variables are equal to zero at each stage. Since they are equal to zero, turning flight and the aircraft flies with 45 degree of heading angle during cruise. This is an expected result because there is no obstacle, so the path should be straight.

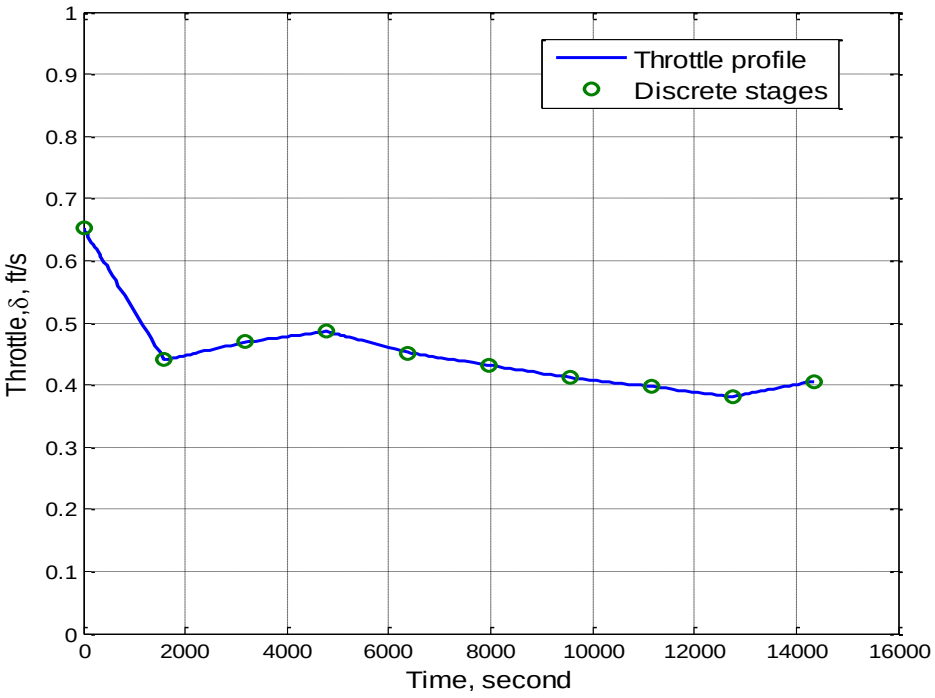

Fig. 4.4 Optimal Throttle Profile Using IVP Method (No obstacles)

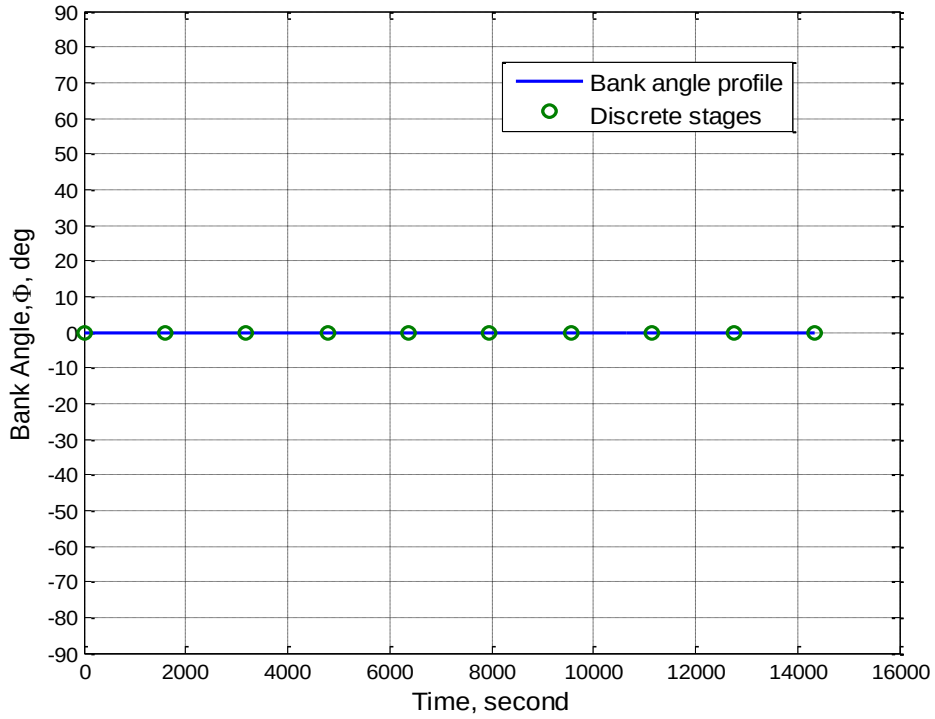

Fig. 4.5 Optimal Bank Angle Profile Using IVP Method (No obstacles)

For method 2 (inverse-dynamics), the same number of stages (10) is taken along the path. Each stage takes into account velocity, heading angle, and mass variables except initial mass of the aircraft because it is fixed to 163.797 slugs. The flight-time scaling factor is an additional optimization variable. Flight time is computed by applying equation (37). The total number of variables is:

$$10(\mathrm{V}) + 10(\Psi) + 9(\mathrm{m}) + a_{flight\_time} = 30 \text{ variables}$$

$$t_{flight} = a_{flight\_time} t_{estimated} \Leftrightarrow (0.9665)(15{,}000) = 14{,}497\ sec \qquad (37)$$

Figure 4.6 shows that velocity decreases with time. According to equation (2), lift is equal to weight for straight, level flight ($\gamma = 0$):

$$mV_\infty \dot{\gamma} = L - W\cos\gamma \ \Rightarrow W = L = \frac{1}{2}\rho_\infty V_\infty^2 S C_L \qquad (2)$$

As the weight decreases with time because of fuel consumption, lift decreases. In straight, level flight, the air density remains constant, so the velocity tends to diminish.

Figure 4.7 shows that heading-angle state variables remain constant at 45 degrees. The aircraft flies directly from (0,0) coordinates to (700,700 nmi) coordinates as shown in Figure 4.3. Figure 4.8 shows the mass profile with time.

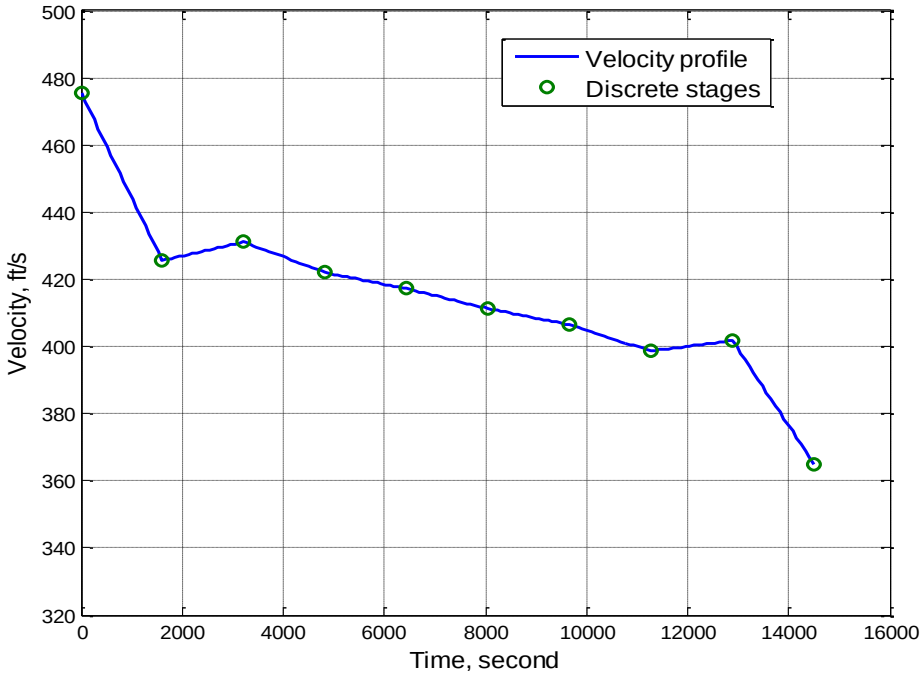

Fig. 4.6 Optimal Velocity Profile Using Inverse Dynamics Method (No obstacles)

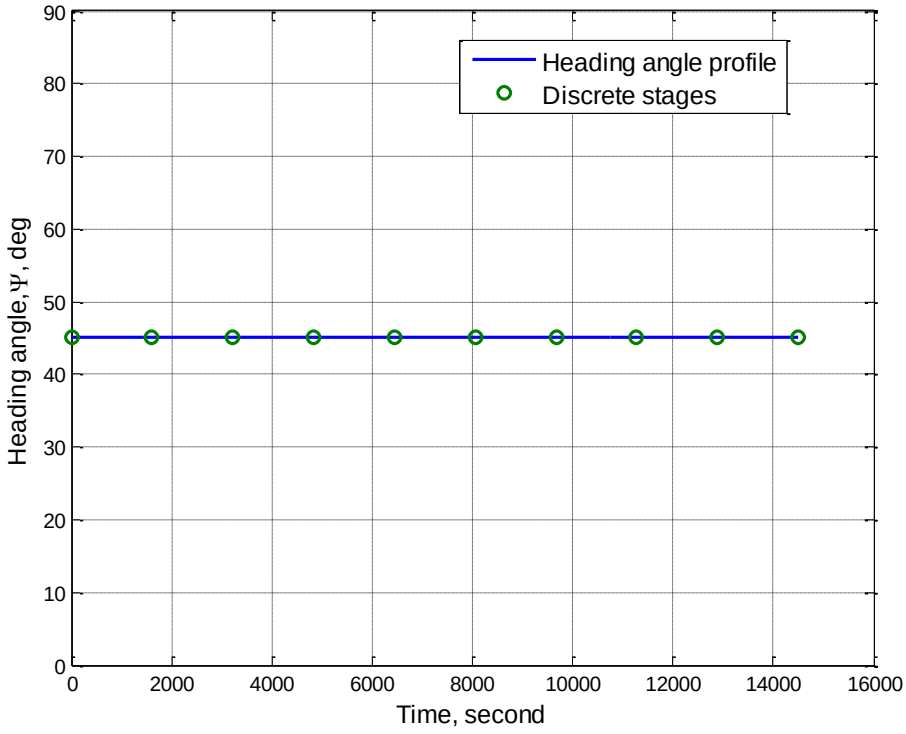

Fig. 4.7 Optimal Heading Angle Profile Using Inverse Dynamics Method (No obstacles)

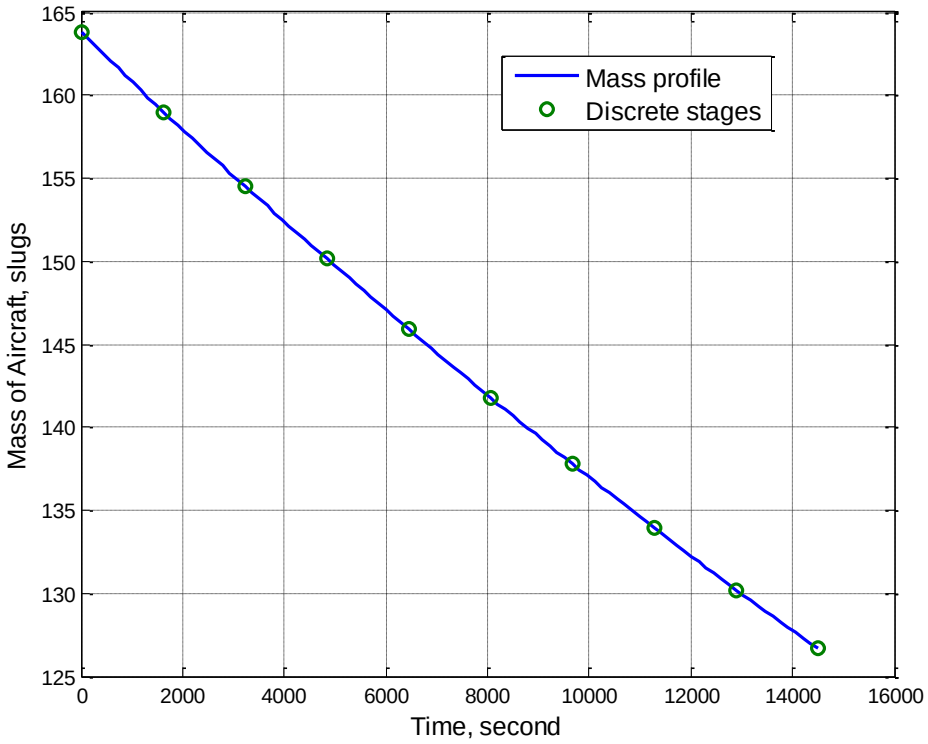

Fig. 4.8 Optimal Mass Profile Using Inverse Dynamics Method (No obstacles)

Table 4.2 shows that the amount of the fuel consumption at segments decreases with time thanks to lower mass.

