UNIVERSITY OF CALIFORNIA,
IRVINE

A Sequential Linear Quadratic Approach for Constrained Nonlinear Optimal Control with Adaptive Time Discretization and Application to Higher Elevation Mars Landing Problem

THESIS

submitted in partial satisfaction of the requirements for the degree of

MASTER OF SCIENCE

in

Mechanical Engineering

by

Amit Sandhu

Thesis Committee:
Professor Athanasios Sideris, Chair
Professor Kenneth D. Mease
Professor Faryar Jabbari

2015

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to express my special appreciation to Professor Athanasios Sideris for his aspiring guidance and imperative advice at every step of my research. He has been a tremendous mentor to me. I would like to thank him for fostering my interest in this topic and for allowing me to grow as an engineer. Without his persistent help this thesis would not have been possible.

I wish to thank my committee members, Professor Kenneth D. Mease and Professor Faryar Jabbari, for consenting to serve as my committee members and offer valuable suggestions and comments.

ABSTRACT OF THE THESIS

A Sequential Linear Quadratic Approach for Constrained Nonlinear Optimal Control with Adaptive Time Discretization and Application to Higher Elevation Mars Landing Problem

by

Amit Sandhu

Master of Science in Mechanical Engineering

University of California, Irvine, 2015

Professor Athanasios Sideris, Chair

A sequential quadratic programming method is proposed for solving nonlinear optimal control problems subject to general path constraints including mixed state-control and state only constraints. The proposed algorithm further develops on the approach proposed in [1] with objective to eliminate the use of a high number of time intervals for arriving at an optimal solution. This is done by introducing an adaptive time discretization to allow formation of a desirable control profile without utilizing a lot of intervals. The use of fewer time intervals reduces the computation time considerably. This algorithm is further used in this thesis to solve a trajectory planning problem for higher elevation Mars landing.

1 INTRODUCTION

An optimal control problem (OCP) consists of a set of differential equations describing the path of state variables given the control variables and a cost function to minimize. The optimal control problem discussed in this work is of the general Bolza form, in which the cost function is divided into Mayer and Lagrange parts. The Mayer part seeks to minimize the final cost while the Lagrange part minimizes the running cost. An OCP is subject to general path constraints including mixed state-control and state-only constraints. To solve a nonlinear optimal control problem (NL-OCP), a sequential quadratic programming (SQP) approach is used here, in which the generally difficult NL-OCP is replaced with a sequence of more manageable linear quadratic optimal control subproblems subject to linear state/control constraints. The subproblems solved at each step of the algorithm inherit the structure of the nonlinear optimal control problem (NL-OCP). At each iteration, a constrained Linear Quadratic Optimal Control subproblem is obtained by taking a quadratic approximation of the cost function and linearizing the dynamical equations and state/control constraints about the current solution candidate. The solution to each of these subproblems then provides a search direction that can be used to step towards the optimal solution of NL-OCP.

The adaptive time approach introduced in this thesis aims at making the algorithm in [1] efficient by reducing the number of time intervals required to solve an optimal control problem. It is done by introducing time as an additional state and the rate of change of time with respect to a new independent time variable as an additional control. This approach optimizes the spacing between time intervals to devise a control profile with the help of fewer intervals. Consequently, the computational time required to achieve an optimal solution is reduced considerably.

This algorithm is applied to solve a trajectory planning problem for a Mars entry vehicle for landing at higher altitudes [2]. The entry guidance task for Mars Science Laboratory (MSL) type vehicles is to deliver the lander accurately to the supersonic parachute deployment point within the parachute altitude, dynamic pressure and Mach number

constraints. The higher the required landing site elevation is, the lower the atmospheric density the lander must maneuver in during the final stage of entry. The trajectory optimization is used to understand the nature of the bank angle profile that leads to the maximum parachute deployment altitude and the trade-off between maximizing final altitude and preserving control authority near the end of entry.

This problem is formulated mainly in two different ways to get the optimal solution. The first step in both the formulations is to define the dynamic equations, constraints and the cost function. Then the algorithm is coded and implemented in MATLAB to plot the state trajectories and bank angle profile. The optimal bank angle profile is obtained with and without the use of adaptive time approach to compare the number of time intervals required to reach an optimal solution. In the first problem formulation, the dynamic equations and the cost functions are kept the same as described in [2]. In the second problem formulation, two extra states are introduced to convexify the Lagrange part of the cost function.

Units of length are taken as km to minimize numerical ill-conditioning and the problem is treated as a fixed final time problem. The system dynamic equations and constraint equations are assumed to be continuously differentiable and the cost twice continuously differentiable with respect to their arguments.

1.1 Nomenclature

r =Radial distance from Mars center

r_p =Mars radius

h =Altitude

θ =Longitude

ϕ =Latitude

V =Velocity

γ =Flight path angle

ψ =Heading angle

σ =Bank angle

ω_p =Mars angular velocity

C_D, C_L =Drag Coefficient, Lift coefficient

L, D =Lift acceleration, Drag acceleration

S =Vehicle Surface Area

m =Vehicle Mass

g =Gravitational acceleration

1.2 Planet Model

Mars is assumed to be spherical and a windless planet. Altitude h is computed as $h = (r - r_p)\ km$, with $r_p = 3397\ km$. The gravitational acceleration is modeled by $g = \frac{\mu}{r^2}\ km/s^2$, with gravitational parameter $\mu = 42409\ km^3/s^2$. The planet rotates around its north pole at a rate of $\omega_p = 7.095 \times 10^{-5}\ rad/s$. The atmospheric density is modeled by the exponential function of $h\ (km)$ as [2]

$$\rho(h) = 0.013\times10^{9}\exp(-9.20\times10^{-5}h-1.94\times10^{-11}h^2-7.51\times10^{-15}h^3+4.20\times10^{-20}h^4)kg/km^3$$

The aerodynamic forces on the vehicle is divided into lift and drag force. The lift and drag accelerations are defined by

$$L = \frac{1}{2}\rho V^2 \frac{S}{m} C_L km/s^2$$
$$D = \frac{1}{2}\rho V^2 \frac{S}{m} C_D km/s^2$$

Table 1: Entry vehicle parameters [2]

Parameter	Value	Units
S	$15.9e{-}6$	km^2
m	2804	kg
C_L	0.62	-
C_D	1.92	-

2 PROBLEM FORMULATIONS

2.1 Basic Formulation

2.1.1 System Dynamics

The entry vehicle dynamics are defined with respect to planet-fixed coordinate frame. The state vector is taken as

$$x = \begin{bmatrix} h & \theta & \phi & V & \gamma & \psi \end{bmatrix}^T \tag{1}$$

(V, γ, ψ) define the planet-relative velocity vector. The heading angle ψ is the angle between the projection of the velocity on the horizontal plane and the east direction (see Figure 1); the convention for the heading angle is $\psi = 0°$ for East and $\psi = 90°$ for North. The flight path angle γ is the angle of the velocity vector with the local horizontal plane and is defined positive when the velocity is above the plane or the vehicle is gaining altitude and negative when the velocity is below the plane or the vehicle is losing altitude.

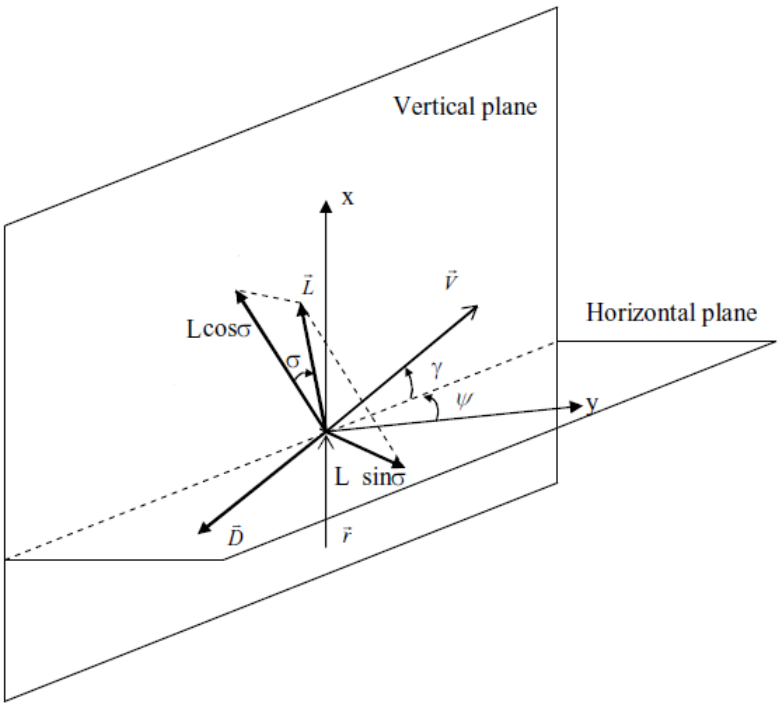

Figure 1: Vehicle Coordinate System

To have increased control authority at the final part of entry, the final flight path angle γ_f, should be reduced in magnitude [2]. Minimizing the magnitude of γ_f implies less steep descent. The rotation angle of lift vector about the planet-relative velocity vector, is defined as the bank angle σ. This angle is taken as the control u. When the lift vector is in the vertical plane defined by the position and velocity vectors, the bank angle is zero.

Equations of motion

The entry vehicle center of mass dynamics are [3]

$$\left.\begin{aligned}
\dot{h} &= V sin\gamma \\
\dot{\theta} &= \frac{V}{r cos\phi} cos\gamma cos\psi \\
\dot{\phi} &= \frac{V}{r} cos\gamma sin\psi \\
\dot{V} &= -D - g sin\gamma \\
\dot{\gamma} &= \frac{1}{V}\left[L cos\sigma - \left(g - \frac{V^2}{r}\right) cos\gamma\right] + 2\omega_p cos\psi cos\phi \\
\dot{\psi} &= -\frac{1}{V cos\gamma}\left(L sin\sigma + \frac{V^2}{r} cos^2\gamma cos\psi tan\phi\right) + 2\omega_p (tan\gamma sin\psi cos\phi - sin\phi)
\end{aligned}\right\} \quad (2)$$

where time derivatives are denoted by a dot over the variable. These equations define the behavior of the states during the trajectory.

2.1.2 Constraints

(i) Final State Constraints

Parachute Deployment Constraints

A supersonic disk-band-gap parachute is assumed for this problem [4]. It can be deployed at dynamic pressures between 300 and 800 Pa and Mach numbers between 1.4 and 2.2. Converting these constraints into velocity and altitude constraints and imposing the minimum deployment altitude of 6 km relative to r_p, the feasible parachute deployment box is shown in Figure 2.

