The Pennsylvania State University
The Graduate School

OPTIMAL FEEDBACK CONTROL DESIGN FOR A HYPERSONIC REENTRY VEHICLE

A Thesis in
Aerospace Engineering
by
Mehrdad Mirzaei

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

August 2019

The thesis of Mehrdad Mirzaei was reviewed and approved* by the following:

Puneet Singla
Associate Professor of Aerospace Engineering
Thesis Advisor

Robert G. Melton
Professor of Aerospace Engineering

Amy R. Pritchett
Professor of Aerospace Engineering
Head of the Department of Aerospace Engineering

*Signatures are on file in the Graduate School.

Abstract

The study of hypersonic flight is of utmost importance for commercial as well as military missions involving orbital and near orbital speeds. Hypersonic trajectory optimization is considered a challenging problem due to its highly nonlinear dynamic equations of motion, and different constraints which must be satisfied during the mission. Although open-loop optimal trajectory planning and feedback control laws for hypersonic flight have been studied in detail separately, to our best knowledge, very little attention has been devoted to the design of optimal feedback control laws. Designing the optimal feedback control law is a difficult task because it requires the solution of a nonlinear partial differential equation (PDE) called the Hamilton Jacobi Bellman (HJB) equation which is a multivariate function of system states and time. The main challenge in providing a numerical solution for the governing PDE is the *curse of dimensionality*. In this thesis, the recently developed sparse collocation method is used for deriving optimal feedback control laws for the hypersonic flight trajectory guidance problem. The sparse collocation method exploits a non-product quadrature rule known as the *Conjugate Unscented Transformation* (CUT) in conjunction with l_1-norm optimization to derive the optimal feedback control laws. The CUT method provides the minimal number of initial conditions for open-loop solutions corresponding to domain of interest. The l_1-norm based sparse approximation provides an interpolating surface for value function through these CUT provided open-loop solutions. The main advantage of this approach is that it does not require any a-priori information or assumption about the optimal control profile. A test hypersonic re-entry case corresponding to maximum energy impact at a known target location is considered to show the efficacy of the developed methodology. Finally, the performance of interpolating surface representing the optimal feedback control law is demonstrated for random initial conditions.

Table of Contents

List of Figures

List of Tables

Acknowledgments

Nobody has been more important to me in the pursuit of this thesis than the members of my family. Most importantly, I wish to thank my loving and supportive wife, Monir, who provides unending inspiration. I would like to thank my parents, whose love and guidance are with me in whatever I pursue.

I would also like to thank my thesis advisor Dr. Puneet Singla. He is the most knowledgeable person in his research area and he taught me a lot in this area, but the most important thing he taught me was how to work independently.

I would also like to acknowledge Dr. Robert G. Melton and Dr. Amy Pritchett as the second readers of this thesis, and I am gratefully indebted to them for their very valuable comments on this thesis.

This material is based upon work supported jointly by the National Science Foundation (NSF) under award #1826990 and Air Force Office for Scientific Research (AFOSR) grant $\#FA9550-17-1-0088$. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF and AFOSR.

Chapter 1
Introduction

The study of hypersonic flight is of utmost importance for commercial as well as military missions involving orbital and near-orbital speeds. A hypersonic vehicle (HV) generally refers to an aero vehicle which is capable of exceeding speeds of Mach 5. Unique features of hypersonic vehicles, such as their high speed and maneuverability, have made them very popular to use for space transportation purposes. Also, the notable ability of hypersonic vehicles in re-entering the atmosphere to quicly attack targets on the ground has made the quick global strike possible.

The control of hypersonic flight vehicles poses unique challenges due to the nonlinear and coupled nature of the vehicle dynamics. In addition, high Mach numbers and the sensitivity of the flight trajectory to model parameters and change in atmospheric conditions add more difficulty in controlling the the hypersonic vehicles. All of the above challenges are even more highlighted at lower altitudes such as during re-entry. A flight can be further characterized by its mission requirements. In some missions, high deceleration in the outer atmosphere is desired during reentry flight causing the vehicle to experience a high angle of attack for a long period of time. On the other hand, some missions require deceleration near Earth's surface, resulting in a high angle of attack for shorter times [2]. Regardless, stable reentry flight guidance is required.

The reentry reference trajectory generation and reentry guidance problem can be posed as an optimal control problem whose solution is classified in two parts: open-loop solution and a closed-loop solution (feedback control). Both open-loop optimal trajectory planning and stable feedback control design pertaining to different hypersonic flight missions have garnered much attention from the control community [3–14].

Generally, the open-loop reference trajectory is constructed offline. If the mission is changed during the flight, the original trajectory is discarded and a new optimal trajectory has to be determined. Open-loop trajectory optimization is not robust with respect to state disturbances or uncertainties and hence, any small deviation from the reference trajectory requires a re-planning of the optimal reentry trajectory. Separately, the optimal feedback guidance calculates the required control inputs from the knowledge of current state vector to follow the desired trajectory. Although optimal open-loop solutions [3, 4, 6, 7, 9, 11–13] and optimal closed-loop feedback solutions [5, 8, 10, 14] for hypersonic flight have been studied in detail separately, very little attention has been devoted to the design of stable optimal feedback solutions. Therefore, the goal of this thesis is to develop optimal feedback guidance for hypersonic reentry flight.