Table 4.2: Fuel Consumption at Segments

Segment number	Fuel consumption (slug)
1	4.79
2	4.50
3	4.39
4	4.24
5	4.11
6	3.98
7	3.85
8	3.76
9	3.52

Table 4.3 compares the numerical results for IVP and inverse dynamics methods. The number of the stages and the total range are the same for both methods, but the inverse dynamics method has seven more variables than the IVP method. Although the fuel consumption of IVP (37.38 slugs) and inverse dynamics (37.14 slugs) are very similar, the inverse dynamics method is approximately four times faster than the IVP method. Therefore, the inverse dynamics method is used for the remainder of the optimization problems in this thesis.

Table 4.3: Optimization Results of IVP and Inverse Dynamics Methods

Method	Number of stages	Number of optimization variables	Computation time (sec)	Minimized fuel consumption (slug)	Total range (nmi)
IVP	10	23	70	37.38	989.95
Inverse Dynamics	10	30	16.8	37.14	989.95

4.3 Optimal Paths with Obstacle Avoidance

The procedure for determining the optimal path by avoiding obstacle differs from the procedure of the optimal path-planning without obstacles. The shortest path and geometric model are considered in this section. Two obstacle-avoidance scenarios are investigated. The first scenario is a one-way path with one destination point, the second scenario is a closed path with three destination points.

4.3.1 Scenario 1 – One-Way Path with One Destination Point

In this scenario, the starting point (0,0) and target destination point (700,700 nmi) are chosen. Sixteen primary surveillance radars (PSRs) are added for scenario 1. The location and detection ranges of these radars are given in Table 4.4.

Table 4.4: Coordinates and Detection Ranges of 16 PSRs for Scenario 1

Radar number	Coordinates radar center (nmi)	Detection range (nmi)
1	(100,130)	70
2	(190,500)	50
3	(200,370)	70
4	(200,250)	50
5	(300,130)	70
6	(320,300)	70
7	(320,450)	50
8	(440,550)	70
9	(480,400)	70
10	(460,250)	70
11	(500,100)	70
12	(590,250)	70
13	(600,460)	70
14	(600,620)	70
15	(700,370)	70
16	(700,130)	50

Figure 4.9 is composed of a grid matrix (800x800 nmi) where the blue areas indicate the safe flight zone and the 16 red color circles indicate PSRs obstacles. The ‘shpath’ code (see Section 3.2.1.2.1) is implemented in order to find the shortest path by avoiding 16 PSRs in Figure 4.9. After the shortest path is found, 11 discrete stages are taken around the shortest path for the geometric model. The stages refer to (X_i,Y_i) coordinates. The total number of variables is 22. The eleven stages are connected by 10 linear segments. The total length of the 10 segments is minimized by using geometric model (see Section 3.2.1.2.2).

The shortest path step can be applied in order to find a global optimal path when the detection areas of the radars are very large and these radars are not close to each other. On the other hand, the shortest path step could provide a local or global optimal path when detection areas of the radars are small and these radars are close to each other. This is because the aircraft could have a sharp turning flight to avoid smaller obstacles.

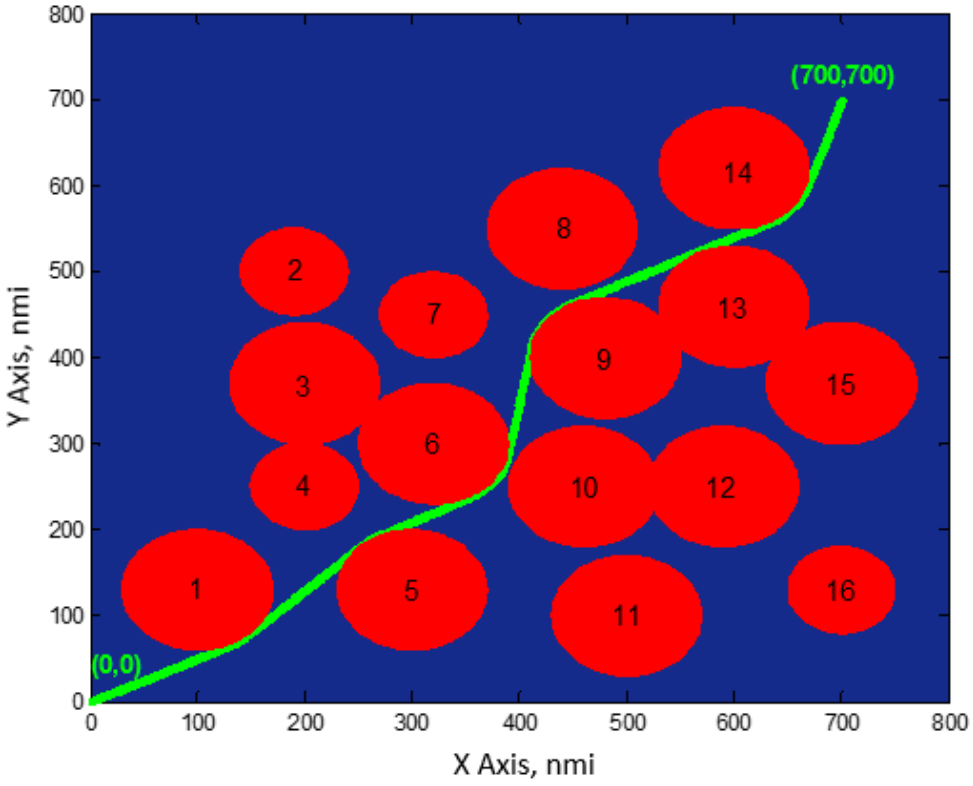

Fig. 4.9 Shortest Path by Avoiding 16 PSRs without Flight Mechanics

Figure 4.10 shows optimization results of the geometric model. The total length of the 10 segments is 1,084.6 nmi. Each segment slope represents an initial estimate of the heading angle for the flight mechanics model. Table 4.5 shows the optimized slopes, which are obtained from the minimized-path for geometric model solution.

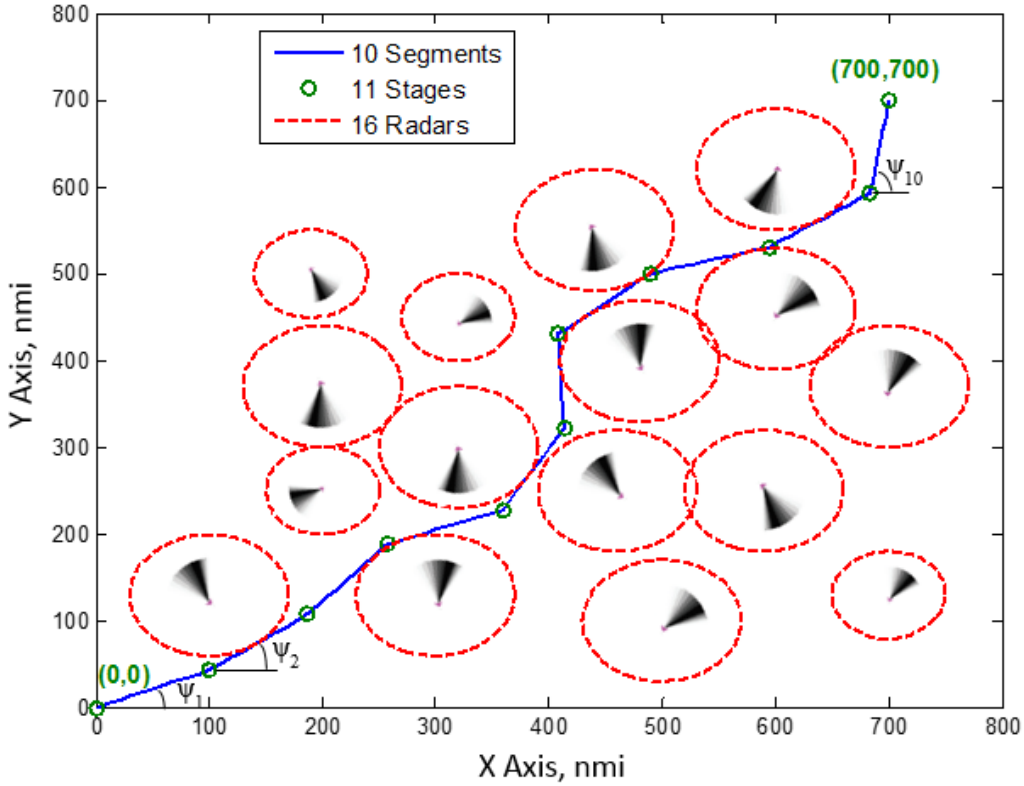

Fig. 4.10 Geometric Model without Flight Mechanics

Table 4.5: Optimized Slope from the Geometric Model Solution

Segment number	Slope (deg)
1	23.1
2	36.6
3	48.8
4	21.2
5	60.2
6	92.9
7	40.5
8	16.2
9	34.6
10	81.4

Reasonable 10 velocity, 9 mass, and a flight-time scaling variables should be estimated in order to use for flight mechanics optimization with 10 heading-angle variables, which are obtained from Table 4.5. These 30 initial guesses are used to minimize the fuel consumption for the optimization problem (see equations (31) – (34)). All local optimal paths are found before determining the global optimal path. The blue curve represents the global optimal path planning by avoiding 16 PSRs in Figure 4.11.