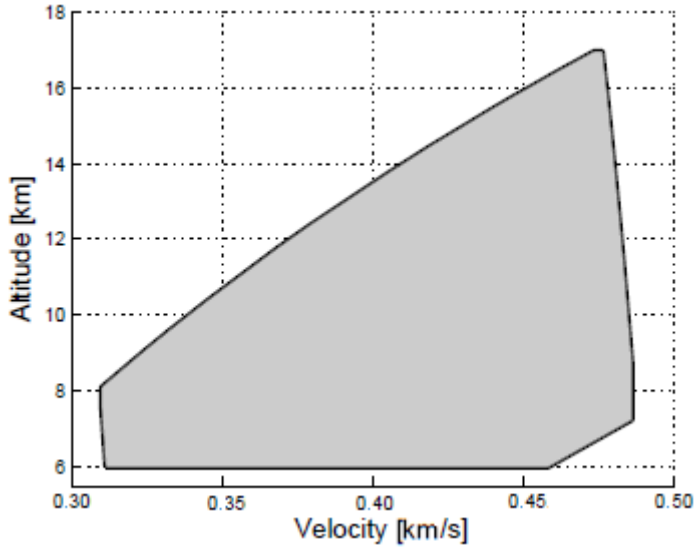

Figure 2: Disk-band-gap parachute deployment box

Taking a linear approximation of the boundaries of parachute deployment box, we get

$$\left.\begin{array}{l} h_f - 6 > 0 \\ V_f - 0.309 > 0 \\ V_f - 0.480 < 0 \\ 40.32V_f - h_f - 12.42742 < 0 \\ 54.27V_f - h_f - 8.77744 > 0 \end{array}\right\} \tag{3}$$

where the subscript 'f' denotes final time. The time t is considered to be the independent variable with $t \in [0, t_f]$, and t_f fixed. The value of $t_f = 300$ is obtained from results published in [2]. This problem can also be solved as a free final time by removing the constraint on final time.

Surface Position Constraints

The radial projection of the vehicle state onto the surface is defined by the downrange DR and crossrange CR or by the longitude θ and latitude ϕ. The plane in which the initial position and velocity vector lies is defined as a great circle plane (see Figure 3). The reference great circle is at the surface radius in this plane. The *crossrange* is defined as the shortest distance from the target position to the great circle, with positive values

when the final position is to the right looking down the surface. The *downrange* is the arc length on the great circle between the initial position and the point from where the crossrange is measured.

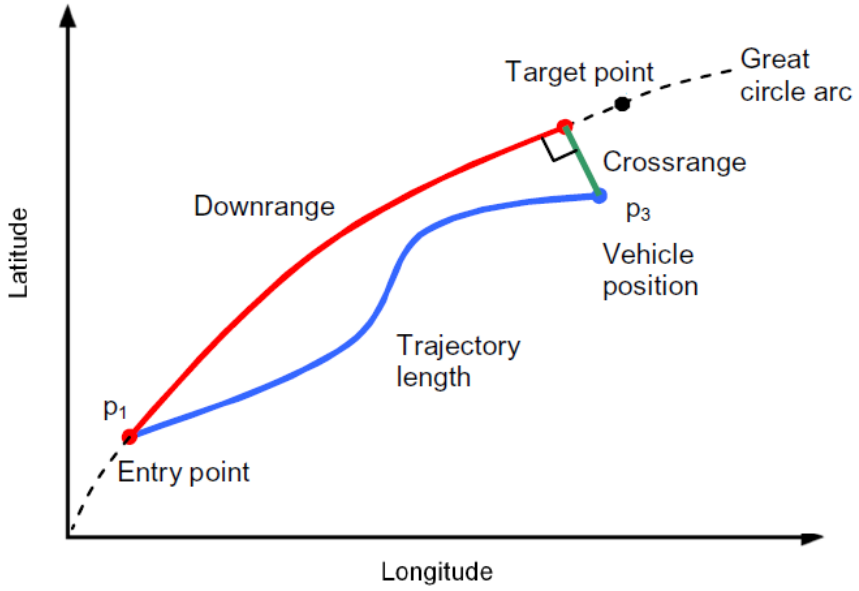

Figure 3: Downrange, crossrange and great circle arc

For every value of CR and DR, the corresponding value of θ_f and ϕ_f can be be obtained by [5]

$$\left.\begin{aligned} \phi_f &= arcsin\left(cos\left(\xi - \psi_0 + \frac{\pi}{2}\right) cos\phi_0 sinLF + sin\phi_0 cosLF\right) \\ \theta_f &= \theta_0 + arcsin\left(\frac{sin(\xi - \psi_0 + \frac{\pi}{2}) sinLF}{cos\phi_f}\right) \\ with & \\ LF &= arccos\left(cos\frac{DR}{r_p} cos\frac{CR}{r_p}\right) \\ \xi &= arcsin\left(\frac{sin\frac{CR}{r_p}}{sinLF}\right) \end{aligned}\right\} \quad (4)$$

where θ_0, ϕ_0 and ψ_0 are initial surface longitude, latitude and heading angle.

Projecting the set of all points on feasible trajectories onto the downrange-crossrange-

altitude space yields the reachable position space set. The lower boundary is a subset of a plane at the minimum altitude 6.0 km at which the parachute can be deployed. The region inside this boundary is called the "landing footprint" of the vehicle in the downrange-crossrange plane. In order to determine the boundary of the landing footprint, optimal control problems are solved [6-8]. Focusing on the central region of the footprint, we get

$$\left.\begin{aligned} Downrange &= 800km \\ Crossrange &= 0km \\ \phi_f &= -0.7203rad \\ \theta_f &= -1.2576rad \end{aligned}\right\} \tag{5}$$

(ii) Control Constraints

To prevent negative lift, the following control constraints are imposed.

$$\left.\begin{aligned} \sigma &< 90^\circ \\ \sigma &> -90^\circ \end{aligned}\right\} \tag{6}$$

The constraints on $\dot{\sigma}$ and $\ddot{\sigma}$ are not imposed in this formulation.

(iii) Initial Constraints

The initial entry state, $x(0) = x_0$ used for all the numerical results presented here, is given by

$$\begin{bmatrix} h_0 & \theta_0 & \phi_0 & V_0 & \gamma_0 & \psi_0 \end{bmatrix}^T = \begin{bmatrix} 143km & -90.07^\circ & -43.90^\circ & 6.082km/s & -15.50^\circ & 4.99^\circ \end{bmatrix}^T \tag{7}$$

2.1.3 Cost Function

(i) Mayer Cost

The Mayer Cost is a weighted combination of the final altitude and the final flight path

angle as follows:

$$M(t_f, x_f) = -k_h h(t_f) + k_\gamma \gamma(t_f)^2 \tag{8}$$

The first term in the equation (8) seeks to maximize the final altitude, while the second term seeks to minimize the final flight path angle magnitude. k_h and k_γ are positive weighting constants with values $k_h = 5km^{-1}$ and $k_\gamma = 91.4(180/pi)^2 rad^{-2}$.

This Mayer cost is modified to include a penalty term that consists of penalty parameters multiplied by a measure of the violation of the final constraints. This facilitates approaching the solution faster.

(ii) Modified Mayer Cost

$$M(t_f, x_f) = -k_h h(t_f) + k_\gamma \gamma(t_f)^2 + 0.5k_1(\theta(t_f) + 1.2576)^2 + 0.5k_2(\phi(t_f) + 0.7203)^2 \tag{9}$$

k_1 and k_2 are the penalty parameters with values $k_1 = k_2 = 1000$.

(iii) Lagrange Cost (non-convex function)

In the following Lagrange Cost, values of the bank angle between $-\sigma_{min}$ and $\sigma_{min} = 18.2^\circ$ are penalized to reserve 5% of the maximum vertical lift for the guidance (see Figure 4). This soft constraint allows bank reversals crossing $\sigma = 0$, but not prolonged use of small bank angles.

$$L(\sigma(t)) = a[tan^{-1}(b(-\sigma(t) + \sigma_{min})) + tan^{-1}(b(\sigma(t) + \sigma_{min}))] \tag{10}$$

The parameters a and b control the height and smoothness of the function and are taken as $a = 90$ and $b = 500$.

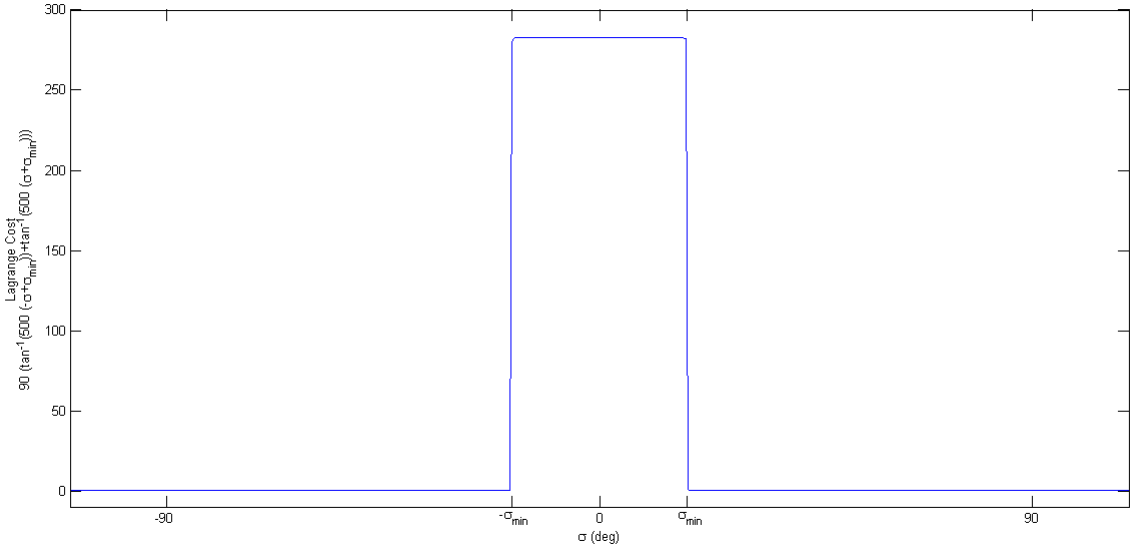

Figure 4: Non-convex bell shaped function used in Lagrange cost of basic formulation

This Lagrange cost is modified to limit the magnitude of the flight path angle from approaching zero. This is done by penalizing small values of γ for better control authority during the trajectory.

(iv) Modified Lagrange Cost

$$L = a[tan^{-1}(b(-\sigma(t) + \sigma_{min})) + tan^{-1}(b(\sigma(t) + \sigma_{min}))] + 0.5c\gamma^2 \tag{11}$$

The penalty on γ is taken to be $c = 10,000$.

2.1.4 Solution

The optimal control problem formulated above is solved using the algorithm described in [1]. This algorithm is briefly described next.

Consider a non-differentiable penalty function:

$M = J + \rho V$ where J is the cost function of the optimal control problem, V represents maximum constraint violation, and ρ>0 is a parameter for trading off J and V.

(1) Choose an arbitrary control sequence and $\rho = \rho_0 > 0$ for the first iteration. Com-

pute the state trajectory corresponding to the current control by simulating the dynamic equations.