While the design of an optimal feedback (i.e., closed-loop) controller is desirable due to its insensitivity to state perturbations and external disturbances and guaranteed performance, the trajectory needs to be derived from Bellman's principle of optimality [15]. Hence, it requires the solution of the Hamilton Jacobi Bellman (HJB) equation, a nonlinear partial differential equation [16, 17] for which there currently does not exist any unique analytical method of solution in the case of an arbitrary cost function and terminal constraints. However, researchers have worked on different numerical approaches to find local solutions to this important problem, which will be discussed in the next chapter [18–31]. A major disadvantage of these numerical approximation methods is that they must assume a fixed structure for the controller before solving for the feedback control law. Also, most of the aforementioned techniques suffer greatly from the *curse of dimensionality*, increasing the computational burden associated with their implementation. This makes these approaches impractical for real-time applications such as hypersonic flight trajectory guidance.

This thesis applies the recently formulated sparse collocation method [32–37] to derive an optimal feedback guidance controller for hypersonic reentry problem. Sparse collocation makes use of a non-product quadrature method known as the *Conjugate Unscented Transformation* (CUT) [38] to develop an accurate and computationally efficient approach to design the optimal feedback control policy for a nonlinear dynamical system. This approach mitigates the effects of the curse of dimensionality, and can be extended to problems subject to terminal constraints.

Building upon those promising prior results [36, 37], this thesis solves the free final-time hypersonic flight problem to maximize the impact energy of a glide weapon at a target.

A brief summary of the chapters and overall organization of this thesis is detailed in the following. Chapter 2 presents an overview of optimal control theory and trajectory optimization. This chapter begins with the introduction of general form of the optimal control problem, and different type of solutions. Next, the open-loop solution is presented, the trajectory optimization is introduced as an application of it, and a summary of different methods in trajectory optimization is provided. The last section reviews the optimal feedback solution of the optimal control problem and studies the challenges in solving the HJB equation and obtaining the closed-loop solution.

Chapter 3 presents the trajectory optimization problem for a hypersonic reentry vehicle in a glide turn mission and derives the necessary condition of optimality for this problem and provides the numerical results of this problem.

Chapter 4 presents the process of deriving the optimal feedback control law for the same problem as defined in the chapter 3. This chapter begins with an introduction to the sparse collocation method, which is the basis of the proposed method in this thesis to design the optimal feedback solution. Next, modifications to sparse collocation method are described to generate optimal feedback solutions for problems with terminal constraints and free final time. In the last section, the results are presented and the performance of the designed optimal feedback control is compared with an open-loop solution.

Chapter 5 summarizes the thesis, and provides the conclusion of this work and suggestions for future research based on this thesis.

Chapter 2
A Review of Optimal Control Theory and Trajectory Optimization

This chapter presents a short review of theorems and methods in optimal control theory and trajectory optimization problem. The process of designing a trajectory while minimizing a performance index and simultaneously satisfying a set of constraints is called trajectory optimization. The trajectory optimization problem is an optimal control problem (OCP) and generally is solved by finding an open-loop solution to the associated OCP. The term trajectory optimization refers to a set of methods that are used to find the best choice of trajectory [39]. In general, the trajectory optimization problem for many applications in aerospace (e.g.,hypersonic reentry vehicle, spacecraft attitude maneuver) is a difficult task because the path must be feasible to fly and simultaneously satisfy the constraints on initial conditions, terminal conditions, and trajectory constraints [40].

The trajectory optimization problem can be formulated in different ways [41–43], but the focus of this thesis is restricted to single-phase continuous-time trajectory optimization problems, i.e. those where the system dynamics are continuous throughout the entire trajectory.

This chapter, first, presents a brief overview of optimal control, followed by a discussion on solution types for this problem. Next, different methods in finding the open-loop solution and trajectory optimization problem are discussed. Finally, the derivation of optimal feedback control is studied.

2.1 Optimal Open-Loop Solution

A general optimal control problem (OCP), for trajectory optimization can be formulated as:

$$\min_{\mathbf{u}(t)} J(\mathbf{x}(t), \mathbf{u}(t), t) = S(\mathbf{x}(t_f), t_f) + \int_{t_0}^{t_f} L(\mathbf{x}(t), \mathbf{u}(t), t)dt \tag{2.1}$$

$$\text{Subject to :} \quad \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t),\ t_0\ given \tag{2.2}$$

$$\mathbf{\Psi}(\mathbf{x}(t_f), t_f) = 0 \tag{2.3}$$

where $\mathbf{x}(t) \in \mathbf{R}^n$ is the state vector, and $\mathbf{u}(t) \in \mathbf{R}^m$ is the control vector, and t is the independent time variable. Equation (2.1) shows the cost functional, which should be minimized, and equation (2.2) is the system of ODEs representing system dynamic constraints on state variable. Equation (2.3) is the terminal state manifold constraint. The terminal constraint implies the constraint on value of all, or a subset of, the state variables at the final time. The final time t_f is assumed to be free and needs to be determined. One can use the theory of Lagrange multipliers to adjoint the terminal constraint to the cost functional:

$$J = S(\mathbf{x}(t_f), t_f) + \boldsymbol{\nu}^T \mathbf{\Psi}(\mathbf{x}(t_f), t_f) + \int_{t_0}^{t_f} L(\mathbf{x}(t), \mathbf{u}(t), t)dt \tag{2.4}$$