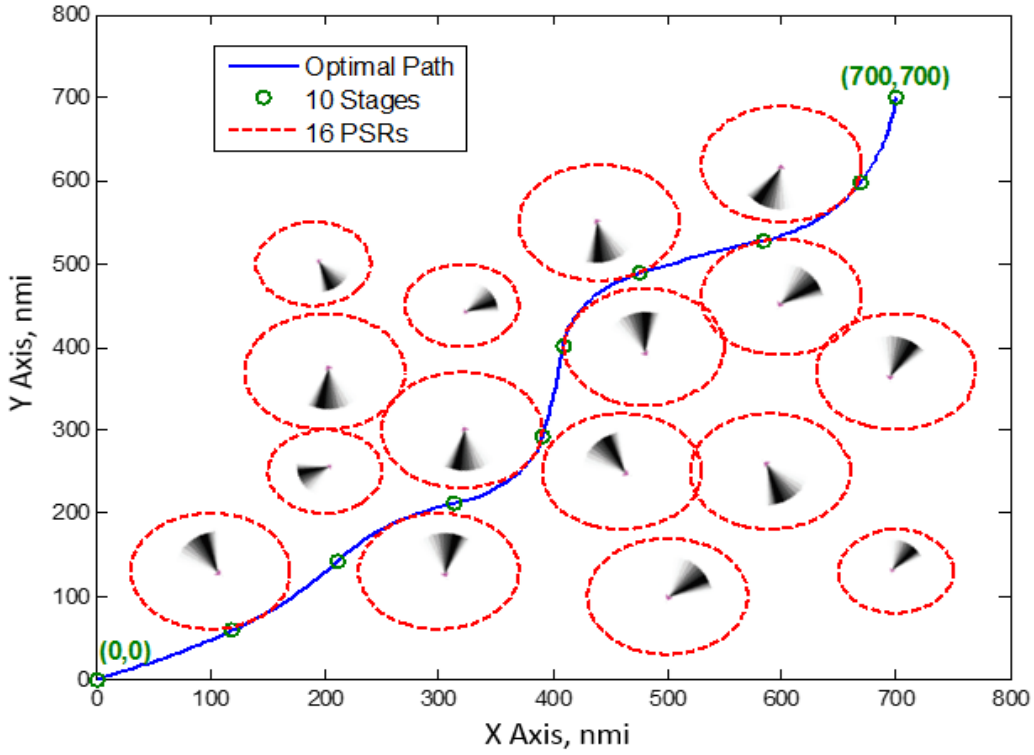

Fig. 4.11 Optimal Path Planning for Scenario 1 for Cruise at 30,000 ft

Optimal velocity, heading angle, and mass values, which are obtained from flight mechanics optimization results, are given in Table 4.6. Optimal heading angles at each stages are similar to the optimized slopes from the geometric minimum-length optimization. This means that the geometric model provides a good estimate for the flight mechanics of the optimization problem.

Table 4.6: Optimization Results of Velocity, Heading Angle, and Mass Variables

Stage number	Optimal Velocity Values (ft/s)	Optimal Heading Angle Values (deg)	Optimal Mass Values (slugs)
1	484.35	21.1	-
2	432.94	31.5	158.46
3	441.53	52.7	153.43
4	424.72	15.3	148.49
5	386.51	77.9	143.76
6	386.30	82.9	139.59
7	406.95	22.8	135.16
8	398.33	17.3	131.03
9	381.17	62.1	126.98
10	360.41	84.4	123.29

There are 29 optimized variables in Table 4.6. The last optimized variable is the flight-time scaling factor of 1.0524. The flight time is found by using equation (37):

$$t_{flight} = a_{flight\_time} t_{estimated} \Leftrightarrow (1.0524)(15{,}000) = 15{,}786\ sec \qquad (37)$$

Table 4.7 shows a comparison of two optimization problems for 10 and 20 discrete stages. As the number of stages on the path increases from 10 to 20 stages, fuel efficiency becomes 0.8% better because the aircraft can turn more effectively with 20 stages in order to avoid 16 PSRs. The extra degrees of freedom provides the opportunity to reduce the total range, and total range of the second optimization is 5.9 nmi shorter than total range of the first optimization. On the other hand, the computation time of the second optimization is much greater than the first optimization.

Table 4.7: Optimization Results for 10 and 20 Discrete Stages at 30,000 ft

Number of stages	Number of variables	Computation time (min)	Minimized fuel consumption (slugs)	Total Range (nmi)
10	30	15.3	40.51	1,062.6
20	60	50.8	40.18	1,056.7

4.3.1.1 Varying Cruise Altitude

Three optimal paths are investigated at three different cruise altitudes of 30,000, 35,000, and 40,000 ft. All other properties of the optimization problem are the same as scenario 1. As altitude increases, the air density decreases. Figure 4.12 shows that the average cruise speed increases with altitude. The higher velocity reduces the flight time. While the flight time is 15,786 sec (4.39 hr) at 30,000 ft, the flight time becomes 12,906 sec (3.59 hr) at 40,000 ft.

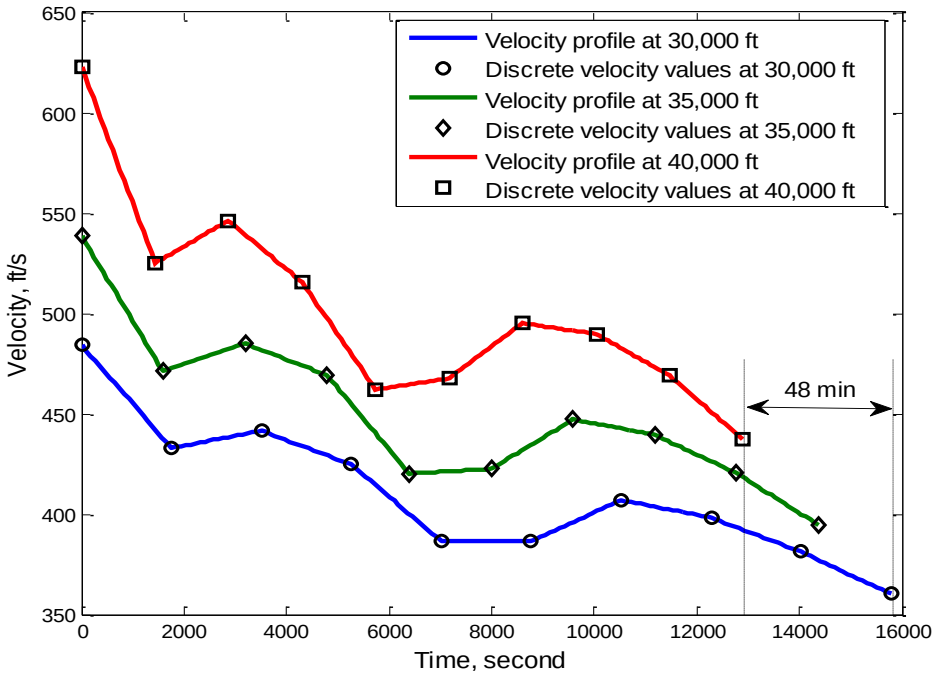

Fig. 4.12 Optimal Velocity Values by Avoiding 16 PSRs at Varying Cruise Altitudes

The aircraft mass profiles at different cruise altitudes are depicted in Figure 4.13. The initial cruise masses of the aircraft are not same because the aircraft consumes more fuel to climb from sea level to 40,000 ft than from sea level to 30,000 ft. Therefore, the initial mass of the aircraft at 40,000ft is less than the initial mass of the aircraft at 30,000 ft. The aircraft consumes less fuel when cruising at higher altitudes because of lower air density. The minimized fuel consumption at 40,000 ft is almost 11% less than the minimized fuel consumption at 30,000 ft.

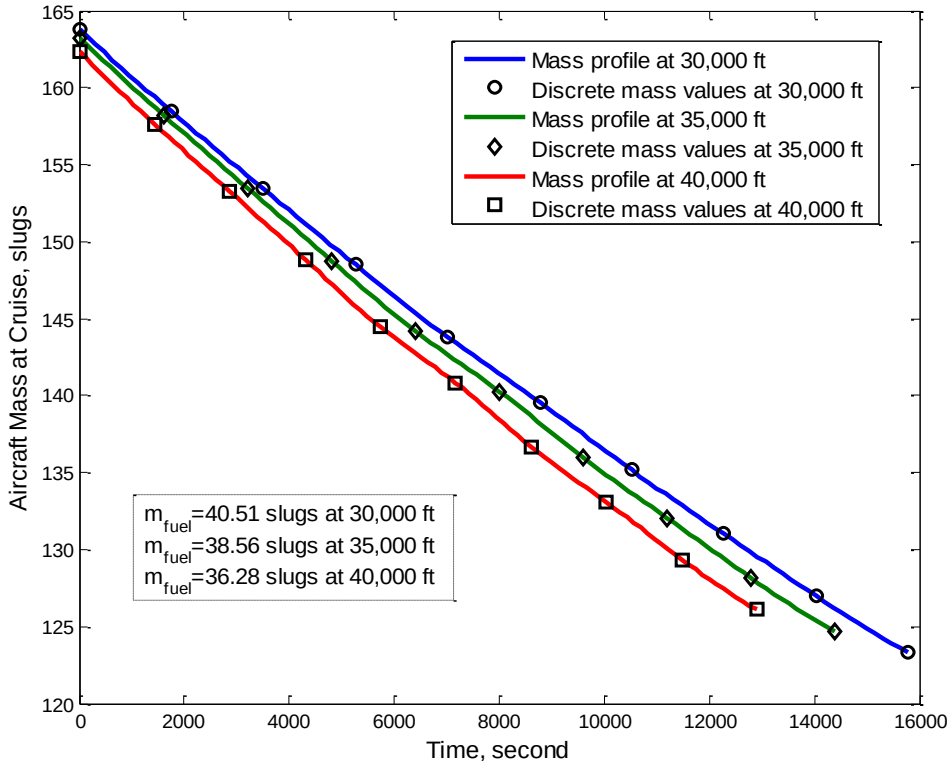

Fig. 4.13 Minimized Fuel Consumption of the Aircraft at Different Cruise Altitudes

4.3.2 Scenario 2 – Closed-Circuit Path with Three Destination Points

In this section, optimal closed-circuit path planning is investigated. For scenario 2, the starting and final destination coordinates are the origin (0,0). Two other fixed way-point

destination coordinates are given as (300,100 nmi) and (100,300 nmi). The mission scenario states that the aircraft must fly over these coordinates while avoiding 16 PSRs. The detection ranges and locations of 16 PSRs are listed in Table 4.8.

Table 4.8: Coordinates and Detection Ranges of 16 PSRs for Scenario 2

Radar number	Coordinates radar center (nmi)	Detection range (nmi)
1	(-50,150)	50
2	(60,340)	40
3	(20,230)	50
4	(40,70)	50
5	(70,-50)	50
6	(140,30)	40
7	(120,130)	50
8	(120,220)	40
9	(160,300)	40
10	(240,290)	50
11	(260,170)	40
12	(220,100)	50
13	(230,-10)	50
14	(310,30)	50
15	(350,140)	40
16	(320,230)	40

The same procedure is followed as scenario 1. This means that the optimization of scenario 2 is applied for flight mechanics model after the shortest path and the geometric model are determined.

In Figure 4.14, the starting coordinates of the path is point A (0,0) and first way-point destination coordinates is point B (300,100 nmi). Once the aircraft flies over the first destination way-point, it flies northwest to reach way-point C (100,300 nmi) and then it returns to the starting coordinates A (0,0). The minimized fuel consumption is 42.73 slugs with range of 1,010.9 nmi.