(2) Formulate the Linear Quadratic Optimal Control Problem (LQ-OCP) by taking a second order approximation of the cost function and linearizing the dynamics and constraints along the current solution. An auxiliary scalar state ξ is introduced in (LQ-OCP) to ensure feasibility. This state is added to the quadratic approximation of the cost with the penalty parameter ρ. Now two cases are included in the (LQ-OCP) formulation. First the solution of (LQ-OCP) is attempted with initial $\xi = 0$ in case (a). If this problem is feasible, the current control becomes the solution of LQ-OCP. If this problem is infeasible because the linearized constraints are infeasible, (LQ-OCP) is solved with leaving initial $\xi \geq 0$ unspecified, as an optimization variable in case (b). The algorithm then sets the value of ξ_0 equal to the maximum linearized constraint violation. If the constraint violation is less than the termination tolerance, the current control is taken as the solution to LQ-OCP. The NL-OCP is considered infeasible if the constraint violation is not reduced to below the tolerance within the specified iterations.

(3) The solution from LQ-OCP gives a descent search direction to compute the next control sequence where the step length is obtained with the help of Armijo's rule.

These steps are repeated until case (a) becomes valid for LQ-OCP and First Order Necessary conditions for optimality are satisfied for Nonlinear Optimal Control Problem (NL-OCP).

Let N be the number of time discretization intervals. These intervals are equally spaced in this formulation. The algorithm is run multiple times with different values of N to study the variation of states and to find the minimum number of intervals required to reach the optimal cost. The initial control sequence used here is an array of length N with each element equal to $-\pi/360$. The initial value of ρ is chosen to be 100.

Table 2: Comparison of final values of states for different N in basic formulation

State Variable	h(km)	θ°	ϕ°	V(km/s)	γ°	ψ°
Initial Value	143	-90.07	-43.90	6.082	-15.50	4.99
Final Value ($N = 9$)	11.2472	-72.0552	-41.2701	0.48	-19.1392	22.7786
Final Value ($N = 50$)	14.2163	-72.0552	-41.2702	0.48	-17.264	19.7495

The minimum number of intervals required by the algorithm to find a feasible solution in this case is 9. For N equals 9, the final altitude achieved is 11.2472 km and the final flight path angle magnitude is 19.1392°. It can be seen from Table 2 that as the number of time intervals increases to 50, the algorithm is able to achieve a higher final altitude of 14.2163 km and a lower final flight path angle magnitude of 17.264°. Figure 5 and Figure 6 show the variation of all the states along the trajectory for different number of time intervals.

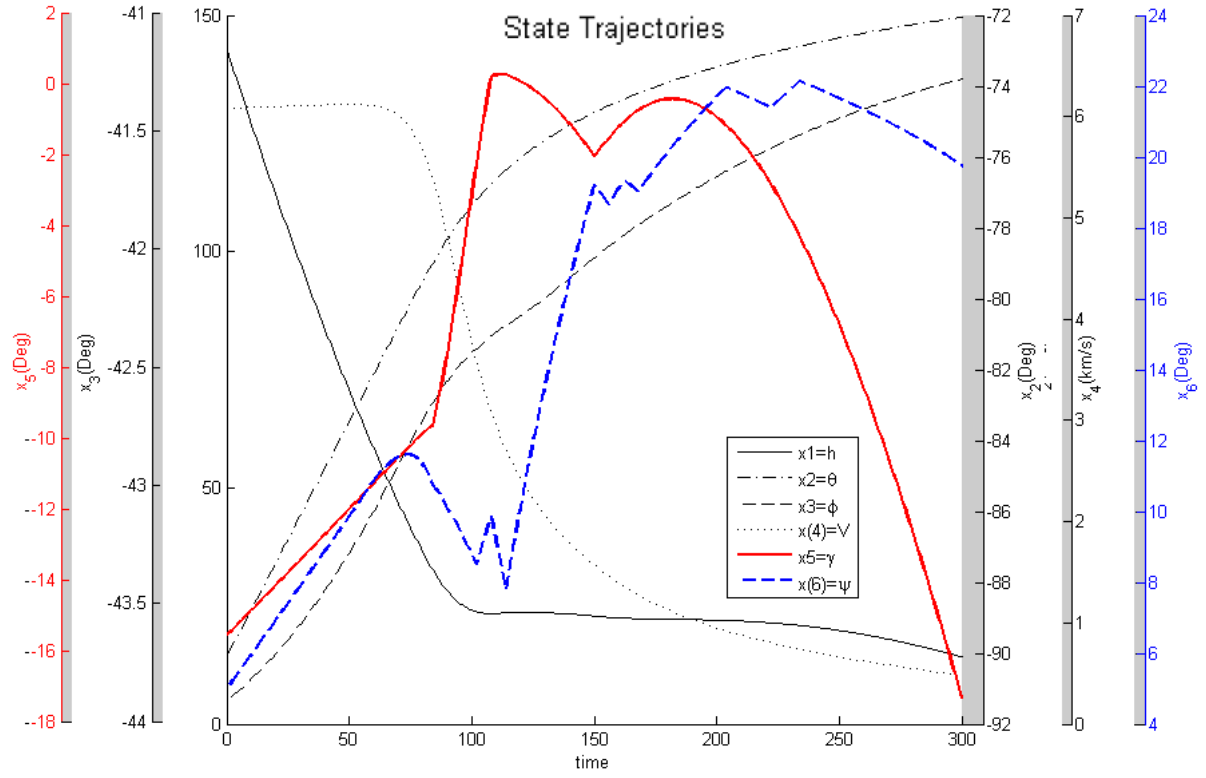

Figure 5: Variation of state variables along the trajectory for N=50 in basic formulation

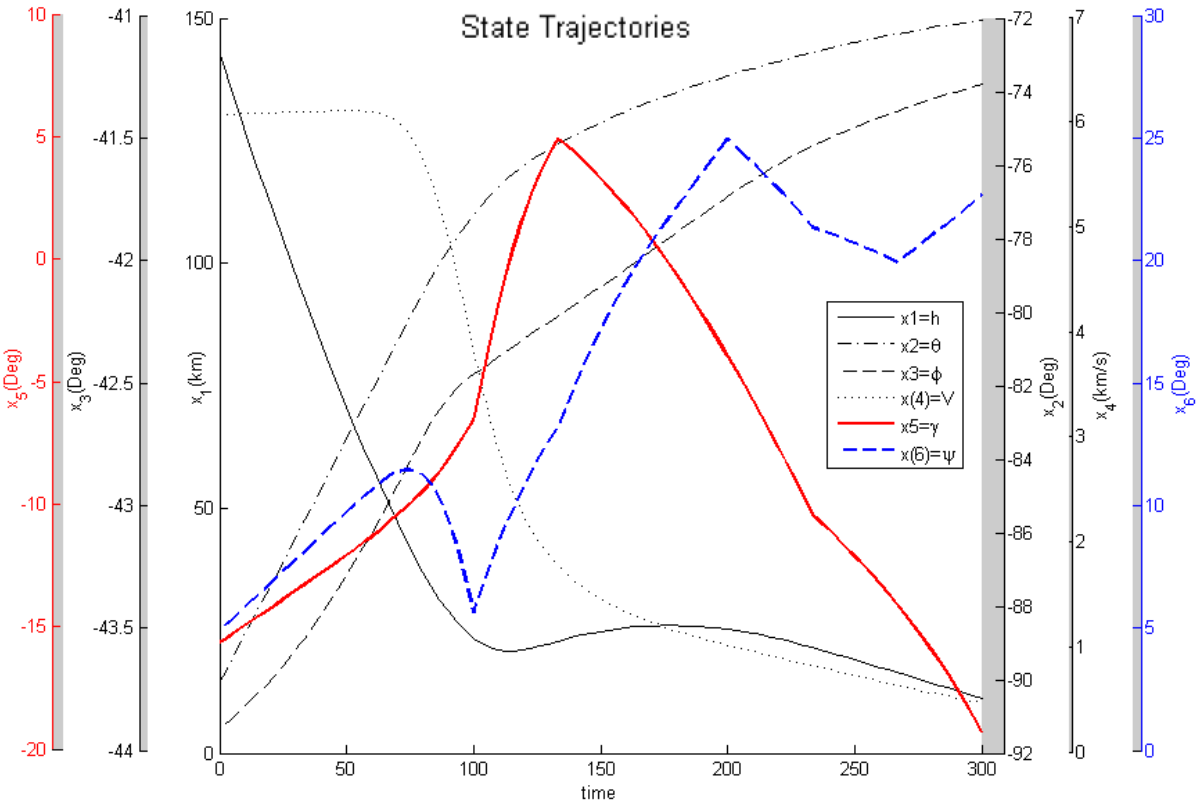

Figure 6: Variation of state variables along the trajectory for N=9 in basic formulation

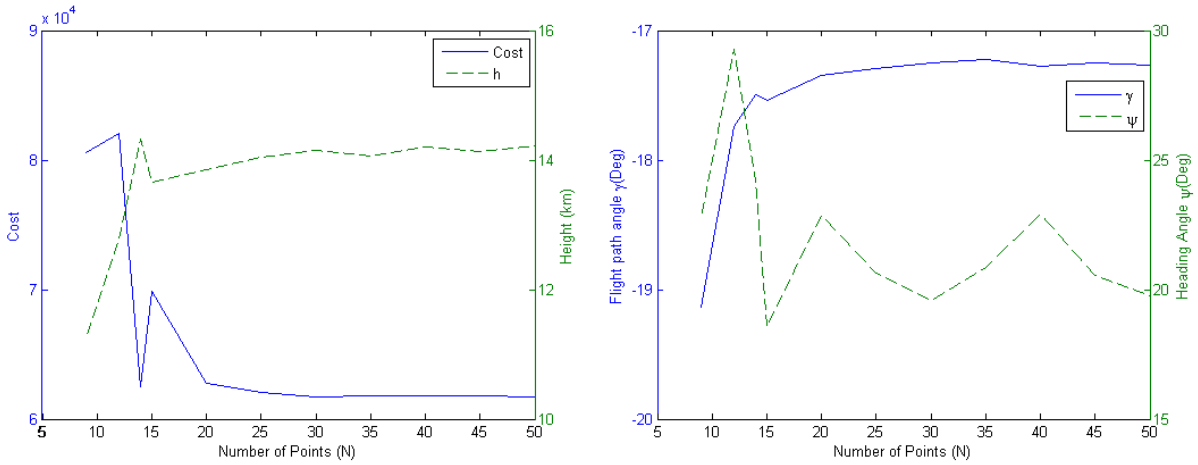

Figure 7: Variation of states and cost with N in basic formulation

Figure 7 shows the variation of optimal cost and states with the number of time intervals considered. The states that are not plotted here do not show considerable variation. It can be inferred that 30 intervals are sufficient to reach an optimal solution in this case.