where $\boldsymbol{\nu}$ is the vector of Lagrange multipliers corresponding to the terminal constraint. The Hamiltonian function, H is introduced to accommodate dynamic constraints:

$$H(\mathbf{x}(t), \mathbf{u}(t), \boldsymbol{\lambda}(t), t) = L(\mathbf{x}(t), \mathbf{u}(t), t) + \boldsymbol{\lambda}^T(t)\mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) \tag{2.5}$$

where $\boldsymbol{\lambda}(t) \in \mathbf{R}^n$ is the Lagrange multiplier vector corresponding to the dynamic constraint (in optimal control theory, it is also known as the costate vector). Using the calculus of variations the first-order necessary conditions of optimality are given as [17]:

$$\dot{\boldsymbol{\lambda}}(t) = -\frac{\partial H}{\partial \mathbf{x}} \tag{2.6}$$

$$\dot{\mathbf{x}}(t) = -\frac{\partial H}{\partial \mathbf{x}} = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) \tag{2.7}$$

$$\frac{\partial H}{\partial \mathbf{u}} = 0 \tag{2.8}$$

$$\boldsymbol{\lambda}^T(t_0)\delta\mathbf{x}(t_0) + \Big[\frac{\partial \overline{S}}{\partial \mathbf{x}_f} - \boldsymbol{\lambda}(t)\Big]_{t_f}^{T} \delta\mathbf{x}(t_f) + \Big[H + \frac{\partial S}{\partial t_f}\Big]_{t_f} \delta t_f = 0 \tag{2.9}$$

Equations (2.6)-(2.8) are the first-order necessary conditions for the optimality of the control law, and equation (2.9) is the transversality condition, which indicates the boundary conditions on the states and costates. $\delta\mathbf{x}(t_0)$, and $\delta\mathbf{x}(t_f)$ represent the variation of states at initial time and final time, respectively. δt_f is the variation of the final time. If any of these variables in the given problem is fixed, then the corresponding variation has to be zero. Hence, the transversality equation has different structure for different problems (free final time, fixed final state, etc.). In the transversality equation $\overline{S}$ is the augmented terminal cost and is defined as:

$$\overline{S}(\mathbf{x}(t_f), t_f) = S(\mathbf{x}(t_f), t_f) + \boldsymbol{\nu}^T \boldsymbol{\Psi}(\mathbf{x}(t_f), t_f) \tag{2.10}$$

Equation (2.8) is the key equation in the solution process. This equation comes from Pontryagin's Minimum Principle (PMP), which states that the optimal control must minimize the Hamiltonian. Equation (2.8) is used to eliminate the control

variable from equations (2.6), and (2.7). Therefore, the OCP is changed to a two-point boundary value problem (TPBVP) with differential equations (2.6), and (2.7), and the boundary condition comes from the transversality equation (2.9), which should be modified for a problem of interest.

Also it should be added that the open-loop solution satisfies only the necessary condition for optimality and this condition can not guarantee the global optimality of the solution.

2.2 Numerical Methods

Generally the methods to solve the OCP associated with a trajectory optimization problem are divided into two categories: *direct* and *indirect* methods. Each of these methods has its own advantages and disadvantages, which are detailed in the following. A major difference between these two methods is that a direct method discretizes and then optimizes, while an indirect method optimizes and then discretizes. The most known direct method is the Pseudospectral method which is described in a separate section because of its importance.

2.2.1 Indirect Methods

Indirect methods analytically construct the necessary conditions for optimality and solves the resulting TPBVP through a discretization process. Indirect methods are applied to find an extremum of the performance index [44, 45], which requires solving a multi-point boundary value problem (BVP). This BVP is constructed from the first order necessary condition of optimality which has been provided in equations (2.6)-(2.9).

Implementing the indirect methods and deriving the necessary condition of optimality can be challenging, and it has to be repeated for each trajectory optimization problem. The corresponding multi-point BVP of indirect method needs a good initial guess in both the state and costate to converge to a solution, but the costate doesn't have a physical meaning in many applications and it's difficult to assign a good initial guess to it.

Indirect methods give a highly accurate solution and also they are known for the fast numerical convergence in the neighborhood of the optimal solution. These

features make them attractive in the aerospace industry.

2.2.2 Direct Methods

Instead of deriving the necessary conditions to find the extremum of a performance index, direct methods discretize the trajectory optimization problem and convert it to a nonlinear programming problem (NLP). This can be done by different methods such as direct shooting method [46], control and state parameterazation methods [47,48], and pseudospectral methods [49–51]. In the direct shooting method the control variable is parameterized and is integrated with explicit numerical approaches to satisfy the equation (2.2). In the second approach one can avoid computationally expensive numerical integration by approximating both the state and control with a polynomial spline. This is due to the fact that parameterized state and control can be substituted in equation (2.2) to satisfy this equation only at the collocation points. Pseudospectral methods have become one of the popular direct methods in solving the trajectory optimization problems in recent years [52], the same as other direct methods, Pseudospectral methods convert the OCP problem to an NLP problem, but the difference in pseudospectral methods is that the nodes (collocation points) are not uniform and are clustered near boundaries. The pseudospectral methods are identified with the type of quadrature points they use, such as Legendre-Gauss points (LG) [53], Legendre-Gauss-Lobatto points (LGL) [54], Legendre-Gauss-Radau points (LGR) [55], and Chebyshev-Gauss-Lobatto points (CGL) [56].