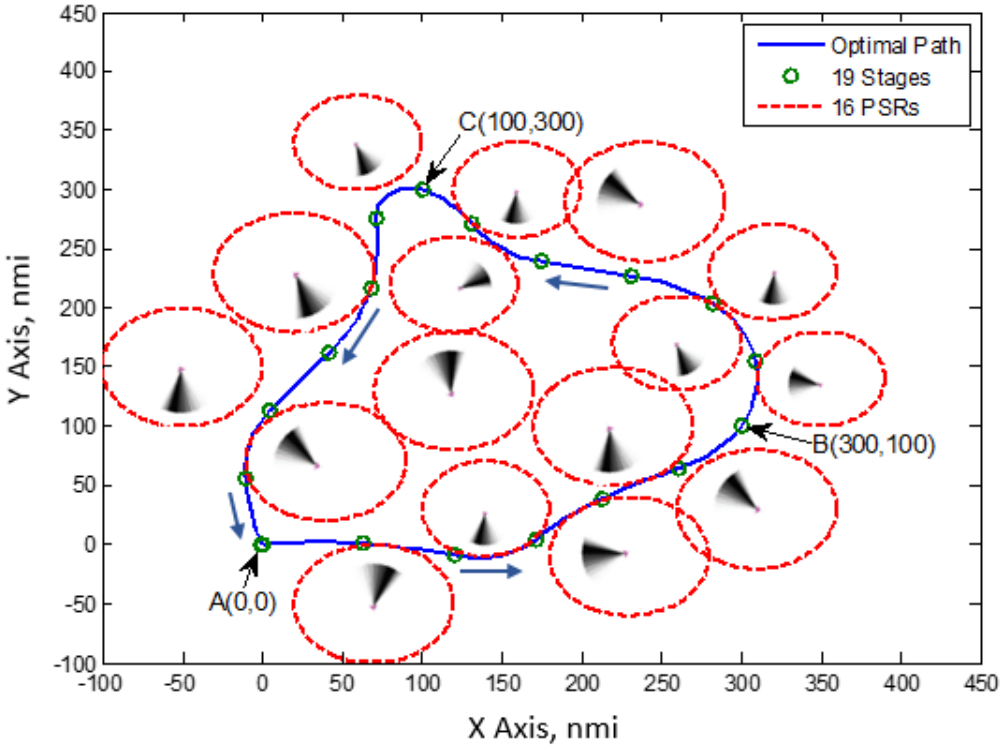

Fig. 4.14 Optimal Path Planning for Scenario 2 with Cruise at 30,000 ft

Figure 4.15 shows optimal velocity values of 19 discrete stages on the path for scenario 2. Although the average velocity range of the curve is between 350 ft/s and 450 ft/s, the velocity at coordinates C (300,100) is too low. This is because the aircraft has a sharp turning flight around way-point C coordinates in Figure 4.14. The velocity should be diminished in order to provide minimum turn radius (R) in equation (18). However, the

velocity should be greater than the stall speed. Therefore, the stall speed is determined by using equation (20):

$$V_{stall} = \sqrt{\frac{2nW}{\rho S C_{Lmax}}} \tag{20}$$

where n is the load factor, which is defined in equation (19), and (C_{Lmax} =1.46) is given in Table 2.3. The stall speed varies along the trajectory because the weight and load factor changes with time in equation (20). The optimal velocity values are greater than the stall speed along the trajectory.

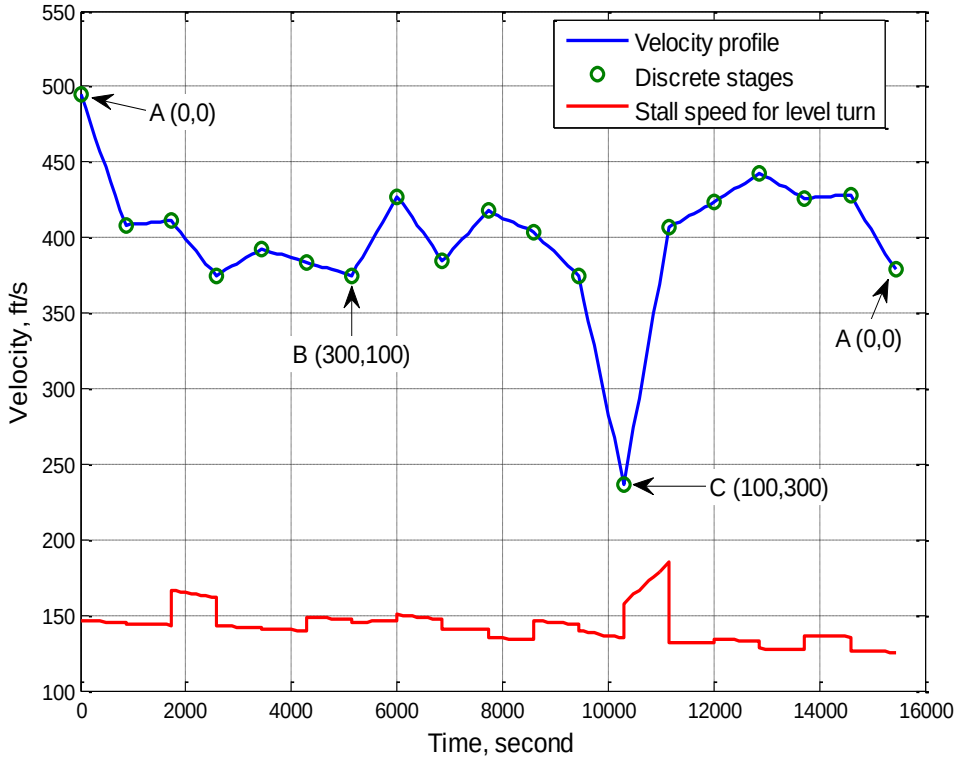

Fig. 4.15 Optimal Velocity Profile for Scenario 2 with Cruise at 30,000 ft

Figure 4.16 shows the optimal bank angle values on the path. The upper and lower bounds are determined by using equation (38). The optimal bank angle values are between the upper and lower bounds. The maximum bank angle occurs between way-point destination point C (100,300) and the following discrete stage in Figure 4.14.

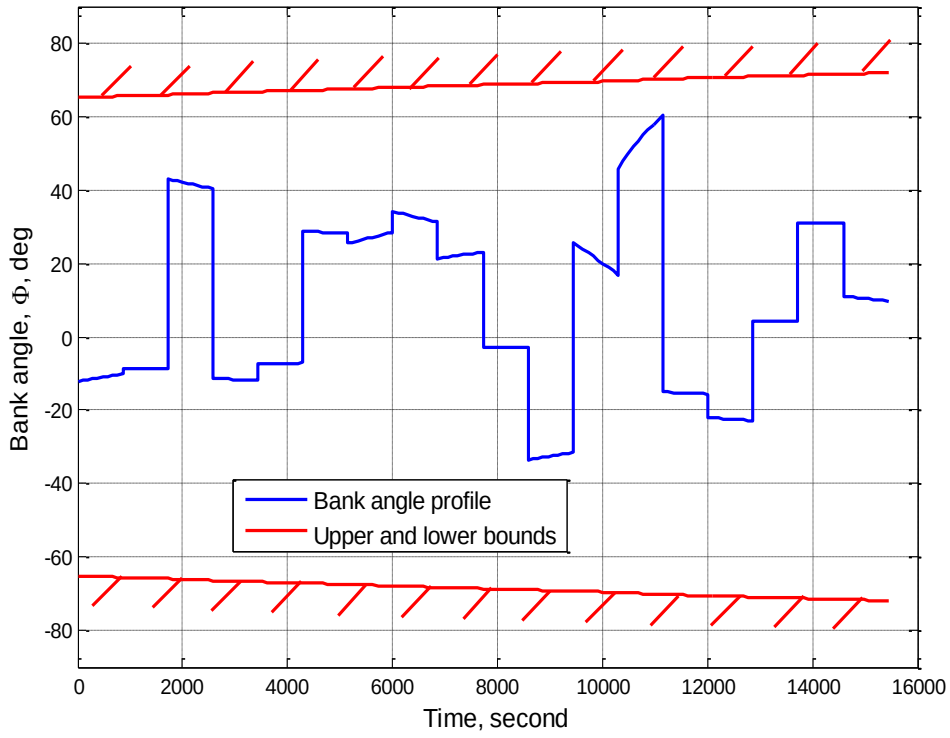

Fig. 4.16 Optimal Bank Angle Profile for Scenario 2 with Cruise at 30,000 ft

Figure 4.17 shows the heading angle profile along the trajectory. Heading angle is the angle between the direction the aircraft's velocity vector and the reference direction (X-axis). The starting heading angle is 2.55 degree and the final heading angle is 286.91 degree. In Figure 4.17, decreasing segments represent right turn and increasing segments represent left turn.

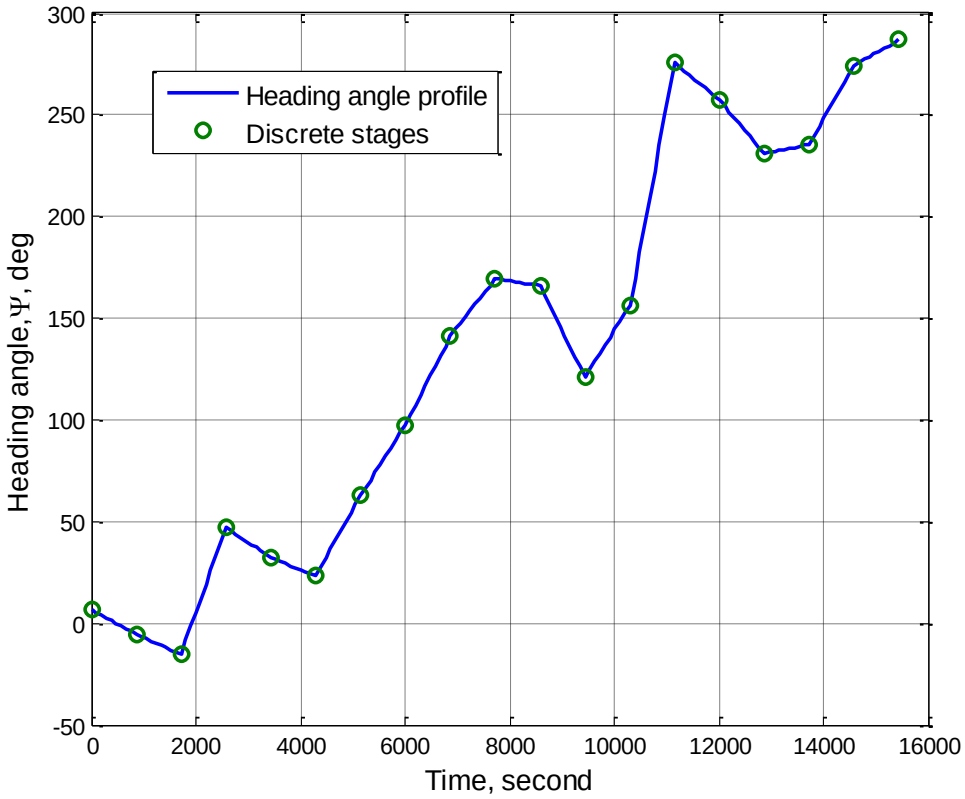

Fig. 4.17 Optimal Heading Angle Profile for Scenario 2 with Cruise at 30,000 ft

CHAPTER 5 – SUMMARY AND CONCLUSION

It is difficult to design an optimal path plan for obstacle-rich environments. In this thesis, a novel approach was implemented to determine a minimum-fuel path and improve convergence. This research focused on the path planning for an UCAV to minimize the fuel consumption so that the range increases for ISR.

The X-47A was modeled for a UCAV. The climbing and cruise flight dynamics of the X-47A were formulated in chapter 2. Non-ideal turbofan cycle analysis was performed for JT15D-5C engine model (see appendix). In this work, primary surveillance radars were obstacles to be avoided. Their known location and detection ranges allowed operators to search safe areas on the map.

The optimization problem was defined in chapter 3. There are five state differential equations (SDEs): (horizontal velocity X and Y components in the directions, acceleration, heading rate, and mass-flow rate), which were solved by two separate integration methods. In initial value problem (method 1), the inputs were discrete control variables vs. time: throttle and bank angle. In inverse dynamics (method 2), inputs were discrete state variables: velocity, heading angle, and mass of the aircraft. A new approach has been developed for method 2. Firstly, the shortest path was found among the safe areas by using ‘shpath’ code in MATLAB. Then the geometric model approach was applied for estimating initial heading angles at discrete stages.

The Fmincon optimization solver was used to find optimum discrete state and control variables. The Fmincon algorithm was selected because it is suitable for constrained

nonlinear multivariable functions. The second reason was that the Fmincon solver is faster than other equivalent optimization solvers.

The results of the comparison of the fuel efficiency for cruise altitude showed that a constant-altitude cruise is more fuel efficient compared to variable cruise altitude. Therefore, constant-altitude was selected for cruise flight. The numerical solution of method 1 and method 2 for obstacle-free situations demonstrated that inverse dynamics method is much faster than IVP method, so only the inverse dynamics method was implemented for the remainder of the optimization problems.