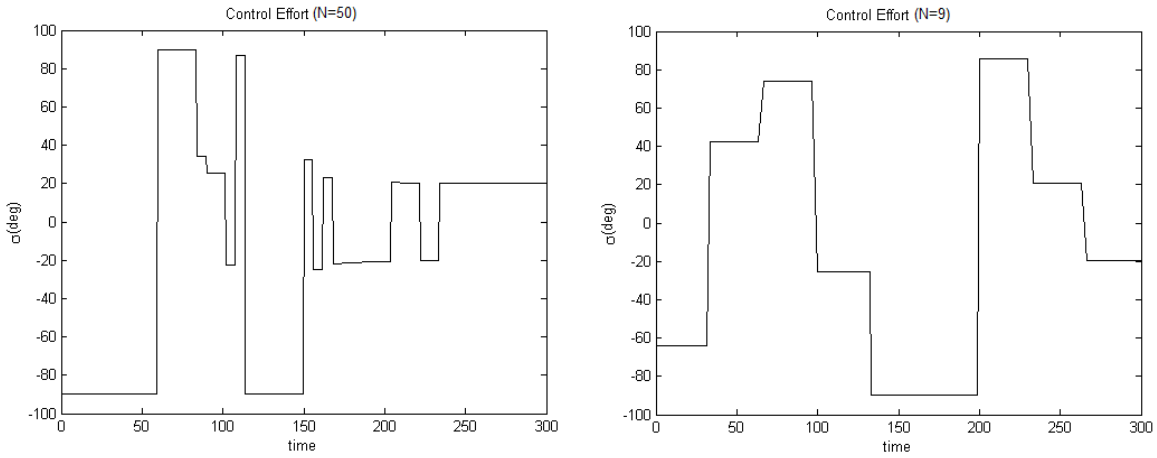

Figure 8: Optimal bank angle profiles in basic formulation

Figure 8 shows the comparison of piecewise constant profiles of σ for large and small N. In this formulation, the transition of control from one value to another occurs at equal time intervals. It is worth mentioning here that the profile would look completely different if the constraints on $\dot{\sigma}$ and $\ddot{\sigma}$ are imposed as the transition rate of control from one value to another will be restricted in that case.

2.2 Alternative formulation

In this formulation, two new states are introduced. The Lagrange cost is made a convex function with the help of these states to establish better results.

2.2.1 System Dynamics

The state vector is taken as

$$x = \begin{bmatrix} h & \theta & \phi & V & \gamma & \psi & z_1 & z_2 \end{bmatrix}^T \tag{12}$$

where z_1 and z_2 are the added states defined as

$$\left.\begin{aligned} z_1 &= sin\sigma \\ z_2 &= cos\sigma \end{aligned}\right\} \tag{13}$$

In this case, the bank angle rate $\dot{\sigma}$ is used as the control:

$$u = \dot{\sigma} \tag{14}$$

Since the control profile generated by the algorithm is piecewise constant, the bank angle profile will be piecewise linear. Two new dynamic equations are added in the equations of motion by taking derivative of z_1 and z_2 with respect of time t.

Equations of motion

Set of equations (2), and

$$\left.\begin{aligned} \dot{z}_1 &= \frac{d\sigma}{dt} cos\sigma = \dot{\sigma} cos\sigma \\ \dot{z}_2 &= -\frac{d\sigma}{dt} sin\sigma = -\dot{\sigma} sin\sigma \end{aligned}\right\} \tag{15}$$

2.2.2 Constraints

(i) Final State Constraints

Set of equations (3) and (5).

(ii) Path Constraints

The value of sigma should lie between $-90°$ and $90°$ to prevent negative lift, as mentioned earlier. Hence $cos\sigma$ will always be positive.

$$z_2 > 0 \tag{16}$$

(iii) Control Constraints

The bank angle rate is limited to $\mid \dot{\sigma} \mid \leq \dot{\sigma}_{max}$, with $\dot{\sigma}_{max} = 20deg/s$.

$$\left.\begin{array}{l} \dot{\sigma} < 20deg/s \\ \dot{\sigma} > -20deg/s \end{array}\right\} \tag{17}$$

The constraint on $\ddot{\sigma}$ is not imposed in this formulation.

(iv) Initial Constraints

Set of equations (7), and leaving the initial values of z_1 and z_2 free for the algorithm to optimize in order to arrive at the solution.

2.2.3 Cost Function

(i) Modified Mayer Cost (9)

$$M(t_f, x_f) = -k_h h(t_f) + k_\gamma \gamma(t_f)^2 + 0.5k_1(\theta(t_f) + 1.2576)^2 + 0.5k_2(\phi(t_f) + 0.7203)^2$$

with $k_h = 5km^{-1}$, $k_g = 91.4(180/pi)^2 rad^{-2}$ and $k_1 = k_2 = 1000$.

(ii) Modified Lagrange Cost

The Lagrange cost is modified to make it a convex function for better working of the algorithm. It places a penalty on values of z_2 between 0 and $cos18.2°$. Soft constraints on the control $u = \dot{\sigma}$ are also imposed here for faster convergence.

$$L = exp(b_1(z_2 - cmin)) + exp(c_1(\dot{\sigma} - umin)) + exp(-c_1(\dot{\sigma} - umax)) \tag{18}$$

The penalty parameters are chosen as

$b_1 = 120$, $c_1 = 10$, $cmin = cos(18.2pi/180)$, $umin = -pi/9$ and $umax = pi/9$

(iii) Comparison of basic and modified Lagrange function

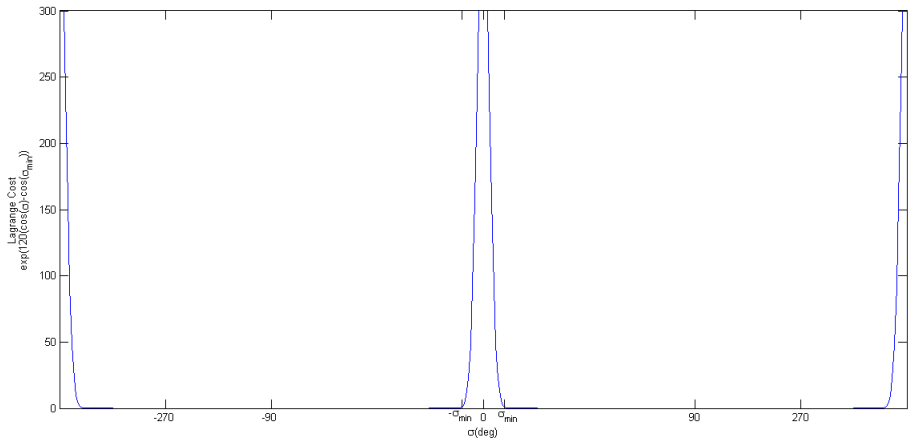

Figure 9: Convex function used in Lagrange cost of alternative formulation

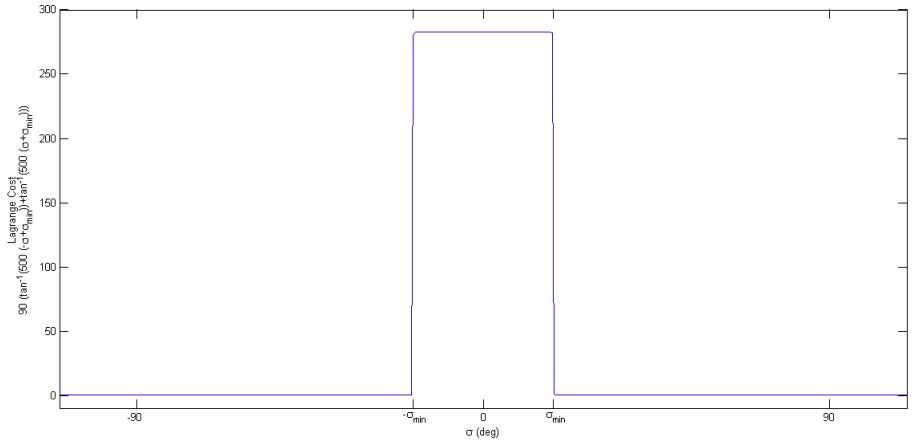

Figure 10: Non-convex function used in Lagrange cost of basic formulation

In both the costs defined in (11) and (18), the value of σ is penalized between $-18.2°$ and $18.2°$ (see Figure 9 and Figure 10). But the function used in alternative formulation has a convex profile with respect to z_2 to facilitate the quadratic approximation of the cost within the algorithm.

2.2.4 Solution

The N time discretization intervals are equally spaced in this formulation also. The initial control sequence used here is an array of length N with each element equal to zero. The initial value of ρ is chosen to be 1000.

The minimum number of intervals required in this formulation to find a feasible solution is 10. For N equals 10, the final altitude achieved is 12.6721 km and the final flight path angle magnitude is 22.5974°. Figure 11 and Figure 12 show the variation of all the states along the trajectory for different number of time intervals.

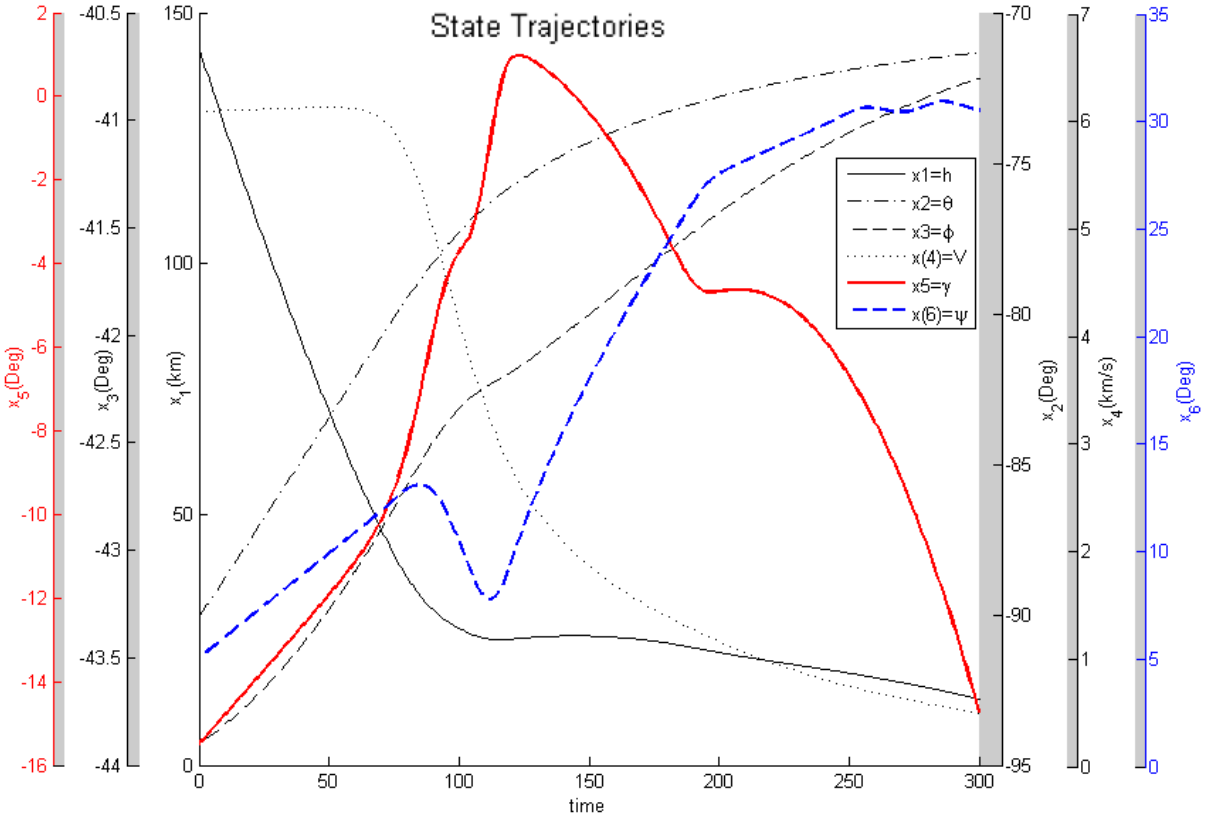

Figure 11: Variation of state variables along the trajectory for N=50 in alternative formulation