The nonlinear programming problems resulting from direct methods often have a very sparse structure, and NLP solvers, such as SNOPT [57], are used to efficiently solve this problem.

The advantage of the direct methods is in the avoidance of using adjoint variables (costates) or changing the structure of the problem, and the disadvantage of direct methods is the low accuracy of the solution in comparison with indirect methods. Table 2.1 shows the advantages and disadvantages of both direct and indirect methods. As a result, designers usually use direct methods instead of indirect methods in conceptual design.

Table 2.1: Comparison between direct and indirect methods [1]

	Advantages	Disadvantages
Direct Methods	Large region of attraction Widespread NLP solvers exist	Computationally intensive Optimality not guaranteed
Indirect Methods	Rapid convergence Necessary conditions satisfied	Small region of attraction Costates introduced

2.2.3 Pseudospectral Method

In this section, the basic principle of the Gauss Pseudospectral Method (GPM) is presented based on references [49, 51, 52, 58]. This method is the basis for the software package known as DIDO used in this thesis. Consider the OCP (2.1) - (2.3); these equations present the continuous OCP in the Bolza form. To use the pseudospectral method, first the state and control variables must be discretized at the collocation points which are Legendre-Gauss nodes, and then approximated with the Lagrange interpolation polynomials. To avoid the explosion in the Lagrange interpolation polynomials, the time variable is normalized in the general interval $\tau \in [-1,\ 1]$ by the following transformation:

$$\tau = \frac{2t}{t_f - t_0} - \frac{t_f + t_0}{t_f - t_0} \tag{2.11}$$

where τ is the normalized time variable and $\tau_0 = -1,\ \tau_f = 1$, therefore the form of the OCP (2.1) - (2.3) will be changed to:

$$J = S(\mathbf{x}(1), t_f) + \frac{t_f - t_0}{2} \int_{-1}^{1} L(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau) d\tau \tag{2.12}$$

$$\frac{d\mathbf{x}}{d\tau} = \frac{t_f - t_0}{2} \mathbf{f}(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau) \tag{2.13}$$

$$\mathbf{\Psi}(\mathbf{x}(1), t_f) = 0 \tag{2.14}$$

Consider $N+2$ discretized nodes including N Legendre-Gauss collocation points τ_k $(k = 1,\ 2,\ ...,\ N)$ and two boundary nodes τ_0, and τ_f, therefore the state trajectory $\mathbf{x}(\tau)$ can be approximated with a $N+1$ order Lagrange interpolation polynomials as:

$$\begin{aligned} \mathbf{x}(\tau) \approx \mathbf{X}(\tau) &= \sum_{i=0}^{N} \mathcal{L}_i(\tau)\mathbf{X}(\tau_i) \\ \mathcal{L}_i(\tau) &= \prod_{j=0, j\neq i}^{N} \frac{\tau - \tau_j}{\tau_i - \tau_j} \qquad (i = 0,\ 1,\ ...,\ N) \end{aligned} \tag{2.15}$$

Where $\mathcal{L}(\tau)$ is the Lagrange interpolation polynomial. Similarly, the control trajectory $\mathbf{u}(\tau)$ can be approximated using a basis of N Lagrange interpolation polynomials, $\mathcal{L}^*$:

$$\begin{aligned} \mathbf{u}(\tau) \approx \mathbf{U}(\tau) &= \sum_{i=1}^{N} \mathcal{L}_i^*(\tau)\mathbf{U}(\tau_i) \\ \mathcal{L}_i^*(\tau) &= \prod_{j=1, j\neq i}^{N} \frac{\tau - \tau_j}{\tau_i - \tau_j} \qquad (i = 1,\ 2,\ ...,\ N) \end{aligned} \tag{2.16}$$

From the equations (2.15) - (2.16) it can be stated that Lagrange interpolation polynomials used to approximate state and control satisfy the following properties:

$$\begin{aligned} \mathcal{L}_i(\tau_j) &= \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases} \\ \mathcal{L}_i^*(\tau_j) &= \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases} \end{aligned} \tag{2.17}$$

The derivative of state can be produced by differentiating of equation (2.15):

$$\dot{\mathbf{x}}(\tau) \approx \dot{\mathbf{X}}(\tau) = \sum_{i=0}^{N} \mathbf{X}(\tau_i)\dot{\mathcal{L}}_i(\tau) \tag{2.18}$$

To describe the derivative of Lagrange polynomials at each Legendre-Gauss node, a differential approximation matrix $D_{N\times N+1}$ is introduced as:

$$D_{ki} = \dot{\mathcal{L}}_i(\tau_k) = \sum_{l=0}^{N} \frac{\prod_{j=0,j\neq i,l}^{N}(\tau_k - \tau_j)}{\prod_{j=0,j\neq i}^{N}(\tau_i - \tau_j)} \tag{2.19}$$

where $k = 1, 2, ..., N$, and $i = 0, 1, ..., N$. The elements of the matrix D can be calculated offline. By using the differentiation matrix the dynamic constraint (2.13) can be converted into the algebraic constraint:

$$\sum_{i=0}^{N} D_{ki}\mathbf{X}(\tau_i) - \frac{t_f - t_0}{2}\mathbf{f}(\mathbf{X}(\tau_k), \mathbf{U}(\tau_k), \tau_k) = 0 \qquad (k = 1,\ 2,\ ...,\ N) \tag{2.20}$$