Optimal paths by avoiding obstacles were investigated using two scenarios. In the first scenario, the one-way path was found while avoiding sixteen obstacles and the same scenario was applied separately at cruise altitudes of 30,000, 35,000, and 40,000 ft. At higher cruise altitudes, the aircraft consumed less fuel because of lower air density. In the second scenario, the minimum-fuel closed path was determined while avoiding sixteen obstacles, and the aircraft flew over two way-point destination coordinates.

Inverse dynamics is the better method for find optimal path planning. It is much faster. Increasing the number of the discrete stages on the path made fuel efficiency slightly better. However, the computation time increased considerably. For the Fmincon algorithm, high-quality initial guesses are very important for finding the best path quickly. The geometric model provides a good estimation for the inverse dynamics method.

REFERENCES

[1] J. Nicas, "Why Some Drone Makers Hate the Word 'Drone' and Want to Change It," *The Wall Street Journal*, 09-Oct-2014 http://www.wsj.com/articles/why-some-drone-makers-hate-the-word-drone-and-want-to-change-it-1412821801 (cited May, 2016).

[2] J. F. Keane and S. S. Carr, "A Brief History of Early Unmanned Aircraft," *John Hopkins APL Tech. Dig.*, vol. 32, no. 3, pp. 558–571, 2013.

[3] UAS Task Force: Airspace Integration Integrated Product Team, "Unmanned Aircraft System Airspace Integration Plan," *Dep. Def.*, no. March, pp. 1–25, 2011.

[4] U.S. Department of Transportation, "Unmanned Aircraft System (UAS) Service Demand 2015-2035: Literature Review and Projections of Future Usage, Version 0.1," *Unmanned Aircr. Syst. Serv. Demand 2015 - 2035 - Lit. Rev. Proj. Futur. Usage*, p. 151, September, 2013.

[5] Lufthansa, "Fuel efficiency at the Lufthansa Group – Cutting costs and protecting the environment Lufthansa German Airlines," *Clim. Environ. Responsib.*, p. 5, 2012.

[6] T. Schouwenaars, B. De Moor, E. Feron, and J. How, "Mixed Integer Programming for Multi-Vehicle Path Planning," 2001.

[7] A. Richards and J. P. How, "Collision Avoidance Using Mixed Integer Linear Programming," *ACC02-AIAA1057*, 2002.

[8] A. Richards, "Trajectory Optimization using Mixed-Integer Linear Programming," University of Cambridge, 2000.

[9] E. J. Forsmo, "Optimal Path Planning for Unmanned Aerial Systems," Norwegian University of Science and Technology, 2012.

[10] P. T. Kabamba, S. M. Meerkov, and F. H. Zeitz, "Optimal Path Planning for Unmanned Combat Aerial Vehicles to Defeat Radar Tracking," *J. Guid. Control. Dyn.*, vol. 29, no. 2, pp. 279–288, 2006.

[11] S. A. Bortoff, "Path planning for UAVs," *Proc. 2000 Am. Control Conf. ACC (IEEE Cat. No.00CH36334)*, vol. 1, no. 6, pp. 364–368 vol.1, 2000.

[12] M. Guido and A. Frediani, "Flight Path Optimization at Constant Altitude," in *Variational Analysis and Aerospace Engineering*, vol. 33, pp. 21–32, 2009.

[13] S. Ansberry, "Cruise Flight Optimization of A Commercial Aircraft with Winds," University of Missouri-Columbia, 2015.

[14] B. R. Geiger, J. F. Horn, A. M. Delullo, and L. N. Long, "Optimal Path Planning of UAVs Using Direct Collocation with Nonlinear Programming," *Direct*, vol. 2, pp. 1–13, 2006.

[15] H. Liu, S. Chen, L. Shen, and J. Chen, "Tactical trajectory planning for stealth

unmanned aerial vehicle to win the radar game," *Def. Sci. J.*, vol. 62, no. 6, pp. 375–381, 2012.

[16] K. Mohan, "Optimal Path Planning and Trajectory Optimization for," University of Florida, 2011.

[17] B. R. Call, "Obstacle Avoidance for Unmanned Air Vehicles," Bringham Young University, 2006.

[18] W. Yang and C. Wenjie, "Path Planning and Obstacle Avoidance of Unmanned Aerial Vehicle Based on Improved Genetic Algorithms," pp. 8612–8616, 2014.

[19] X-47A Specifications 1, from https://en.wikipedia.org/wiki/Northrop_Grumman_X-47A_Pegasus (cited, May, 2016).

[20] X-47A Specifications 2, from http://www.airforce-technology.com/projects/x47/ (cited, May, 2016).

[21] A. Levy, M. Katz, O. Katzuni, A. Konevsky, J. Frumkin, T. Buium, and M. Argaman, "Final report Project 7-8 Team Cerberus-UCAV," 2009.

[22] Three-View Drawing of X-47A, from http://community.thefoundry.co.uk/discussion/topic.aspx?f=9&t=26067 (cited, May, 2016).

[23] M. Sadraey, "Drag Force and Drag Coefficient," in *Aircraft Performance: Analysis*, VDM Verlag Dr. Müller, 2009, pp. 1–282.

[24] S. A. Brandt, R. J. Stiles, J. J. Bertin, and R. Whitford, "Wings and Airplanes," in *Introduction to Aeronautics: A Desing Perspective*, Second Edi., J. A. Schetz, Ed. AIAA Education Series, pp. 130–131.

[25] JT15D Turbofans from Pratt & Whitney Canada, from http://www.pwc.ca/en/engines/jt15d (cited, May, 2016).

[26] European Aviation Safety Agency, "EASA-TCDS-E.077_(IM)_Pratt_and_Whitney_Canada_JT15D_series_engines-01-07092006.pdf," 2006.

[27] D. R. Greatrix, *Powered flight: The engineering of aerospace propulsion*, vol. 9781447124. 2012.

[28] R. A. Golub and J. S. Preisser, "Test-Engine and Inlet Performance of an Aircraft Used for Investigating flight Effects on Fan Noise," *NASA Tech. Pap.*, 1984.

[29] P. Vabre, "Air Traffic Services Surveillance Systems, Including An Explanation of Primary and Secondary Radar." .

[30] E. Ince, "Havacilik Elektronigi-3," *Tech. Artic. EMO* (in Turkish).

[31] D. Jenn, "Radar Fundamentals.", from http://faculty.nps.edu/jenn/Seminars/RadarFundamentals.pdf (cited, May, 2016).

[32] Radar figure, from https://en.wikipedia.org/wiki/Radar (cited, May, 2016).

[33] Fmincon Algorithm in MATLAB from, http://www.mathworks.com/help/optim/ug/fmincon.html *MathWorks*, (cited, May, 2016).

[34] Ode45 in MATLAB, from http://www.mathworks.com/help/matlab/ref/ode45.html *MathWorks* (cited, May, 2016).

[35] M. Kleder, "Shpath" code in MATLAB, from http://www.mathworks.com/matlabcentral/fileexchange/8625-shortest-path-with-obstacle-avoidance--ver-1-3-/all_files, 2008 (cited, May, 2016).

.

APPENDIX A

MATLAB programming language was implemented for computation and simulation.

Script and function m-files, which were used in this thesis, are presented below.

JT15D-5C Turbofan Engine Function M-file:

```
function [T,ct] = engine_JT15D5C(h,V)
      % Given inputs of flight conditions (altitude and velocity) this
      % funtion computes the non-ideal cycle analysis of the Pratt and
      % Whitney - Canada JT15D-5C turbofan engine.

%%%%% Start of Code %%%%%
[temp,asp,press,rho,rhoh] = atmos_1976(h);
      % holding atmospheric data for temperature, pressure, speed of
      % sound, density, and the height rate of change of density.

%%%%% Constants %%%%%
Y = 1.4;            % freestream value of gamma, from 0 to 1
Yc = 1.4;           % gamma c, assumed value, dimensionless
Yt = 1.35;          % gamma t, assumed value, dimensionless
CPc = 0.24;         % Btu/lbm-deg R
CPt = 0.262;        % Btu/lbm-deg R
Hc = 18500;         % Btu/lbm-deg R
R = 1716.4;         % gas constant, units of ft*lbf/slug*deg R
g = 32.17;          % acceleration of gravity, ft/s^2
alpha = 2;          % bypass ratio,
d = 27/12;          % inlet diameter, ft
A1 = (pi*(d^2))/4; % inlet area, ft^2
M1 = 0.45632;       % Mach number assumed at diffuser
M0 = V/asp;         % freesteam mach number, dimensionless
Tt0 = temp*(1+((Y-1)/2*(M0^2))); % stag temp of the freestrem air,deg R
Tt1 = Tt0;
T1 = Tt1/(1+((Y-1)/2*(M1^2))); % temp at diffuser, deg R
pt0 = press*(1+((Y-1)/2*(M0^2)))^(Y/(Y-1));% stagnation pressure at
        % freestream, psf pt0 = pt1 again, since the flow is isentropic
pt1 = pt0;
p1 = pt1/(1+((Y-1)/2*(M1^2)))^(Y/(Y-1)); % pressure at diffuser
rho1 = p1/(R*T1);  % density of air at the diffuser (station 1)
a1 = sqrt(Y*R*T1); % speed of sound at the diffuser
V1 = M1*a1;         % air speed at the diffuser
mdot = rho1*A1*V1; % mass-flow rate

%%%%% Efficiency Approximations %%%%%
Nb=0.98;            % burner efficiency, assumed value
Nm=0.99;            % mechanical efficiency, assumed value
Ec=0.92;            % polytropic efficiency in compressor, assumed value
Ecfan= 0.92;        % polytropic efficiency through the fan, assumed v.
Et = 0.91;          % polytopic efficiency in turbine, assumed value
```

```
%%%%% Pressure Ratios %%%%%
      % Assuming a matched nozzle, p9/p0=p9'/p0=1
Pid=0.99;      % pressure loss ratio for diffuser, assumed value
Pic=12.3;      % pressure ratio for compressor
Picfan = 1.5;  % fan pressure ratio
Pib = 0.98;    % pressure loss ratio for burner, assumed value
Pin = 0.99;    % pressure loss ratio for nozzle, assumed value
Pinfan = 0.99; % pressure loss ratio through fan, assumed value
Pir = (1 + ((Yc-1)/2)*M0^2)^(Yc/(Yc-1));
      % stagnation pressure ratio for the freestream (station 0)

%%%%% Tau Values %%%%%
Taur = 1 + ((Yc-1)/2)*M0^2;  % temp ratio for the freestream station 0
Tauc = Pic^((Yc-1)/(Yc*Ec)); % temperature ratio across the compressor
Taucfan = Picfan^((Yc-1)/(Yc*Ecfan));
Tt4 = 1751.7;                % Max temp of 1751.7 deg R
Taulambda = Tt4/temp;

f = ((Taulambda-(Taur*Tauc))/(((Nb*Hc)/(CPc*temp))-Taulambda));
      % fuel to air mass-flow rate, dimensionless value
Taut = 1-((Taur/(Nm*Taulambda*(1+f)))*((Tauc-1)+alpha*(Taucfan-1)));
      % stagnation temperature ratio for the turbine
Pit = Taut^(Yt/(Et*Yt-Et));
      % stagnation pressure ratio across the turbine
pt9_p9 = Pir*Pid*Pic*Pib*Pit*Pin;
      % stagnation temp @ station 9 to static temp @ 9 ratio
pt9f_p9f = Pir*Pid*Picfan*Pinfan;
      % stagnation temp @ station 9 to static temp @ 9 ratio,for fan
M0V9_V0 = ((2/(Yc-1))*Taulambda*Taut*(1-(pt9_p9)^(-(Yt-1)/Yt)))^(0.5);
      % core velocity ratio, M0*V9/V0
M0V9f_V0 = ((2/(Yc-1))*Taur*Taucfan*(1-(pt9f_p9f)^(-(Yc-1)/Yc)))^(1/2);
      % fan velocity ratio
core_sp_thrust = (((1+f)*(M0V9_V0))-M0)*(asp/(1+alpha));
      % the core thrust
fan_sp_thrust = (alpha*(M0V9f_V0 - M0))*(asp/(1+alpha));
      % fan specific thrust
specific_thrust = core_sp_thrust + fan_sp_thrust;
      % specific thrust value, ft/s
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
ct = ((f*3600*g)/((1+alpha)*specific_thrust))
      % Thrust Specific Fuel Consumption, 1/hr
T = specific_thrust*mdot;
      % Thrust, in lbf
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
end
```