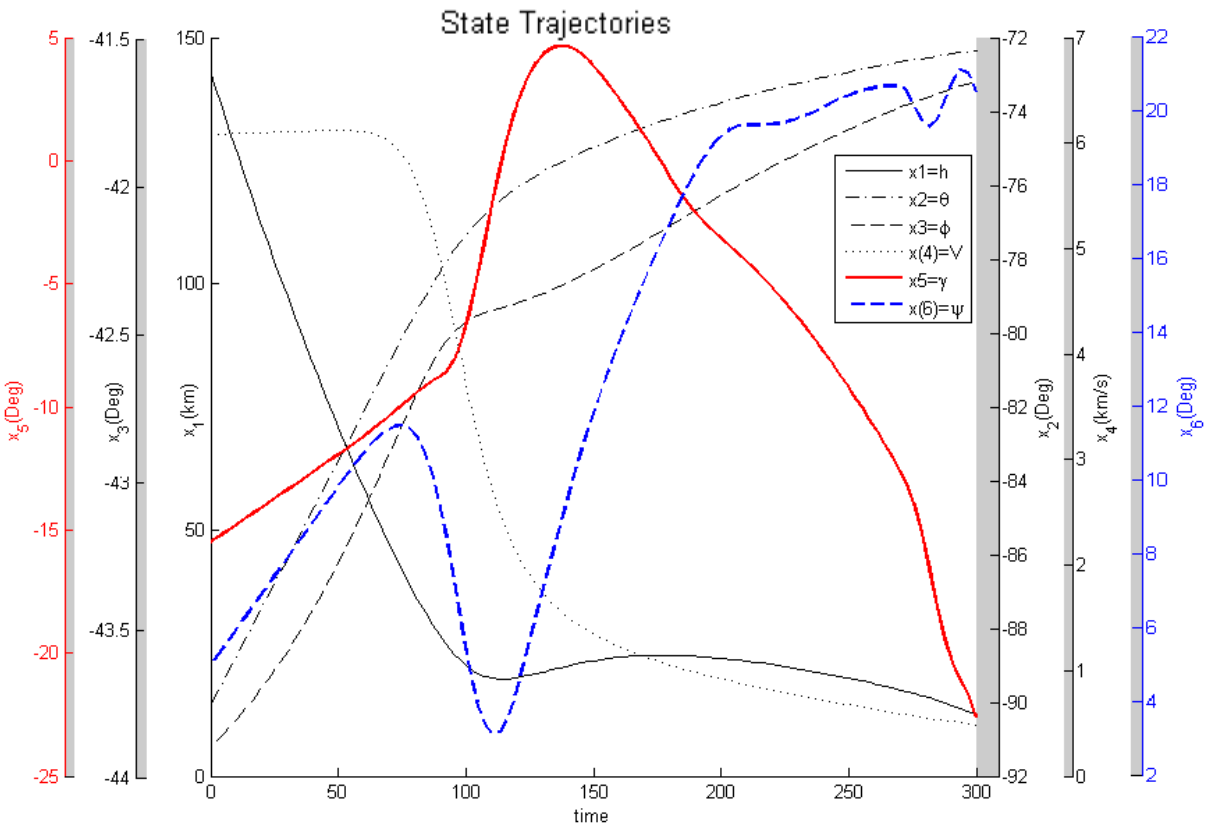

Figure 12: Variation of state variables along the trajectory for N=10 in alternative formulation

Table 3: Comparison of final values of states for different N in alternative formulation

State Variable	h(km)	θ°	ϕ°	V (km/s)	γ°	ψ°
Initial Value	143	-90.07	-43.90	6.082	-15.50	4.99
Final Value ($N = 10$)	12.6721	-72.3485	-41.6473	0.4814	-22.5974	20.5492
Final Value ($N = 50$)	13.1870	-71.3275	-40.8052	0.4850	-14.7357	30.4514

It can be seen from Table 3 that as the number of time intervals increases to 50, the algorithm is able to achieve a higher final altitude of 13.1870 km and a lower final flight path angle magnitude of 14.7357°. Thus, it is able to achieve a better control authority at the end than in the basic formulation.

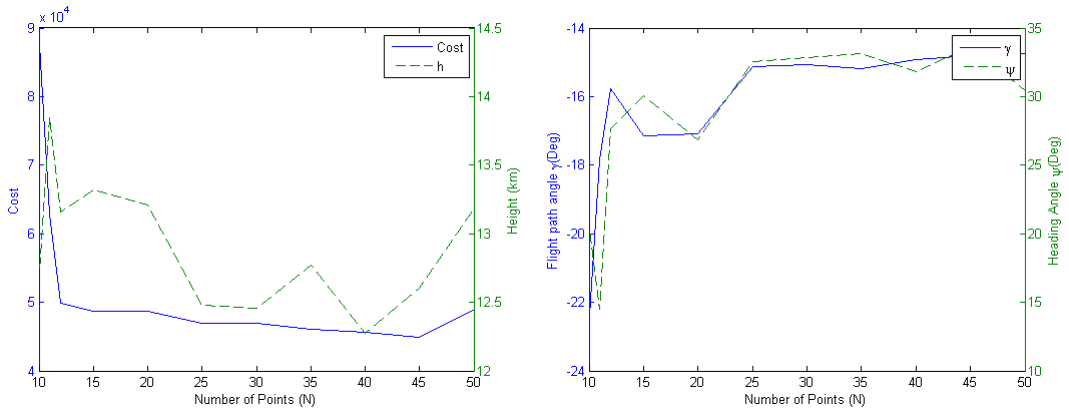

Figure 13: Variation of states and cost with N in alternative formulation

Figure 13 shows the variation of optimal cost and states with the number of time intervals considered. The states not plotted here do not show considerable variation. It can be inferred that 30 intervals are sufficient to reach the optimal solution in this case. The comparison of piecewise linear bank angle profiles for large and small value of N is shown in Figure 14.

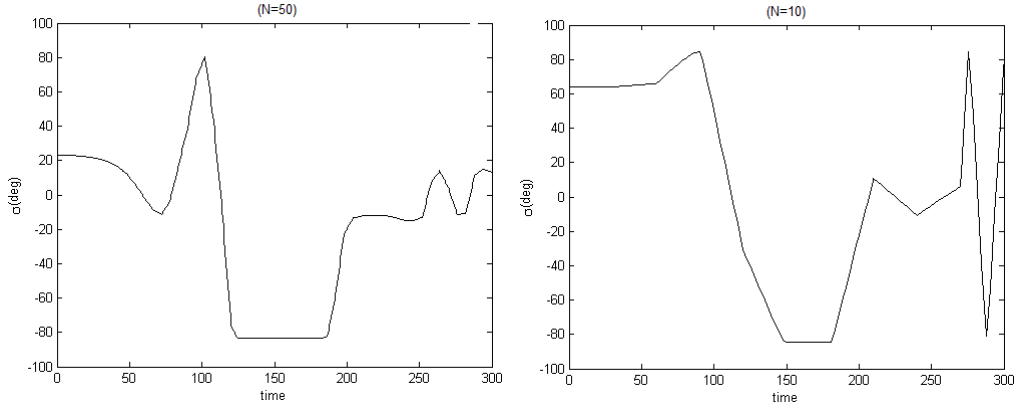

Figure 14: Optimal bank angle profiles in alternative formulation

It can be seen in Figure 14 that by using a higher number of intervals, the resulting profile behaves similarly to a piecewise constant profile desired for the problem. Also, due to the linear nature of the bank angle profile, the bank angle spends more time in the penalized zone during transition from one value to another which is not desirable.

3 ADAPTIVE TIME APPROACH

An adaptive time discretization approach is introduced here to solve an optimal control problem with fewer number of time intervals. In this approach, time t is considered as an additional state variable which is optimized with respect to another independent time variable τ. So now, algorithm is given the freedom to optimize spacing between the time intervals. The intervals will be closely spaced where greater authority over the control profile is required. This judicious use of spacing facilitates achieving desired a control profile with fewer number of time intervals.

A mathematical analysis of this approach is given next.

Consider the dynamic equation

$$\dot{x}(t) = f(x(t), u(t)) \tag{19}$$

and express time t as a state variable that depends on τ such that

$$\frac{dt(\tau)}{d\tau} = W(\tau) \tag{20}$$

where $W(\tau)$ is the additional control introduced.

Let

$$\tau = [0, \frac{1}{N}, ...\frac{k}{N}, \frac{k+1}{N}, ...1] \tag{21}$$

and take $W(\tau) \equiv W_k$ in $\tau \in [\frac{k}{N}, \frac{k+1}{N}]$. The spacing between t_k and t_{k+1} is controlled by the value of W_k.

Let $x(t(\tau))$ and $u(t(\tau))$ be denoted as $\overline{x}(\tau)$ and $\overline{u}(\tau)$ respectively. The dynamic equations have to be modified to show the variation of states and control with respect to independent variable τ. So,

$$\dot{\overline{x}} = \frac{d\overline{x}(\tau)}{d\tau} = \frac{dx(t)}{dt}\frac{dt(\tau)}{d\tau} = W(\tau)f(\overline{x}(\tau), \overline{u}(\tau)) \tag{22}$$

$$\dot{\overline{x}} = Wf(\overline{x}, \overline{u}) \tag{23}$$

where $\dot{\overline{x}}$ and $\dot{\overline{u}}$ represent $\frac{d\overline{x}(\tau)}{d\tau}$ and $\frac{d\overline{u}(\tau)}{d\tau}$ respectively.