It should be noted that the dynamic constraint is collocated only at the collocation nodes not at the boundary nodes (τ_0, and τ_f). The state approximation (2.15) does not include the state at the final time, therefore an additional constraint for the final state can be constructed by using the Gauss quadrature method [59] as:

$$\mathbf{X}(\tau_f) - \mathbf{X}(\tau_0) = \frac{t_f - t_0}{2}\sum_{k=1}^{N} \omega_k \mathbf{f}(\mathbf{X}(\tau_k), \mathbf{U}(\tau_k), \tau_k) \tag{2.21}$$

where ω_k are the Gauss weights. Using the Gauss quadrature method [59], the cost function (2.12) is approximated as:

$$J = S(\mathbf{X}(1), t_f) + \frac{t_f - t_0}{2}\sum_{k=1}^{N} \omega_k L(\mathbf{X}(\tau_k), \mathbf{U}(\tau_k), \tau_k) \tag{2.22}$$

Finally, the OCP (2.1) - (2.3) can be transcribed to the NLP with the cost function (2.22), and dynamic algebraic constraint (2.21), and boundary con-

straints (2.14), and (2.21). The solution of this NLP is a approximation of the continues OCP solution.

2.3 Optimal Feedback Control Solution

While the trajectory optimization provides the way point, the feedback control solutions are required to achieve the optimal path in real-time due to uncertainties in system state, unknown dynamics and external disturbances. The open-loop solution suffers from the lack of robustness with respect to the initial condition uncertainty and process noise, therefore, a new OCP must be solved for any expected case. Optimal feedback law solution can overcome this shortcoming by using the current state value to generate the optimal control input, therefore any disturbances in the system will be considered and cancelled during the feedback process.

Bellman's principle states that, an optimal policy has the property that no matter what the previous decision (i.e., controls) have been, the remaining decisions must constitute an optimal policy with regard to the state resulting from those previous decisions. Bellman's principle plays a similar role to system control as Pontryagin's minimum principle in the calculus of variations approach. It limits the number of potentially optimal control decisions that must be searched. One of the important results of the optimality principle is that the optimal control trajectory must be obtained backward from the final state. From the principle of optimality it can be stated that the OCP is naturally a backward in time problem and to find the optimal strategy it should be investigated backward from the final state.

Bellman's principle is the basis of a mathematical optimization method known as Dynamic Programming (DP). The method was developed by Richard Bellman in the 1950s and refers to simplifying a complicated problem by breaking it down into simpler sub-problems in a backward direction. Applying DP on a general OCP leads to a partial differential equation (PDE), which is the governing PDE of the optimal control value function, and known as Hamilton Jacobi Bellman (HJB) equation:

$$\frac{\partial V(\mathbf{x}(t),t)}{\partial t} + \min_{\mathbf{u}(t)} \left[H(\mathbf{x}(t), \frac{\partial V}{\partial \mathbf{x}}, \mathbf{u}(t), t) \right] = 0 \tag{2.23}$$

The HJB equation provides both necessary and sufficient conditions for optimal-

ity of the value function. The term $\frac{\partial V}{\partial \mathbf{x}}$ is the costate, which is the sensitivity of the value function with respect to the state. From Pontryagin's minimum principle:

$$\min_{\mathbf{u}(t)} \left[H(\mathbf{x}(t), \frac{\partial V}{\partial \mathbf{x}}, \mathbf{u}(t), t) \right] = H(\mathbf{x}(t), \frac{\partial V^*}{\partial \mathbf{x}}, \mathbf{u}^*(t), t) \tag{2.24}$$

Note that the terminal cost appears only in the boundary condition and not in the HJB equation itself. In fact, the specifics about the terminal cost and terminal time did not play a role in the HJB equation. For different problems, the boundary condition changes (as it is already discussed) but the HJB equation remains the same.

The HJB equation, together with a boundary condition that specifies the terminal cost, sufficiently characterizes the optimal solution to the OCP. Since it is expressed over the whole state space, solutions to the HJB equation yield optimal feedback law. Unfortunately, the HJB equation cannot be solved analytically in most settings. Therefore, numerical techniques, such as the value iteration method [27, 60] and Dynamic Programming [16], must be employed.As a classic solution in this field, Bryson presented the LQR problem and offered a method [17] to determine the linear feedback solution for a problem which has a quadratic cost function. Subsequently, different techniques have been offered to find the closed-loop solution of problems where the cost function and system dynamics are smooth functions [18–22, 25]. The state-Riccati equation method absorbs the nonlinear functions of the state inside the control function [18]. Alternative methods like $\theta - D$ approach [20] and its variants [19] use quasi-linearization technique to update the value function recursively. Another method developed by Albrecht where the value function is approximated by a series expansion and optimal control law is written as a function of the high order nonlinear feedback terms [23–25]. Furthermore, computational approaches such as, adaptive finite difference [26], Galerkin method [27,28,61], finite element [29], collocation methods [30, 62, 63], and level set methods [31, 64] to directly solve the the HJB equation has been considered for problems not involving terminal time constriants. A major disadvantage of the different numerical approximation methods to derive optimal feedback laws is that all of them assume a fixed structure for the controller before solving for the feedback control law. Finally, most of the aforementioned techniques suffer greatly from *curse of dimensionality* and hence the computational burden associated with their

implementation makes these approaches impractical for real-time applications such as hypersonic flight trajectory guidance. The recently formulated sparse collocation method [32–37] can overcome the problem of curse of dimensionality in obtaining the optimal feedback control law.