Climbing Flight from Sea Level to 30,000 ft:

```
clear all; clc;
%%%%% Constants %%%%%
b=27.8;                      % wingspan, ft
S=387.29;                    % wing area, ft^2;
AR=b^2/S;                    % aspect ratio
e=0.75;                      % Osmwald coefficient
K=1/(pi*e*AR);               % induced-drag coefficient
CD0=0.011;                   % zero-lift drag coefficient
We=3836;                     % empty weight, lbf
Wf=1580;                     % fuel weight, lbf
Wto=(We+Wf)*0.995;           % takeoff weight, lbf
Np=31;                       % the number of waypoints

h=linspace(0,30000,Np);          % altitude points
W1=linspace(5388.9,5273.2,Np);   % weight points

for j=1:1:Np;
[temp, a_SL, press, rho_SL, rhoh]=atmos_1976(0);
      % atmospheric data at sea level
[temp, a, press, rho, rhoh]=atmos_1976(h(j));
      % atmospheric data at climbing flight

%%%%% Constants for engine m-file %%%%%
M1 = 0.45632;  Ec=0.92;  Ecfan= 0.92;  Et = 0.91;  Hc = 18500;
alpha = 2;  d = 27/12;  A1 = (pi*(d^2))/4; Yc = 1.4;  Yt = 1.35;
CPc = 0.24;  CPt = 0.262; R = 1716.4;  Nb=0.98;  Nm=0.99;
Pic=12.3; Picfan = 1.5; Pid=0.99;  Pib = 0.98;  Pin = 0.99;
Pinfan = 0.99; Tt4 = 1751.7;Taulambda = Tt4/temp;

Npts=1000;                       % the number of waypoints
v=linspace(90,810,Npts);         % velocity points
for i=1:1:Npts
    %%%%% turbofan engine calculation %%%%%
    M0(i)    = v(i)/a;
    T1(i)    = temp*(1+((Yc-1)/2*(M0(i)^2)))/(1+((Yc-1)/2*(M1^2)));
    p1(i)    = press*(T1(i)/temp)^(Yc/(Yc-1));
    rho1(i)  = p1(i)/(R*T1(i));
    V1(i)    = M1*sqrt(Yc*R*T1(i));
    mdot(i)  = rho1(i)*A1*V1(i);
    Taur(i)  = 1 + ((Yc-1)/2)*M0(i)^2;
    Pir(i)   =(1 + ((Yc-1)/2)*M0(i)^2)^(Yc/(Yc-1));
    Tauc     = Pic^((Yc-1)/(Yc*Ec));
    Taucfan  = Picfan^((Yc-1)/(Yc*Ecfan));
    f(i)     = ((Taulambda-(Taur(i)*Tauc))/(((Nb*Hc)/(CPc*temp))-
               Taulambda));
    Taut(i)  = 1-((Taur(i)/(Nm*Taulambda*(1+f(i))))*((Tauc-
               1)+alpha*(Taucfan-1)));
    Pit(i)   = Taut(i)^(Yt/(Et*Yt-Et));
    pt9_p9(i)   = Pir(i)*Pid*Pic*Pib*Pit(i)*Pin;
    pt9f_p9f(i)= Pir(i)*Pid*Picfan*Pinfan;
    M0V9_V0(i) = ((2/(Yc-1))*Taulambda*Taut(i)*(1-(pt9_p9(i))^(-(Yt-
                 1)/Yt)))^(0.5);
    M0V9f_V0(i)= ((2/(Yc-1))*Taur(i)*Taucfan*(1-(pt9f_p9f(i))^(-(Yc-
                 1)/Yc)))^(0.5);
```

```
    core_sp_thrust(i)= (((1+f(i))*(M0V9_V0(i)))-M0(i))*(a/(1+alpha));
    fan_sp_thrust(i) = (alpha*(M0V9f_V0(i) - M0(i)))*(a/(1+alpha));
    specific_thrust(i) = core_sp_thrust(i) + fan_sp_thrust(i);
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    T(i) = specific_thrust(i)*mdot(i);
    ct(i)= ((f(i)*3600*32.174)/((1+alpha)*specific_thrust(i)));
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    cl(j,i)=2*(W1(j))/(rho*v(i)^2*S);
            % lift coefficient at climbing
    cd(j,i)=CD0+K*cl(j,i)^2;
            % drag coefficient at climbing
    tr(j,i)=0.5*rho*v(i)^2*S*cd(j,i);
            % thrust required, lbf
    pr(j,i)=tr(j,i)*v(i);
            % power required, lbf-ft/s
    pa(i)=T(i)*v(i);
            % power available, lbf-ft/s
    pex(j,i)=pa(i)-pr(j,i);
            % excess power, lbf-ft/s
    rc(j,i)=(pa(i)-pr(j,i))/W1(j);
            % rate of climb, ft/s
end
[p_excess_max(j,:),index(j,:)]=max(pex(j,:));
Max_excess_power(j,:)=p_excess_max(j,:);
V_max_excess_power(j,:)=v(index(j,:));

[Rate_of_climb_max(j,:),index(j,:)]=max(rc(j,:));
 Max_rate_of_climb(j,:)=Rate_of_climb_max(j,:);
V_max_rate_of_climb(j,:)=v(index(j,:));
l(j,:)=Max_rate_of_climb(j,:)/V_max_rate_of_climb(j,:);
Flight_path_angle(j,:)=asin(l(j,:))*180/pi;

end
flight_path_angle=Flight_path_angle';
            % flight path angle at each point
v_for_max_rate_of_climb=V_max_excess_power';
            % velocity for max RC at each point
max_rate_of_climb=Max_rate_of_climb';
            % max RC at each point
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Npts2=1000;                                 % the number of waypoints
hh=linspace(0,30000,Npts2);                 % altitude points
hhh=linspace(0,30000,Np);                   % altitude points
VmaxRCp=v_for_max_rate_of_climb;
maxRCp=max_rate_of_climb;
flight_p_a_p=flight_path_angle;
VmaxRC=interp1(hhh,VmaxRCp,hh);             % interpolated values of V
maxRC=interp1(hhh,maxRCp,hh);               % interpolated values of max RC
flight_p_a=interp1(hhh,flight_p_a_p,hh);% interpolated values of f.p.a.

for i=1:1:Npts2;
    if hh(i)<=36089;
W1(1)=Wto;
%%%%% Standard atmosphere properties %%%%%
aa1(i)=(1-(6.875*10^-6*hh(i)));
```

```
aa2(i)=(1-(6.875*10^-6*hh(i)))^5.2561;
rhoo(i)=rho_SL*(aa2(i)/aa1(i));
else
aa1(i)=0.75189;
aa2(i)=0.2234*exp(4.806*(10^-5)*(36089-hh(i)));
rhoo(i)=rho_SL*(aa2(i)/aa1(i));
    end
[T(i),ct(i)] = engine_JT15D5C(hh(i),VmaxRC(i));
      % JT15D-5C turbofan engine function m-file

%%%%% Time to climb %%%%%
tt(i)=(1/maxRC(i));
if i==1
    Time(i)=0;
else
    H = hh(1:i);
    Time1(i)=trapz(H,tt);
    Time(i)=Time1(i)-Time1(i-1);
end
Wf_RC(i)=ct(i)*T(i)*Time(i)/3600;
W1(i+1)=W1(i)-Wf_RC(i);

sdot(i)=VmaxRC(i)*cosd(flight_p_a(i))/6076;
hdot(i)=VmaxRC(i)*sind(flight_p_a(i))/6076;
end
W1(:,1)=[];
W2=W1(Npts2)

time_climb=cumtrapz(hh,tt);
range_climb=cumtrapz(Time1,VmaxRC)/6076;

s=cumtrapz(time_climb,sdot);    % horizontal position, nmi
h=cumtrapz(time_climb,hdot);    % vertical position, nmi
```

Gliding Flight from 30,000ft to 15,000 ft :

Running Ode45 M-file:

```
clc; clear all;
tic             % start stopwatch timer
format shorteng % output display format

[tt xx]= ode45(@fun_gliding,[0 750*0.4667],[0   30000   426.15  0  ]);
                % syntax of ode45 solver

toc             % end stopwatch timer
```

Main Function for Gliding Flight:

```
function dx =fun_gliding(t,x)

%%%%% Constants %%%%%
b=27.8;                     % wingspan, ft
S=387.29;                   % wing area, ft^2;
AR=b^2/S;                   % aspect ratio
e=0.75;                     % Osmwald coefficient
K=1/(pi*e*AR);              % induced-drag coefficient
CD0=0.011;                  % zero-lift drag coefficient
W1=5270;                    % initial weight at cruise, lbf
g=32.174;                   % acceleration of gravity, ft/s^2
m=W1/g;                     % constant mass of the aircraft, slug
CLs=sqrt(CD0/K);            % lift coefficient for max L/D

%%%%% State variables %%%%%
s=x(1);                     % horizontal position
h=x(2);                     % vertical position / altitude
V=x(3);                     % velocity
gamma=x(4);                 % flight path angle

[temp, a, press, rho, rhoh]=atmos_1976(h);
% atmospheric data at gliding

q=(V^2*rho)/2;              % dynamic pressure, slug/ft-s^2
CL=CLs;                     % lift coefficient at gliding phase
CD=CD0+K*CL^2;              % drag coefficient at gliding phase
D=q*S*CD;                   % drag force at gliding phase, lbf
L=q*S*CL;                   % lift force at gliding phase, lbf

%%%%%State differential equations%%%%%
dx=zeros(4,1);
dx(1)=V*cos(gamma);                 % sdot
dx(2)=V*sin(gamma);                 % hdot
dx(3)=-(D/m)-(g*sin(gamma));        % Vdot
dx(4)=(L)/(m*V)-(g*cos(gamma)/V);   % gammadot
end
```

Running Optimization M-file for Initial Value Problem (IVP):

```
clc; clear all;
tic             % start stopwatch timer
format shorteng % output display format

global deltaaa  bank  a_tof  a_vof  psi
      % global variables

deltaaa=[0.651 0.440 0.468 0.486 0.451 0.431 0.411 0.397 0.381 0.406]';
      % initial estimated throttle values
bank=[0 0 0 0 0 0 0 0 0 0]';
      % initial estimated bank angle values, rad
a_vof=1;        % velocity scaling factor, assumed value
psi=45*pi/180;  % heading angle scaling factor, assumed value , rad
a_tof=1;        % flight-time scaling factor, assumed value
V0=450;         % initial estimated velocity, ft/s

initiall=[deltaaa; bank; a_vof; a_tof; psi];
      % initial point, dimension of the matrix, 23x1

lb=[0 0 0 0 0 0 0 0 0 0 -1.117 -1.117 -1.117 -1.117 -1.117 -1.117...
    -1.117 -1.117 -1.117 -1.117  0.54   0.5 ];
      % lower bounds
ub=[1 1 1 1 1 1 1 1 1 1  1.117  1.117  1.117  1.117  1.117  1.117...
     1.117  1.117  1.117  1.117  1.58   2   ];
      % upper bounds
options = optimset('Display','iter','Algorithm','interior-point');
      % optimization options
[x,fval] =fmincon(@obj_funA10, initiall, [], [], [], [], lb, ub,
@nonlconstrA10,options)
      % syntax of Fmincon algorithm
toc             % end stopwatch timer
```