The initial conditions conditions are taken as

$$\overline{x}(0) = x(t_0) \tag{24}$$

$$t(0) = t_0 \tag{25}$$

If t_f is fixed, then the constraint

$$t(1) = t_f \tag{26}$$

is used. Now, the augmented state becomes

$$x_a = \begin{bmatrix} \overline{x} \\ t \end{bmatrix} \tag{27}$$

and the augmented control becomes

$$u_a = \begin{bmatrix} \overline{u} \\ W \end{bmatrix} \tag{28}$$

To make sure that time keeps on increasing with respect to τ, an additional constraint has to be imposed, namely:

$$W{>}0 \tag{29}$$

Next, consider the Lagrange function

$$\mathcal{J}_L(t) = \int_{t_0}^{t} L(x(t'), u(t'))dt' \tag{30}$$

Differentiating (30) with respect to time yields:

$$\dot{\mathcal{J}}_L(t) = L(x(t'), u(t'))dt' \tag{31}$$

Let $\mathcal{J}_L(t(\tau))$ be denoted as $\overline{\mathcal{J}_L}(\tau)$. Differentiating (30) with respect to τ gives,

$$\dot{\overline{\mathcal{J}_L}}(\tau) = \frac{\overline{\mathcal{J}_L}(\tau))}{d\tau} = \frac{d\mathcal{J}_L(t)}{dt}\frac{dt(\tau)}{d\tau} \tag{32}$$

$$\dot{\overline{\mathcal{J}_L}}(\tau) = W(\tau)\dot{\mathcal{J}}_L(\overline{x}, \overline{u}) \tag{33}$$

Comparing (31) and (33), we get the modified Lagrange Cost

$$\overline{L} = W(\tau)L(\overline{x}, \overline{u}) \tag{34}$$

Finally, the Jacobians of the dynamics and the cost have to be modified as follows to include the additional state and control:

(i) System Dynamics

$$f_{x_a} = \begin{bmatrix} Wf_x & 0 \\ 0 & 0 \end{bmatrix}; f_{u_a} = \begin{bmatrix} Wf_u & f \\ 0 & 1 \end{bmatrix}$$

(ii) Lagrange Cost

$$L_{x_a} = \begin{bmatrix} WL_x & 0 \end{bmatrix}; L_{u_a} = \begin{bmatrix} WL_u & L \end{bmatrix}$$

$$L_{x_a x_a} = \begin{bmatrix} WL_{xx} & 0 \\ 0 & 0 \end{bmatrix}; L_{u_a u_a} = \begin{bmatrix} WL_{uu} & L_u^T \\ L_u & 0 \end{bmatrix}; L_{x_a u_a} = \begin{bmatrix} WL_{xu} & L_x^T \\ 0 & 0 \end{bmatrix}$$

(iii) Mayer Cost

$M_{x_a} = \begin{bmatrix} M_x & 0 \end{bmatrix} M_{x_a u_a} = \begin{bmatrix} M_{xx} & 0 \\ 0 & 0 \end{bmatrix}$ where the subscript $'x'$ and $'u'$ denote Jacobians with respect to state and control respectively. The subscript $'a'$ denotes augmented or

modified Jacobian.

3.1 Basic Formulation with Adaptive Time

3.1.1 System Dynamics

The modified state vector with t as an additional state becomes

$$x = \begin{bmatrix} h & \theta & \phi & V & \gamma & \psi & t \end{bmatrix}^T \tag{35}$$

and the modified control becomes

$$u = \begin{bmatrix} \sigma \\ W \end{bmatrix} \tag{36}$$

The dynamic equations of motion are modified according to (23)

$$\left.\begin{aligned}
\dot{h} &= WVsin\gamma \\
\dot{\theta} &= \frac{WV}{rcos\phi}cos\gamma cos\psi \\
\dot{\phi} &= \frac{WV}{r}cos\gamma sin\psi \\
\dot{V} &= W(-D - gsin\gamma) \\
\dot{\gamma} &= \frac{W}{V}[Lcos\sigma - (g - \frac{V^2}{r})cos\gamma] + 2W\omega_p cos\psi cos\phi \\
\dot{\psi} &= -\frac{W}{Vcos\gamma}(Lsin\sigma + \frac{V^2}{r}cos^2\gamma cos\psi tan\phi) + 2W\omega_p(tan\gamma sin\psi cos\phi - sin\phi) \\
\dot{t} &= W
\end{aligned}\right\} \tag{37}$$

3.1.2 Constraints

(i) Final State Constraints

Set of equations (3) & (5), and for a fixed final time problem:

$$t_f = 300 \tag{38}$$

(ii) Control Constraints

Set of equations (6) & (29)

(iii) Initial Constraints

Set of equations (7), and setting the initial time

$$t_0 = 0 \tag{39}$$

3.1.3 Cost Function

(i) Modified Mayer Cost (9)

$$M(t_f, x_f) = -k_h h(t_f) + k_\gamma \gamma(t_f)^2 + 0.5k_1(\theta(t_f) + 1.2576)^2 + 0.5k_2(\phi(t_f) + 0.7203)^2$$

with $k_h = 5km^{-1}$, $k_g = 91.4(180/pi)^2 rad^{-2}$, $k_1 = k_2 = 1000$.

(ii) Modified Lagrange Cost (11) & (34)

$$L = Wa[tan^{-1}(b(-\sigma(t) + \sigma_{min})) + tan^{-1}(b(\sigma(t) + \sigma_{min}))] + 0.5Wc\gamma^2$$

with $\sigma_{min} = 18.2°$, $a = 90$, $b = 500$, and the penalty on γ is $c = 10000$.

3.1.4 Solution

In this case, the spacing between N time intervals is optimized by the algorithm. The initial control sequence used here is an array of length N with each element equal to $-\pi/360$. The initial value of ρ is chosen to be 100,000. The minimum number of intervals required in this formulation to find a feasible solution is 4. For N equals 4, the final altitude achieved is 11.91 km and the final flight path angle magnitude is 22.3258°. It can be seen from Table 4 that as the number of time intervals increases to 15, the algorithm is able to achieve a higher final altitude of 14.4215 km and a lower final flight path angle magnitude of 17.5835°.

Table 4: Comparison of final values of states for different N in basic formulation with adaptive time

State Variable	h(km)	θ°	ϕ°	$V(km/s)$	γ°	ψ°
Initial Value	143	-90.07	-43.90	6.082	-15.50	4.99
Final Value ($N = 4$)	11.9102	-72.0551	-41.2702	0.48	-22.3258	-3.588
Final Value ($N = 15$)	14.4215	-72.0549	-41.2701	0.4801	-17.5835	17.2135

The performance of this adaptive time formulation matches with that of "basic formulation" (2.1) but without utilizing a lot of time intervals. Figure 15 and Figure 16 show the variation of all the states along the trajectory for different number of time intervals.

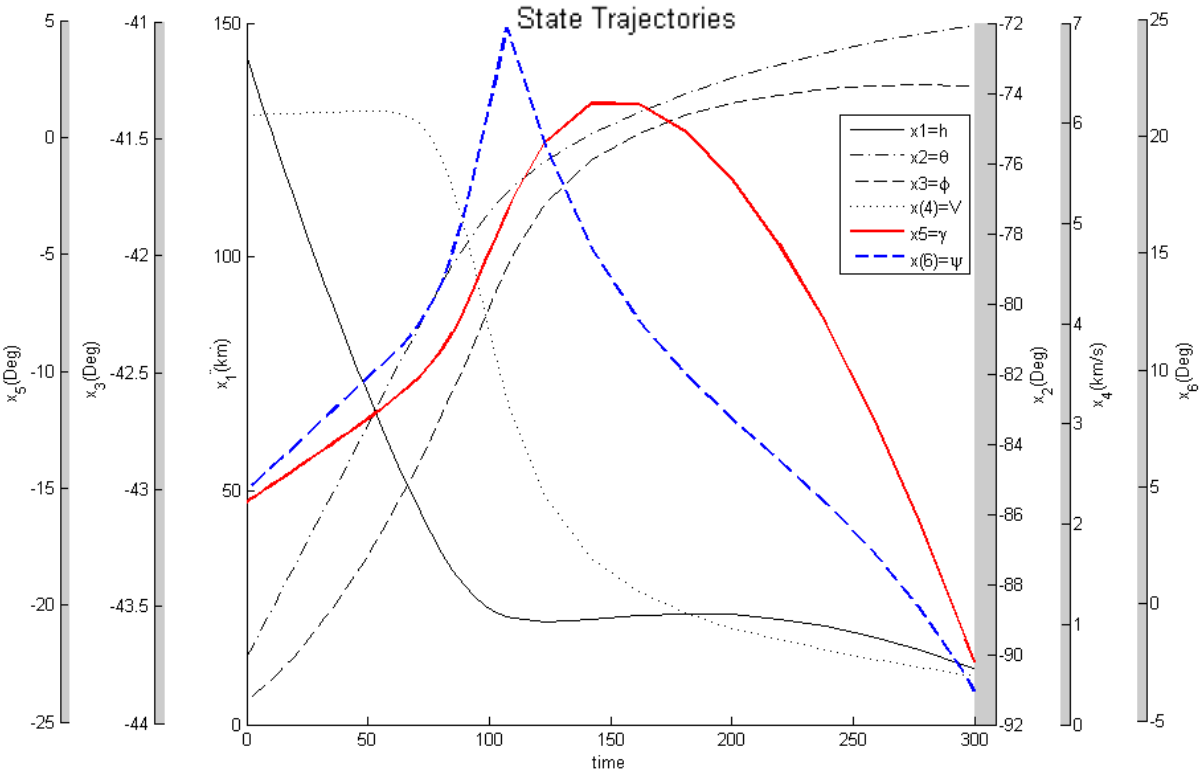

Figure 15: Variation of state variables along the trajectory for N=4 in basic formulation with adaptive time