The HJB equation with terminal time constraint is more difficult to solve since the final value function is only defined for terminal manifold. The terminal constraint can be added to the boundary conditions using the Lagrange multiplier method. The terminal constraint's Lagrange multipliers can be determined for a given trajectory with open-loop solutions. Only few methods exist to obtain the Lagrange multipliers in the case of closed-loop solutions. As an extension to classical LQR problem, the Bryson's sweep method provides the optimal feedback solutions for linear systems with quadratic performance index and terminal constraints [17]. In reference [65] the linearization method has been used to perform a closed-loop solution for optimal control with terminal constraint. Vadali and Sharma use a series expansion approach to expand the value function into power series in terms of state variables and the Lagrange multipliers [25]. The expanded value function is substitude into the HJB equation, and then by collecting the terms the governing ODEs of control gains and the Lagrange multipliers can be obtained. The most important challange in this method is the curse of dimensionality. In references [36, 37] it has been shown that how the sparse collocation method can handle the existence of the terminal constraint in the OCP. Sparse collocation method makes use of non-product quadrature method known as *Conjugate Unscented Transformation* (CUT) [38] to generate optimal sample points in the domain of interest for initial conditions. For each of these CUT points, an open-loop OCP is solved to generate optimal control profile, co-states and the cost-to-go function (i.e., value function). The value function is approximated as a polynomial series of state and the Lagrange multipliers corresponding to terminal constraints. The unknown coefficients of this series expansion are solved by solving an iterative l_1 optimization problem at each time subject to the constraints that the polynomial series is able to exactly reproduce value function and co-state values for CUT points. Qualitatively speaking, an iterative l_1 optimization problem has the tendency to produce the sparse series expansion and hence, it automatically chooses the appropriate feedback structure for the value function and the controller. In our previous work, this method has been shown to exactly reproduce the analytical solution of the HJB equation for

the infinite horizon spin stabilization problem [32]. Also this method has been used recently to solve the OCP of spacecraft attitude maneuver with the fixed final time and including a terminal constraint on the state variable [37]. Building upon those promising prior results, this thesis solves the free final-time hypersonic flight problem to maximize the impact energy of the glide weapon at the target.

Chapter 3
Hypersonic Reentry Vehicle Trajectory Optimization Problem

This chapter presents the process of obtaining the open-loop solution for the optimal control problem corresponding to the optimizing the hypersonic reentry trajectory in a glide turn maneuver. The open-loop solutions generated in this chapter will be expolited to derive optimal feedback control laws in the next chapter.

3.1 Problem Statement

Consider the problem of hypersonic flight trajectory optimization with the objective of maximizing the impact energy and hence the terminal velocity at known target

Table 3.1: State and control variables

Symbol	Variable(units)
h	Altitude (m)
v	Velocity magnitude (m/s)
γ	Flight path angle (rad)
ϕ	Latitude (rad)
θ	longitude (rad)
ψ	Atmospheric relative heading angle (rad)
α	Angle of attack (rad)
σ	Bank angle (rad)

location. For this purpose, the following low-fidelity flight dynamics model [66] has been considered for the forward-in-time evolution of vehicle states:

$$\begin{cases} \dot{h} = v \sin\gamma \\ \dot{\theta} = \dfrac{v \cos\gamma \sin\psi}{r \cos\phi} \\ \dot{\phi} = \dfrac{1}{r} v \cos\gamma \cos\psi \\ \dot{v} = -\dfrac{D}{m} - \dfrac{\mu \sin\gamma}{r^2} \\ \dot{\gamma} = \dfrac{L \cos\sigma}{mv} - \dfrac{\mu \cos\gamma}{vr^2} + \dfrac{v}{r} \cos\gamma \\ \dot{\psi} = \dfrac{L \sin\sigma}{mv \cos\gamma} + \dfrac{v}{r} \cos\gamma \sin\psi \tan\phi \end{cases} \tag{3.1}$$

Table 3.1 provides the physical description of states $(h, \theta, \phi, v, \gamma, \psi)$ and control variables (α, σ) for the aforementioned dynamical model. Furthermore, $r = h + r_e$, and expressions for the lift L and drag D are given as follows:

$$\begin{aligned} L &= \frac{1}{2} \rho v^2 C_l A_{ref} \\ D &= \frac{1}{2} \rho v^2 C_d A_{ref} \end{aligned} \tag{3.2}$$

An exponential density model is assumed for the atmosphere density ρ and the lift coefficient C_l and drag coefficient C_d are assumed to be given as follows [7]:

$$\begin{aligned} \rho &= \rho_0 \exp(-\frac{h}{H}) \\ C_d &= 1.7\alpha^2 + 0.06 \\ C_l &= 1.6\alpha \end{aligned} \tag{3.3}$$

The various environmental parameters and vehicle specific parameters are summarized in Table 3.2 and are adapted from Ref. [7].