Objective Function M-file for IVP:

```
function [fvalll]=obj_funA10(x)
global deltaaa  bank  a_tof  a_vof  psi
      % global variables

%%%%% Variables %%%%%
deltaaa=x(1:10);
bank=x(11:20);
a_vof=x(21);
a_tof=x(22);
psi=x(23);

tspan=[0   a_tof*15000];
      % interval of integration
y0=[0   0   450*a_vof   5270/32.174   psi];
      % initial condition of state variables
[tt xx]= ode45(@funA10,tspan,y0);
      % syntax of ode45 solver
```

```
fvalll=(xx(1,4)-xx(end,4));
      % the difference between initial and final masses of the aircraft
end
```

Nonlinear Constraint Function M-file for IVP:

```
function [c,ceq]=nonlconstrA10(x)
global deltaaa  bank  a_tof  a_vof  psi
      % global variables
%%%%% Variables %%%%%
deltaaa=x(1:10);
bank=x(11:20);
a_vof=x(21);
a_tof=x(22);
psi=x(23);

tspan=[0   a_tof*15000];
      % interval of integration
y0=[0   0   450*a_vof   5270/32.174   psi];
      % initial condition of state variables
[tt xx]= ode45(@funA10,tspan,y0);
      % syntax of ode45 solver

c=[];
      % inequality constraints
ceq=[xx(end,1)/6076-700, xx(end,2)/6076-700];
      % equality constraints
```

Main Function for IVP:

```
function dx =funA10(t,x)
global deltaaa  bank  a_vof  a_tof  psi
      % global variables

%%%%% Constants %%%%%
b=27.8;                     % wingspan, ft
S=387.29;                   % wing area, ft^2;
AR=b^2/S;                   % aspect ratio
e=0.75;                     % Osmwald coefficient
K=1/(pi*e*AR);              % induced-drag coefficient
CD0=0.011;                  % zero-lift drag coefficient
W1=5270;                    % initial weight at cruise, lbf
g=32.174;                   % acceleration of gravity, ft/s^2
rho=8.8938e-004;            % density of air, slug/ft^3
h=30000;                    % altitude, ft
LDmax=0.5/(CD0*K)^0.5;      % max L/D
CLs=sqrt(CD0/K);            % lift coefficient at max L/D

%%%%%State variables%%%%%
X=x(1);
Y=x(2);
V=x(3);
m=x(4);
```

```
psi=x(5);

[T,ct] = engine_JT15D5C(h,V);
      % JT15D-5C turbofan engine function m-file
t_pts=linspace(0,a_tof*15000*0.956,10);
      % time points
delta=interp1(t_pts,deltaaa,t);
      % interpolated values of throttle
phi=interp1(t_pts,bank,t);
      % interpolated values of bank angle

TTT=delta*T;                % thrust coming from engine, lbf
q=(V^2*rho)/2;              % dynamic pressure, slug/ft-s^2
CL=(m*g)/(q*S*cos(phi));    % lift coefficient at cruise
CD=CD0+K*CL^2;              % drag coefficient at cruise
D=q*S*CD;                   % drag force, lbf

%%%%%State differential equations%%%%%
dx=zeros(5,1);
dx(1)=V*cos(psi);                % Xdot
dx(2)=V*sin(psi);                % Ydot
dx(3)=(TTT/m)-(D/m);             % Vdot
dx(4)=-ct*TTT/(g*3600);          % Psidot
dx(5)=(q*S*CL*sin(phi))/(m*V);   % mdot
end
```

Running Optimization M-file for Inverse Dynamics:

```
clc; clear all;
tic             % start stopwatch timer
format shorteng % output display format
global Vel a_tof psi Mass t Npts_j Npts_i
      % global variables

Vel=[484 432 441 424 386 386 406 398 381 360]';
      % initial estimated velocity values
psi=[21 31 52 15 77 82 22 17 62 84]';
      % initial estimated heading angle values
Mass=[158 153 148 143 139 135 131 127 123]';
      % initial estimated mass values

a_tof=1;        % flight-time scaling factor, assumed value
t=15000;        % estimated flight time, second
Npts_j=9;       % the number of segments on the path
Npts_i=20;      % the number of waypoints at each segment

initial=[Vel; a_tof; psi; Mass];
      % initial point, dimension of the matrix, 30x1
lb=[225 225 225 225 225 225 225 225 225 225 0.1]; % lower bounds
ub=[683 684 685 686 688 689 690 691 693 695 1.5]; % upper bounds

options = optimset('Display','iter','Algorithm','interior-point');
      % optimization options
[x,fval] =fmincon(@obj_funB10, del_initial, [], [], [], [], lb, ub,
@nonlconstrB10,options)
      % syntax of Fmincon algorithm
toc             % end stopwatch timer
```

Objective Function M-file for Inverse Dynamics:

```
function [min_fuel]=obj_funB10(x)
global Vel a_tof psi Mass t Npts_j Npts_i
      % global variables


%%%%% Variables %%%%%
Vel=x(1:Npts_j+1);
a_tof=x(Npts_j+2);
psi=x(Npts_j+3:2*Npts_j+3);
Mass=x(2*Npts_j+4:3*(Npts_j+1));

%%%%% Constants %%%%%
b=27.8;                     % wingspan, ft
S=387.29;                   % wing area, ft^2;
AR=b^2/S;                   % aspect ratio
e=0.75;                     % Osmwald coefficient
K=1/(pi*e*AR);              % induced-drag coefficient
CD0=0.011;                  % zero-lift drag coefficient
W1=5270;                    % initial weight at cruise, lbf
```

```
g=32.174;                    % acceleration of gravity, ft/s^2
rho=8.8938e-004;             % density of air, slug/ft^3
h=30000;                     % altitude, ft
time=0:t*a_tof/Npts_j:t*a_tof; % time points

for j=1:Npts_j;

    psi_pts(j,:)=linspace(psi(j),psi(j+1),Npts_i);
      % heading angle waypoints between two stages
    V_pts(j,:)=linspace(Vel(j),Vel(j+1),Npts_i);
      % velocity waypoints between two stages
    time_pts(j,:)=linspace(time(j),time(j+1),Npts_i);
      % time waypoints between two stages
    a_pts(j)=(Vel(j+1)-Vel(j))/(t*a_tof/Npts_j);
      % acceleration values of each segment
    b_pts(j)=(psi(j+1)-psi(j))/(t*a_tof/Npts_j);
      % heading rate values of each segment

    if j==1
        m_pts(j,:)=linspace(W1/g,(Mass(j)),Npts_i);
    else
        m_pts(j,:)=linspace(Mass(j-1),Mass(j),Npts_i);
    end

    for i=1:1:Npts_i
        [T(j,i),ct(j,i)] = engine_JT15D5C(h, V_pts(j,i));
            % JT15D-5C turbofan engine function m-file
        q(j,i)=(V_pts(j,i)^2*rho)/2;
            % dynamic pressure, slug/ft-s^2
        phi(j,i)=atand(b_pts(j)*V_pts(j,i)/g);
            % bank angle, deg
        CL(j,i)=(m_pts(j,i)*g)/(q(j,i)*S*cosd(phi(j,i)));
            % lift coefficient at cruise
        CD(j,i)=CD0+K*CL(j,i)^2;
            % drag coefficient at cruise
        D(j,i)=q(j,i)*S*CD(j,i);
            % drag force, lbf
        T1(j,i)=m_pts(j,i)*a_pts(j)+D(j,i);
            % thrust force, lbf
        mdot(j,i)=ct(j,i)*T1(j,i)/(g*3600);
            % mass-flow rate, slug/s
        Xdot(j,i)=V_pts(j,i)*cosd(psi_pts(j,i))/6076;
            % horizontal velocity in direction X axis, ft/s
        Ydot(j,i)=V_pts(j,i)*sind(psi_pts(j,i))/6076;
            % horizontal velocity in direction Y axis, ft/s
    end
        zz(j,:)=cumtrapz(time_pts(j,:),mdot(j,:));
            % numerical integration of mdot
        Xz(j,:)=cumtrapz(time_pts(j,:),Xdot(j,:));
            % numerical integration of Xdot
        Yz(j,:)=cumtrapz(time_pts(j,:),Ydot(j,:));
            % numerical integration of Ydot
end

for k=1:Npts_j;
    fuel(k,:)=zz(k,end);
```

```
            % fuel consumption at each stage, slug
end
min_fuel=sum(fuel);
            % minimizing fuel consumption
end
```