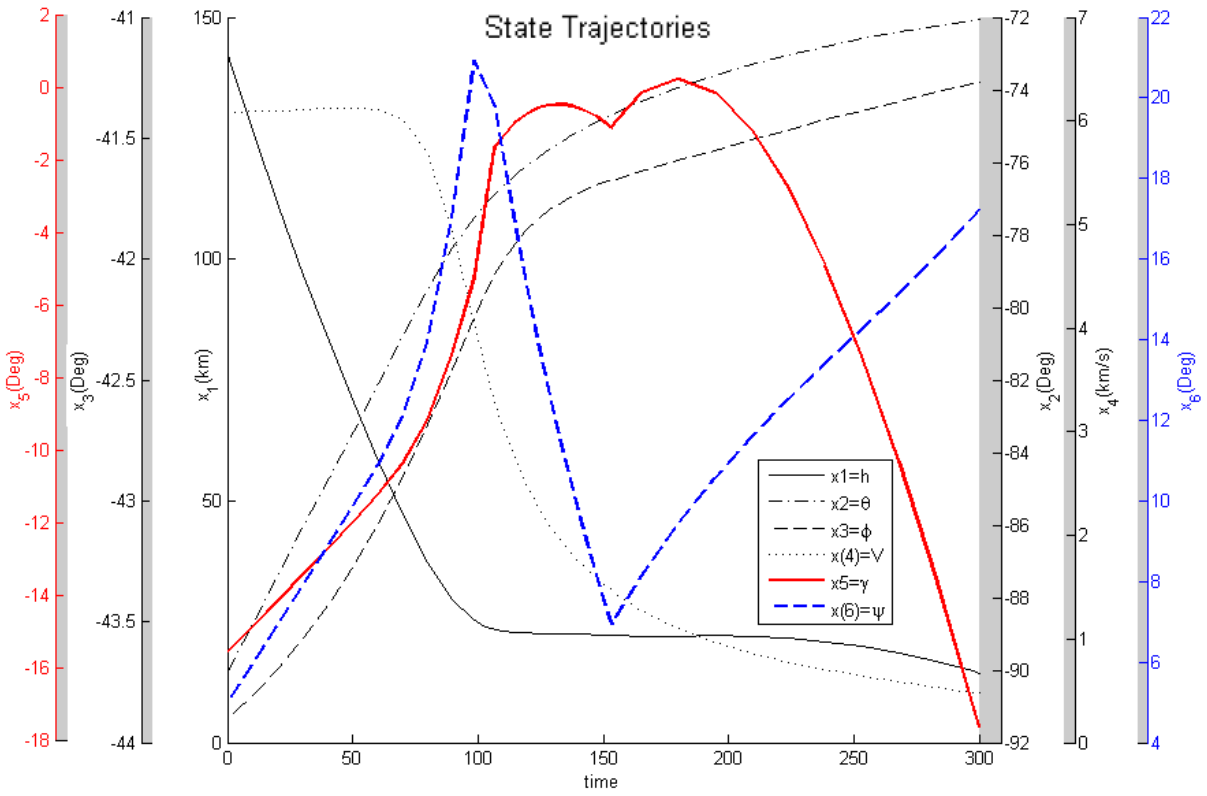

Figure 16: Variation of state variables along the trajectory for N=15 in basic formulation with adaptive time

The plots in Figure 17 show the variation of the optimal cost and states with the number of time intervals considered. The states not plotted here do not show considerable variation. It can be inferred that 8 intervals are sufficient to reach the optimal solution in this case.

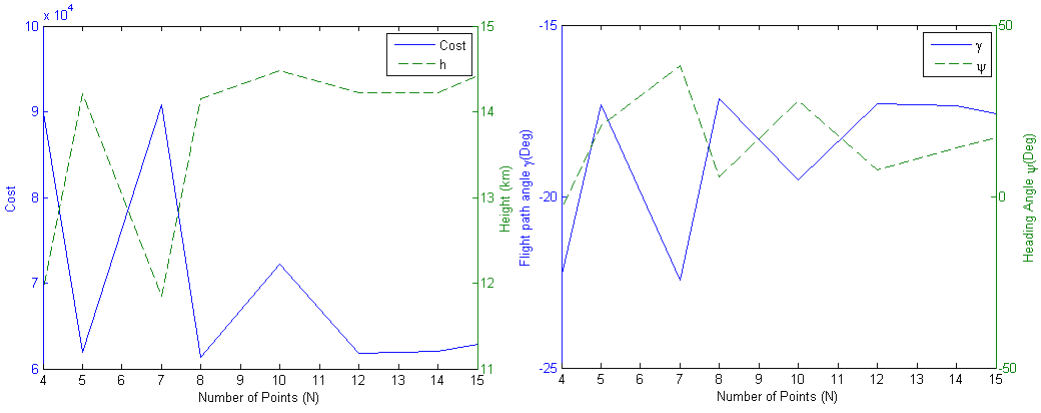

Figure 17: Variation of states and cost with N in basic formulation with adaptive time

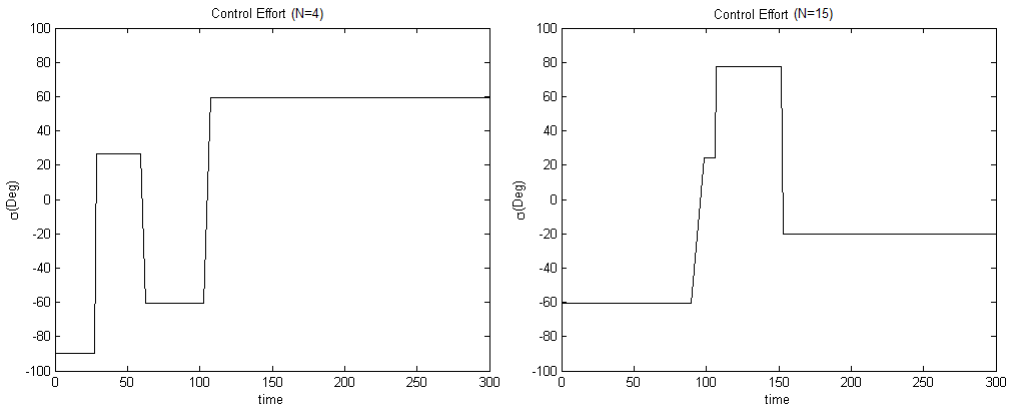

Figure 18: Optimal bank angle profiles in basic formulation with adaptive time

It can be seen in Figure 18 that the transition of control from one value to another does not occur at equal time intervals. The spacing between these intervals is determined by the value of control W at the corresponding values of τ.

3.2 Alternative formulation with Adaptive Time

This formulation includes the extra states introduced in 2.2 and also time as an additional state.

3.2.1 System Dynamics

The modified state vector is

$$x = \begin{bmatrix} h & \theta & \phi & V & \gamma & \psi & z_1 & z_2 & t \end{bmatrix}^T \tag{40}$$

where z_1 and z_2 are given by (15)

and the modified control is

$$u = \begin{bmatrix} \dot{\sigma} \\ W \end{bmatrix} \tag{41}$$

Since the control profile generated by the algorithm is piecewise constant, the bank angle profile will be piecewise linear.

Equations of motion

The equations of motion consist of the set of equations (37), the equations

$$\left.\begin{aligned} \dot{z}_1 &= W\dot{\sigma}cos\sigma \\ \dot{z}_2 &= -W\dot{\sigma}sin\sigma \end{aligned}\right\} \tag{42}$$

obtained by differentiating z_1 and z_2 with respect to independent variable τ, and (20).

3.2.2 Constraints

The constraints consist of

(i) Final State Constraints

Set of equations (3),(5) & (38).

(ii) Path Constraints

Set of equations (16)

(iii) Control Constraints

Set of equation (17) & (29).

(iv) Initial Constraints

Set of equations (7),(39) and leaving the initial values of z_1 and z_2 free for the algorithm to optimize in order to arrive at the solution.

3.2.3 Cost Function

(i) Modified Mayer Cost (9)

$$M(t_f, x_f) = -k_h h(t_f) + k_\gamma \gamma(t_f)^2 + 0.5k_1(\theta(t_f) + 1.2576)^2 + 0.5k_2(\phi(t_f) + 0.7203)^2$$

with $k_h = 5km^{-1}$, $k_g = 91.4(180/pi)^2 rad^{-2}$ and $k_1 = k_2 = 1000$.

(ii) Modified Lagrange Cost (18) & (34)

$$L = W(exp(b_1(z2 - cmin)) + exp(c_1(\dot{\sigma} - umin)) + exp(-c_1(\dot{\sigma} - umax)))$$

with $b_1 = 120$, $c_1 = 10$, $cmin = cos(18.2pi/180)$, $umin = -pi/9$ and $umax = pi/9$.

3.2.4 Solution

In this case too, the spacing between time intervals is optimized by the algorithm. The initial control sequence used here is an array of length N with each element equal to zero. The initial value of ρ is chosen to be 1000. Figure 19 shows the variation of all the states along the trajectory.

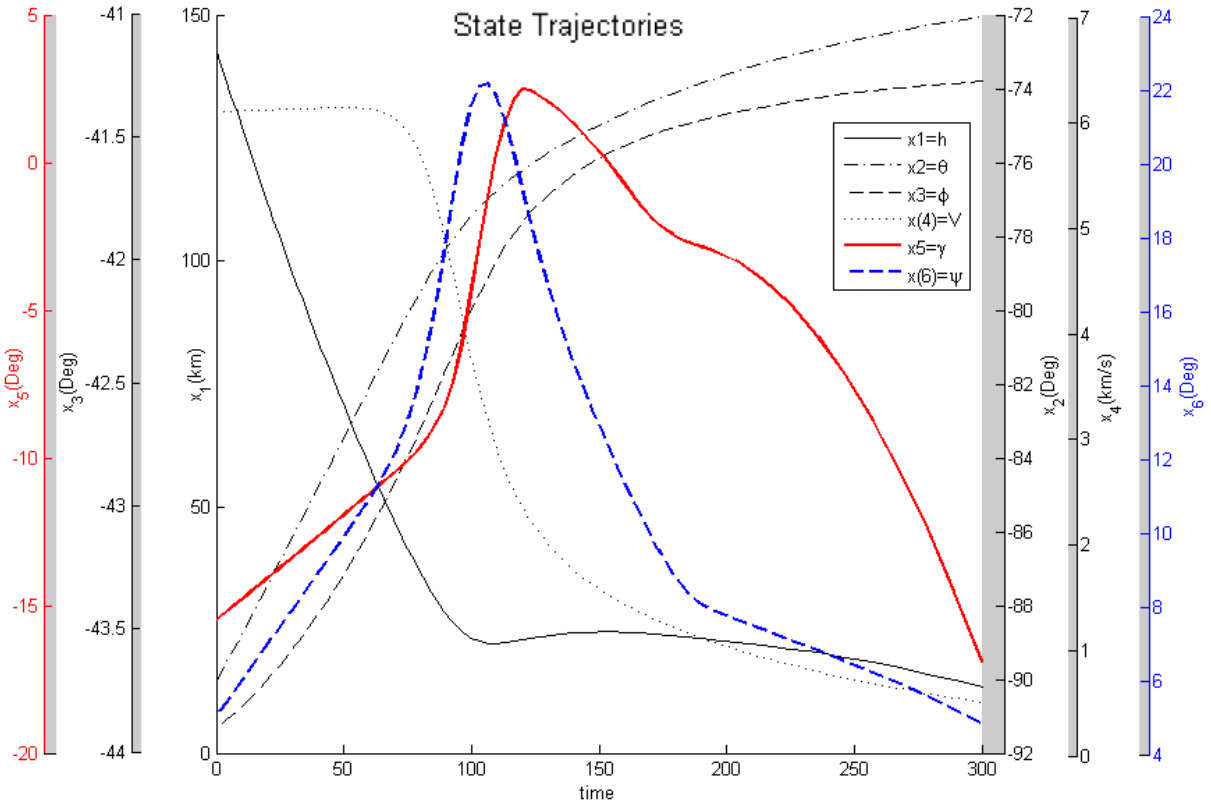

Figure 19: Variation of state variables along the trajectory in alternative formulation with adaptive time