Table 3.2: Physical parameters

Parameter Value	Parameter Description
$r_e = 6.378 \times 10^6\ m$	Radius of Earth
$\mu = 3.986 \times 10^{14}\ m^3/s^2$	Gravitional constant (Earth)
$\rho_0 = 1.2\ kg/m^3$	Surface density
$H = 7500\ m$	Scale height
$m = 340\ kg$	Mass
$A_{ref} = 0.3\ m^2$	Reference area

The solution of a trajectory optimization problem is often obtained same as the open-loop solution of the optimal control problem (3.4)-(3.6):

$$J = \Phi(\mathbf{x}(t_f), t_f) + \int_{t_0}^{t_f} l(\mathbf{x}(t), \mathbf{u}(t), t)dt \tag{3.4}$$

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) \tag{3.5}$$

$$\mathbf{\Psi}(\mathbf{x}(t_f), t_f) = 0 \tag{3.6}$$

where J in equation (3.4) is the cost functional which should be minimized and includes terminal cost $\Phi(\mathbf{x}(t_f), t_f)$, and path cost $l(\mathbf{x}(t), \mathbf{u}(t))$. Equation (3.5) represents the equation of motion and equation (3.6) represents the terminal constraint. The terminal constraint implies the value of all, or a subset of, the state variables at the final time. One can use the theory of Lagrange multipliers to augment the terminal constraint to cost functional:

$$J = \Phi(\mathbf{x}(t_f), t_f) + \boldsymbol{\nu}^T \mathbf{\Psi}(\mathbf{x}(t_f), t_f) + \int_{t_0}^{t_f} l(\mathbf{x}(t), \mathbf{u}(t), t)dt \tag{3.7}$$

Table 3.3: Boundary constraints for a hypersonic glide turn

State	h	θ	ϕ	v	γ	ψ
Initial value	80000	0	0	5000	Free	0
Final value	0	$\frac{\pi}{90}$	$\frac{\pi}{900}$	Free	Free	Free

where $\boldsymbol{\nu}$ is the vector of Lagrange multipliers corresponding to the terminal constraint.

By defining the state vector as $\mathbf{x}^T = [h, \theta, \phi, v, \gamma, \psi]$ and the control vector as $\mathbf{u}^T = [\alpha, \sigma]$, the optimal control problem associated with the hypersonic reentry vehicle trajectory optimization problem can be compactly summarized in (3.8):

$$\begin{aligned} &\min_{\mathbf{u}(t)} J = -v(t_f) \\ &\text{Subject to :} \quad \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) \\ &\boldsymbol{\Psi}(\mathbf{x}(t_f), t_f) = 0 \qquad t_f \text{ is free} \end{aligned} \tag{3.8}$$

In this problem the goal is to maximize the final velocity, therefore the cost function J doesn't include any path cost and only has the terminal cost part. Notice that the function $\mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t)$ is given by (3.1). In this problem the final time t_f is free and the terminal constraint is defined as: $\boldsymbol{\Psi} = [x_1(t_f),\ x_2(t_f) - \pi/90,\ x_3(t_f) - \pi/900]^T$.

3.2 Open-Loop Solution

As discussed in the last chapter, numerical methods to find the open-loop solution are divided into two categories: indirect methods and direct methods. The indirect methods use the result of Pontryagin's minimum principle and yield a two point boundary value problem (TPBVP). On the other hand, direct methods generally transform the optimal control problem to a nonlinear programming problem (NLP). The advantage of the direct method is its typically superior numerical robustness to the initial guess [48, 67]. In this work, a combination of both direct and indirect methods is used to solve the open-loop problem corresponding to each

CUT point. First, the pseudo spectral methods [50, 51, 68–71] is applied to find the open-loop solution which is used further as initial guess to solve the TPBVP to refine the solution provided by the psuedo spectral methods. The MATLAB toolbox DIDO [72] is used for the application of the psuedo-spectral method (direct approach). Furthermore, the MATLAB's inbuilt function *bvp4c* is used to solve the underlying TPBVP problem with the initial guess provided by DIDO solution.

The Hamiltonian of the optimal control problem (3.11) is given as:

$$H = \boldsymbol{\lambda}^T \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) \tag{3.9}$$

where $\boldsymbol{\lambda}^T = [\lambda_h,\ \lambda_\theta,\ \lambda_\phi,\ \lambda_v,\ \lambda_\gamma,\ \lambda_\psi]^T$ is the co-state vector. The first order necessary conditions for optimality can be derived by applying the calculus of variations method as:

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) \\ \dot{\boldsymbol{\lambda}} = -H_{\mathbf{x}}(\mathbf{x}, \mathbf{u}, t) \\ H_{\mathbf{u}} = 0 \\ \dfrac{dH}{dt} = \dfrac{\partial(-v_f)}{\partial t}\Big|_{t_f} = 0 \\ \boldsymbol{\lambda}^T(0)\delta\mathbf{x}(0) = 0 \\ (\boldsymbol{\lambda}(t_f) - \dfrac{\partial(-v_f)}{\partial \mathbf{x}}\Big|_{t_f} - \boldsymbol{\nu}^T \dfrac{\boldsymbol{\partial\Psi}}{\partial \mathbf{x}}\Big|_{t_f})^T \delta\mathbf{x}(t_f) = 0 \end{cases} \tag{3.10}$$

Where $\boldsymbol{\nu}$ is the Lagrange multiplier corresponds to the terminal constraint [17]. By substituting costate vector and system equations in (3.9), Hamiltonian can be rewritten as:

$$H = \lambda_h \dot{h} + \lambda_\theta \dot{\theta} + \lambda_\phi \dot{\phi} + \lambda_v \dot{v} + \lambda_\gamma \dot{\gamma} + \lambda_\psi \dot{\psi} \tag{3.11}$$