Nonlinear Constraint Function M-file for IVP:

```
function [c,ceq]=nonlconstrB10(x)
global Vel a_tof psi Mass t Npts_j Npts_i
        % global variables

%%%%% Variables %%%%%
Vel=x(1:Npts_j+1);
a_tof=x(Npts_j+2);
psi=x(Npts_j+3:2*Npts_j+3);
Mass=x(2*Npts_j+4:3*(Npts_j+1));

%%%%% Constants %%%%%
b=27.8;                         % wingspan, ft
S=387.29;                       % wing area, ft^2;
AR=b^2/S;                       % aspect ratio
e=0.75;                         % Osmwald coefficient
K=1/(pi*e*AR);                  % induced-drag coefficient
CD0=0.011;                      % zero-lift drag coefficient
W1=5270;                        % initial weight at cruise, lbf
g=32.174;                       % acceleration of gravity, ft/s^2
rho=8.8938e-004;                % density of air, slug/ft^3
h=30000;                        % altitude, ft
time=0:t*a_tof/Npts_j:t*a_tof; % time points

for j=1:Npts_j;

    psi_pts(j,:)=linspace(psi(j),psi(j+1),Npts_i);
      % heading angle waypoints between two stages
    V_pts(j,:)=linspace(Vel(j),Vel(j+1),Npts_i);
      % velocity waypoints between two stages
    time_pts(j,:)=linspace(time(j),time(j+1),Npts_i);
      % time waypoints between two stages
    a_pts(j)=(Vel(j+1)-Vel(j))/(t*a_tof/Npts_j);
      % acceleration values of each segment
    b_pts(j)=(psi(j+1)-psi(j))/(t*a_tof/Npts_j);
      % heading rate values of each segment

    if j==1
        m_pts(j,:)=linspace(W1/g,(Mass(j)),Npts_i);
    else
        m_pts(j,:)=linspace(Mass(j-1),Mass(j),Npts_i);
    end

    for i=1:1:Npts_i
        [T(j,i),ct(j,i)] = engine_JT15D5C(h, V_pts(j,i));
            % JT15D-5C turbofan engine function m-file
        q(j,i)=(V_pts(j,i)^2*rho)/2;
            % dynamic pressure, slug/ft-s^2
```

```
        phi(j,i)=atand(b_pts(j)*V_pts(j,i)/g);
            % bank angle, deg
        CL(j,i)=(m_pts(j,i)*g)/(q(j,i)*S*cosd(phi(j,i)));
            % lift coefficient at cruise
        CD(j,i)=CD0+K*CL(j,i)^2;
            % drag coefficient at cruise
        D(j,i)=q(j,i)*S*CD(j,i);
            % drag force, lbf
        T1(j,i)=m_pts(j,i)*a_pts(j)+D(j,i);
            % thrust force, lbf
        mdot(j,i)=ct(j,i)*T1(j,i)/(g*3600);
            % mass-flow rate, slug/s
        Xdot(j,i)=V_pts(j,i)*cosd(psi_pts(j,i))/6076;
            % horizontal velocity in direction X axis, ft/s
        Ydot(j,i)=V_pts(j,i)*sind(psi_pts(j,i))/6076;
            % horizontal velocity in direction Y axis, ft/s
    end
        zz(j,:)=cumtrapz(time_pts(j,:),mdot(j,:));
            % numerical integration of mdot
        Xz(j,:)=cumtrapz(time_pts(j,:),Xdot(j,:));
            % numerical integration of Xdot
        Yz(j,:)=cumtrapz(time_pts(j,:),Ydot(j,:));
            % numerical integration of Ydot
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for k=1:Npts_j;
    xx(k,:)=Xz(k,end);
end
X=sum(xx);
      % final position of aircraft in the direction X axis, nmi
for k=1:Npts_j;
    yy(k,:)=Yz(k,end);
end
Y=sum(yy);
      % final position of aircraft in the direction Y axis, nmi
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for k=1:Npts_j;
    if k==1;
        XX(k,:)=Xz(k,:);
    else
        for m=1:k-1
            XXz(k,m)=Xz(m,end);
        end
        XXX(k,:)=sum(XXz(k,:));
        XX(k,:)=XXX(k,:)+Xz(k,:);
    end

end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
XxX=[XX(1,:) XX(2,:) XX(3,:) XX(4,:) XX(5,:) XX(6,:)...
     XX(7,:) XX(8,:) XX(9,:) ];
      % the position at each stage in the direction X axis, nmi
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for k=1:Npts_j;
    if k==1;
        YY(k,:)=Yz(k,:);
    else
```

```
        for m=1:k-1
            YYz(k,m)=Yz(m,end);
        end
        YYY(k,:)=sum(YYz(k,:));
        YY(k,:)=YYY(k,:)+Yz(k,:);
    end

end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
YyY=[YY(1,:) YY(2,:) YY(3,:) YY(4,:) YY(5,:) YY(6,:)...
     YY(7,:) YY(8,:) YY(9,:) ];
      % the position at each stage in the direction Y axis, nmi
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for k=1:Npts_j;
    if k==1;
        mass(k)=W1/g-zz(k,end);
    else
        for m=1:k
            zzz(k,m)=zz(m,end);
        end
        zzzz(k,:)=sum(zzz(k,:));
        mass(k)=W1/g-zzzz(k,:);
    end
end
      % fuel consumption at each stage, slug

%%%%% Inequality constraints %%%%%
c=[-(((XxX-100).^2)+((YyY-130).^2))+4900;...
   -(((XxX-200).^2)+((YyY-370).^2))+4900;...
   -(((XxX-300).^2)+((YyY-130).^2))+4900;...
   -(((XxX-320).^2)+((YyY-300).^2))+4900;...
   -(((XxX-440).^2)+((YyY-550).^2))+4900;...
   -(((XxX-460).^2)+((YyY-250).^2))+4900;...
   -(((XxX-480).^2)+((YyY-400).^2))+4900;...
   -(((XxX-590).^2)+((YyY-250).^2))+4900;...
   -(((XxX-600).^2)+((YyY-460).^2))+4900;...
   -(((XxX-600).^2)+((YyY-620).^2))+4900;...
   -(((XxX-700).^2)+((YyY-370).^2))+4900;...
   -(((XxX-190).^2)+((YyY-500).^2))+2500;...
   -(((XxX-210).^2)+((YyY-250).^2))+4900;...
   -(((XxX-320).^2)+((YyY-450).^2))+2500;...
   -(((XxX-500).^2)+((YyY-100).^2))+2500;...
   -(((XxX-700).^2)+((YyY-130).^2))+2500;];

%%%%% Equality constraints %%%%%
ceq=[X-700; Y-700;
     mass(1)-Mass(1); mass(2)-Mass(2); mass(3)-Mass(3); mass(4)-
Mass(4);...
     mass(5)-Mass(5); mass(6)-Mass(6); mass(7)-Mass(7); mass(8)-
Mass(8);...
     mass(9)-Mass(9);];
```

Shortest Path Code:

```
close all;clear all; clc;
```

```
tic         % start stopwatch timer

xi=1;                       % intitial position in X axis
yi=1;                       % intitial position in Y axis
xf=700;                     % final position in X axis
yf=700;                     % final position in Y axis
n = 800;                    % number of matrix dimension
A = zeros(n);               % 800x800 zeros matrix
[rr, cc] = meshgrid(1:n);   % grid vector

%%%%% Inequality constraints %%%%%
C1 = sqrt((rr - 100).^2 + (cc - 130).^2)<= 70;
C2 = sqrt((rr - 200).^2 + (cc - 370).^2)<= 70;
C3 = sqrt((rr - 200).^2 + (cc - 250).^2)<= 50;
C4 = sqrt((rr - 190).^2 + (cc - 500).^2)<= 50;
C5 = sqrt((rr - 300).^2 + (cc - 130).^2)<= 70;
C6 = sqrt((rr - 320).^2 + (cc - 300).^2)<= 70;
C7 = sqrt((rr - 320).^2 + (cc - 450).^2)<= 50;
C8 = sqrt((rr - 460).^2 + (cc - 250).^2)<= 70;
C9 = sqrt((rr - 440).^2 + (cc - 550).^2)<= 70;
C10 = sqrt((rr - 500).^2 + (cc - 100).^2)<= 70;
C11 = sqrt((rr - 480).^2 + (cc - 400).^2)<= 70;
C12 = sqrt((rr - 590).^2 + (cc - 250).^2)<= 70;
C13 = sqrt((rr - 600).^2 + (cc - 460).^2)<= 70;
C14 = sqrt((rr - 600).^2 + (cc - 620).^2)<= 70;
C15 = sqrt((rr - 700).^2 + (cc - 130).^2)<= 50;
C16 = sqrt((rr - 700).^2 + (cc - 370).^2)<= 70;

M=A+C1+C2+C3+C4+C5+C6+C7+C8+C9+C10+C11+C12+C13+C14+C15+C16;
      % grid matrix of zeros and ones
[y,x,da]=shpath(M,yi,xi,yf,xf);
      % compute path
toc         % end stopwatch timer
```

Running Optimization M-file for Geometric Model:

```
clc; clear all;
tic             % start stopwatch timer
format short    % output display format
global X Y
      % global variables

X=[0 100 186 258 359 413 407 490 594 683 700]';
      % initial estimated X values
Y=[0  43 108 189 228 322 430 500 531 592 700]';
      % initial estimated Y values
initials=[X; Y]
      % initial point, dimension of the matrix, 22x1

options = optimset('Display','iter','Algorithm','interior-point');
      % optimization options
[x,fval] =fmincon(@obj_funC11, initials, [], [], [], [], [], [],
@nonlconstrC11,options)
      % syntax of Fmincon algorithm
```

```
XX=x(1:11);      % optimized X values
YY=x(12:22);     % optimized Y values

for i=1:10
    if XX(i+1)>=XX(i) && YY(i+1)>=YY(i);
       anglee(i)=atand((YY(i+1)-YY(i))/(XX(i+1)-XX(i)));
       angle(i)=anglee(i);
    elseif XX(i+1)>=XX(i) && YY(i+1)<=YY(i);
       angleee(i)=atand((YY(i+1)-YY(i))/(XX(i+1)-XX(i)));
       angle(i)=angleee(i);
    elseif XX(i+1)<=XX(i) && YY(i+1)>=YY(i);
       angleeee(i)=atand((YY(i+1)-YY(i))/(XX(i+1)-XX(i)));
       angle(i)=angleeee(i)+180;
    elseif XX(i+1)<=XX(i) && YY(i+1)<=YY(i);
       angleeeee(i)=atand((YY(i+1)-YY(i))/(XX(i+1)-XX(i)));
       angle(i)=angleeeee(i)+180;
    elseif XX(i+1)>=XX(i) && YY(i+1)<=YY(i);
       angleeeeee(i)=atand((YY(i+1)-YY(i))/(XX(i+1)-XX(i)));
       angle(i)=angleeeeee(i)+360;
    end
end

%%%%% Inclination angles of segments %%%%%
angle1=angle(1);
angle2=angle(2);
angle3=angle(3);
angle4=angle(4);
angle5=angle(5);
angle6=angle(6);
angle7=angle(7);
angle8=angle(8);
angle9=angle(9);
angle10=angle(10);

anglee=[ angle1 angle2 angle3 angle4 angle5...
         angle6 angle7 angle8 angle9 angle10]'
      % initial estimation of heading angle, deg
toc              % end stopwatch timer
```

Objective Function M-file for Geometric Model:

```
function [min_length]=obj_funC11(x)
global X Y
      % global variables

%%%%% Variables %%%%%
X=x(1:11);
Y=x(12:22);

Npts_j=10;  % the number of segments on the path
for j=1:Npts_j;
    length(j)=sqrt((X(j+1)-X(j))^2+(Y(j+1)-Y(j))^2);
end
min_length=sum(length); % minimizing the length of the total segments
end
```

Nonlinear Constraint Function M-file for Geometric Model:

```
function [c,ceq]=nonlconstrC11(x)
global X Y
% global variables

%%%%% Variables %%%%%
X=x(1:11);
Y=x(12:22);

Npts_j=10;  % the number of segments on the path
Npts_i=30;  % the number of waypoints at each segment

for j=1:Npts_j;
    x1(j,:)=linspace(X(j),X(j+1),Npts_i);
    y1(j,:)=linspace(Y(j),Y(j+1),Npts_i);
end

XxX=[x1(1,:) x1(2,:) x1(3,:) x1(4,:) x1(5,:)...
     x1(6,:) x1(7,:) x1(8,:) x1(9,:) x1(10,:)];
% the position at each stage in the direction X axis, nmi
YyY=[y1(1,:) y1(2,:) y1(3,:) y1(4,:) y1(5,:)...
     y1(6,:) y1(7,:) y1(8,:) y1(9,:) y1(10,:)];
% the position at each stage in the direction Y axis, nmi

for j=1:Npts_j;
    length(j)=sqrt((X(j+1)-X(j))^2+(Y(j+1)-Y(j))^2);
end

%%%%% Inequality constraints %%%%%
c=[-(((XxX-100).^2)+((YyY-130).^2))+4900;...
   -(((XxX-200).^2)+((YyY-370).^2))+4900;...
   -(((XxX-300).^2)+((YyY-130).^2))+4900;...
   -(((XxX-320).^2)+((YyY-300).^2))+4900;...
   -(((XxX-440).^2)+((YyY-550).^2))+4900;...
   -(((XxX-460).^2)+((YyY-250).^2))+4900;...
   -(((XxX-480).^2)+((YyY-400).^2))+4900;...
   -(((XxX-590).^2)+((YyY-250).^2))+4900;...
   -(((XxX-600).^2)+((YyY-460).^2))+4900;...
   -(((XxX-600).^2)+((YyY-620).^2))+4900;...
   -(((XxX-700).^2)+((YyY-370).^2))+4900;...
   -(((XxX-190).^2)+((YyY-500).^2))+2500;...
   -(((XxX-210).^2)+((YyY-250).^2))+4900;...
   -(((XxX-320).^2)+((YyY-450).^2))+2500;...
   -(((XxX-500).^2)+((YyY-100).^2))+2500;...
   -(((XxX-700).^2)+((YyY-130).^2))+2500;];

%%%%% Equality constraints %%%%%
ceq=[ X(end)-700; Y(end)-700; X(1)-0; Y(1)-0;
    length(1)-length(2); length(2)-length(3); length(3)-length(4);
    length(4)-length(5); length(5)-length(6); length(6)-length(7);
    length(7)-length(8); length(8)-length(9); length(9)-length(10);];
```