Table 5: Final values of states in alternative formulation with adaptive time

State Variable	h(km)	$\theta°$	$\phi°$	$V(km/s)$	$\gamma°$	$\psi°$
Initial Value	143	-90.07	-43.90	6.082	-15.50	4.99
Final Value ($N = 10$)	13.3152	-72.0552	-41.2702	0.48	-16.8902	4.7859

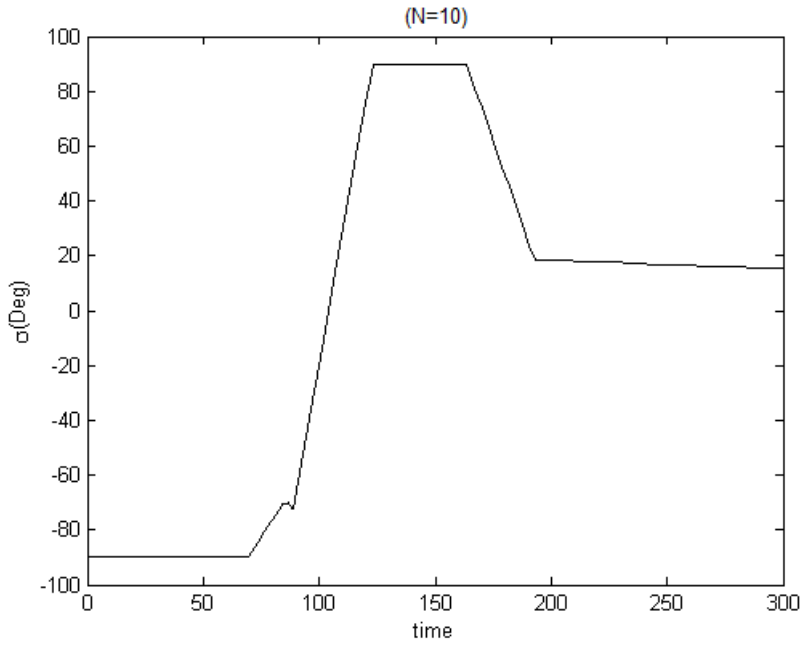

Figure 20: Optimal bank angle profile in alternative formulation with adaptive time

For this formulation, the algorithm is able to find an optimal solution for N equal to 10 only. If we compare the performance of this approach with the alternative formulation, it is evident from Table 5 that for same number of intervals, this formulation is able to achieve better control authority, i.e. lower magnitude of flight path angle (16.89) at a higher landing altitude (13.3152km). In this formulation, the spacing between the time intervals at which the transition of control takes place is decided by W (see Figure 20). Thus we are able to obtain an optimal control profile by using fewer time intervals.

4 CONCLUSIONS

A sequential linear quadratic approach with adaptive time discretization has been developed to solve nonlinear optimal control problems. With this approach it is shown that an optimal solution can be achieved by considering fewer time intervals. In the application of this algorithm, an optimal bank angle profile for higher elevation Mars landing is developed. The number of intervals required to obtain bank angle profile in the formulation without adaptive time approach is shown to be much higher than in formulations that use this approach. The computational time required to attain optimal solution is also greatly reduced. We compare the results obtained here and in [2] in Table 6.

Table 6: Comparison between final altitude h_f and final flight path angle γ_f for different solutions

Performance comparison		$h_f(km)$	$\gamma_f(deg)$
Problem Formulations	basic formulation (N=50)	14.22	-17.26
	alternative formulation (N=50)	13.19	-14.74
	basic Formulation (Adaptive time N=15)	14.4	-17.58
	alternative formulation (Adaptive time N=10)	13.4	-16.89
Profiles considered in [2]	Bank-limited optimal	13.2	-10.1
	Planner	13.3	-9.6

REFERENCES

[1] Rodriguez, Luis A., and Sideris, A., "A sequential linear quadratic approach for constrained nonlinear optimal control," *American Control Conference (ACC)*, 2011, pp. 1470-1475.

[2] Benito, J., Soler, L., Bombelli, A., and Mease, K. D., "Entry Trajectory Planner for Higher Elevation Mars Landing", preprint, 2014.

[3] Vinh, N. X., "Optimal Trajectories In Atmospheric Flight", *Elsevier Scientific Publishing Co.*, Amsterdam, 1981.

[4] Braun, R. D. and Manning, R. M., "Mars Exploration Entry, Descent and Landing Challenges," *J. Spacecraft and Rockets*, Vol. 44, 2007, pp. 310-323

[5] Cavallo, A. and Ferrara, F., "Atmosphere Re-Entry Control For Low Lift/Drag Vehicles," *J. Guidance, Control and Dynamics*, Vol. 19, No. 1, 1996, pp. 47-53.

[6] Rao, A. V., Benson, D. A., et al. "Algorithm 902: GPOPS, a Matlab Software for Solving Multiple-Phase Optimal Control Problems Using the Gauss Pseudospectral Method," *ACM Transactions on Mathematical Software (TOMS)*, Vol. 37, No. 2, 2010, pp. 1-39.

[7] Patterson, M. P. and Rao, A. V., "GPOPS-II, A Matlab Software for Solving Multiple-Phase Optimal Control Problems Using hp-adaptive Gaussian Quadrature Collocation Methods and Sparse Nonlinear Programming," *ACM Transactions on Mathematical Software (TOMS)*, Vol. 41, No. 1, 2014, pp. 1-37.

[8] Darby, C. L., Hager, W. W., and Rao, A.V., "hp-adaptive Pseudospectral Method for Solving Multiple-Phase Optimal Control Problems," *Optimal Control Applications and Methods*, Vol. 32, No. 4, 2011, pp. 476-502.

APPENDIX

Code for obtaining the equations of motion

```
%w= Mars Angular Velocity, rp= Mars radius, g= Acceleration gravity,
%r=density, Surface Area S, mass m, Lift and drag coefficients are
%considered constant;
rp=3397;
mu=42409
w=7.095e-5;
CL=0.62;
CD=1.92;
m=2804;
S=15.9e-6; %  km^2
% change r <--> h
%x(1)=x(1)+rp; % now x(1)=h --> r
r=x(1)+rp;
h=x(1);
g=mu/r^2;
ro=0.013e9*exp(-9.2e-2*h-1.94e-5*h^2-7.51e-6*h^3+4.2e-8*h^4);   % (h in km)
    rho in kg/km^3
dro=ro*(-9.2e-2-2*1.94e-5*h-3*7.51e-6*h^2+4*4.2e-8*h^3);%differentiation of
    ro wrt x(1) % (h in km) drho in kg/km^4
L=0.5*x(4)^2*ro*(S/m)*CL;    % (km/sec^2)
D=0.5*x(4)^2*ro*(S/m)*CD;    % (km/sec^2)
% r=x(1), theta=x(2); phi=x(3); v=x(4); gamma=x(5); psi=x(6)
xdot=[x(4)*sin(x(5));...
x(4)*cos(x(5))*cos(x(6))/(r*cos(x(3)));...
x(4)*cos(x(5))*sin(x(6))/r;...
-D-g*sin(x(5));...
(L*cos(u)-(g-x(4)^2/r)*cos(x(5)))/x(4)+2*w*cos(x(6))*cos(x(3));...
-L*sin(u)/(x(4)*cos(x(5)))-x(4)*cos(x(5))*cos(x(6))*tan(x(3))/r+...
2*w*(tan(x(5))*sin(x(6))*cos(x(3))-sin(x(3)))];
```

```
r_v=sin(x(5));
r_gam=x(4)*cos(x(5));
th_r=-x(4)*cos(x(5))*cos(x(6))/(r^2*cos(x(3)));
th_phi=x(4)*cos(x(5))*cos(x(6))*sec(x(3))*tan(x(3))/r;
th_v=cos(x(5))*cos(x(6))/(r*cos(x(3)));
th_gam=-x(4)*sin(x(5))*cos(x(6))/(r*cos(x(3)));
th_psi=-x(4)*cos(x(5))*sin(x(6))/(r*cos(x(3)));
phi_r=-x(4)*cos(x(5))*sin(x(6))/r^2;
phi_v=cos(x(5))*sin(x(6))/r;
phi_gam=-x(4)*sin(x(5))*sin(x(6))/r;
phi_psi=x(4)*cos(x(5))*cos(x(6))/r;
v_r=-0.5*x(4)^2*(S/m)*CD*dro+2*mu*sin(x(5))/r^3;
v_v=-x(4)*ro*(S/m)*CD;
v_gam=-g*cos(x(5));
gam_r=0.5*x(4)*(S/m)*CL*cos(u)*dro+2*mu*cos(x(5))/(x(4)*r^3)-x(4)*cos(x(5))
    /(r^2);
gam_phi=-2*w*cos(x(6))*sin(x(3));
gam_v=0.5*ro*(S/m)*CL*cos(u)+g*cos(x(5))/x(4)^2+cos(x(5))/r;
gam_gam=(g-x(4)^2/r)*sin(x(5))/x(4);
gam_psi=-2*w*sin(x(6))*cos(x(3));
psi_r=-0.5*x(4)*(S/m)*CL/cos(x(5))*sin(u)*dro+x(4)/r^2*cos(x(5))*cos(x(6))*
    tan(x(3));
psi_phi=-x(4)/r*cos(x(5))*cos(x(6))*(sec(x(3)))^2+2*w*(-tan(x(5))*sin(x(6))
    *sin(x(3))-cos(x(3)));
psi_v=-0.5*ro*(S/m)*CL*sin(u)/cos(x(5))-cos(x(5))*cos(x(6))*tan(x(3))/r;
psi_gam=-L*sin(u)*sec(x(5))*tan(x(5))/x(4)+x(4)/r*sin(x(5))*cos(x(6))*tan(x
    (3))+2*w*((sec(x(5)))^2*sin(x(6))*cos(x(3)));
psi_psi=(x(4)/r*cos(x(5))*sin(x(6))*tan(x(3)))+2*w*(tan(x(5))*cos(x(6))*cos
    (x(3)));
%Jacobian with respect to x
fx=[0 0 0 r_v  r_gam 0;...
th_r 0 th_phi th_v th_gam th_psi;...
phi_r 0 0 phi_v phi_gam phi_psi;...
v_r 0 0 v_v v_gam 0;...
```

```
gam_r 0 gam_phi gam_v gam_gam gam_psi;...
psi_r 0 psi_phi psi_v psi_gam psi_psi;...
];
%Jacobian with respect to u
fu = [0;0;0;0;-L*sin(u)/x(4);-L*cos(u)/(x(4)*cos(x(5)))];
```