Now from the second equation from the necessary conditions (3.10) the governiong differential equation of costates can be derived as (3.12) - (3.17):

$$\dot{\lambda}_h = -H_h \;\Rightarrow\; \dot{\lambda}_h = \frac{v\cos\gamma}{r^2}(\frac{\lambda_\theta\cos\psi}{\cos\phi} + \lambda_\phi\sin\psi + \lambda_\gamma - \lambda_\psi\cos\psi\tan\phi) + \frac{L}{mHv}(\lambda_\gamma\cos\sigma + \frac{\lambda_\psi\sin\sigma}{\cos\gamma}) - \frac{2\mu}{r^3}(\lambda_v\sin\gamma + \frac{\lambda_\gamma\cos\gamma}{v}) - \frac{\lambda_v D}{mH} \tag{3.12}$$

$$\dot{\lambda}_\theta = -H_\theta \;\Rightarrow\; \dot{\lambda}_\theta = 0 \tag{3.13}$$

$$\dot{\lambda}_\phi = -H_\phi \;\Rightarrow\; \dot{\lambda}_\phi = \frac{v\cos\gamma\cos\psi}{r\cos^2\phi}(\lambda_\psi - \lambda_\theta\sin\phi) \tag{3.14}$$

$$\dot{\lambda}_v = -H_v \;\Rightarrow\; \dot{\lambda}_v = \frac{\rho A_{ref}}{2m}(2\lambda_v v C_d - \lambda_\gamma C_l\cos\sigma - \lambda_\psi C_l\frac{\sin\sigma}{\cos\gamma}) - \frac{\cos\gamma}{r}(\frac{\lambda_\theta\cos\psi}{\cos\phi} + \lambda_\phi\sin\psi + \lambda_\gamma(1+\frac{\mu}{rv^2}) - \lambda_\psi\cos\psi\tan\phi) - \lambda_h\sin\gamma \tag{3.15}$$

$$\dot{\lambda}_\gamma = -H_\gamma \;\Rightarrow\; \dot{\lambda}_\gamma = \frac{v\sin\gamma}{r}(\frac{\lambda_\theta\cos\psi}{\cos\phi} + \lambda_\phi\sin\psi + \lambda_\gamma - \lambda_\psi\cos\psi\tan\phi) + \frac{\mu}{r^2}(\lambda_v\cos\gamma - \frac{\lambda_\gamma\sin\gamma}{v}) - \lambda_h v\cos\gamma - \frac{\lambda_\psi L\sin\sigma\sin\gamma}{mv\cos^2\gamma} \tag{3.16}$$

$$\dot{\lambda}_\psi = -H_\psi \;\Rightarrow\; \dot{\lambda}_\psi = \frac{v\cos\gamma}{r}(\frac{\lambda_\theta\sin\psi}{\cos\phi} - \lambda_\phi\cos\psi - \lambda_\psi\sin\psi\tan\phi) \tag{3.17}$$

The boundary conditions of costate variables can be determined from the last two equations from the necessary conditions (3.16) and corresponding boundary conditions for state variable, the table 3.4 represents the boundary condition for costate variables.

Table 3.4: Boundary conditions for costate variables

Costate	λ_h	λ_θ	λ_ϕ	λ_v	λ_γ	λ_ψ
Initial value	$Free$	$Free$	$Free$	$Free$	0	$Free$
Final value	$Free$	$Free$	$Free$	-1	0	0

Using the stationarity condition $H_{\mathbf{u}} = 0$ with $\mathbf{u}^T = [\alpha,\ \sigma]$, the relationship between optimal control profile, co-states and state is given as follows:

$$H_\sigma = 0 \ \Rightarrow \ \sigma^* = \tan^{-1}(\frac{\lambda_\psi}{\lambda_\gamma \cos\gamma}) \tag{3.18}$$

$$H_\alpha = 0 \ \Rightarrow \ \alpha^* = \frac{1.6}{3.4\lambda_v v}(\lambda_\gamma \cos\sigma^* + \frac{\lambda_\psi \sin\sigma^*}{\cos\gamma}) \tag{3.19}$$

One can substitute the control variables from (3.18), and (3.19) into the system equations (3.1), and costate equations (3.12) - (3.17) to construct a system of differential equations including 12 ODEs (6 state equations and 6 costate equations) which are only function of states and costates. Considering the boundary conditions for state and costate equations from table 3.3, and table 3.4 with the system of ODEs for state and costate, the optimal control problem can be convert to a TPBVP, which can be solved with numerical methods which are discussed earlier. The only challange in solving this TPBVP is that the final time is free. To overcome this difficulty one can use following change of variable to solve the the TPBVP in fixed final time:

$$\tau = \frac{t}{t_f} \tag{3.20}$$

where τ is the new time variable which is fixed between 0, and 1, using this change of variable adds the final time t_f to the TPBVP as a unknown parameter which should be determined during the solution process. Another advantage of using normalized time variable is that all the trajectories with different final time t_f can be compared at the τ domain. First, the DIDO (taken from queen Dido) [72] as a direct method (Pseudospectral method) is used with 30 nodes at Legendre-

Gauss-Lobatto points, and using these 30 nodes as an initial guess for Bvp4c the TPBVP is solved. Figures 3.1-3.3 show the state, costate, and control trajectories, which are obtained by solving the open-loop problem. The boundary conditions is considered as Table 3.3. The final time after the solution has been obtained as $t_f = 153.92\ s$.

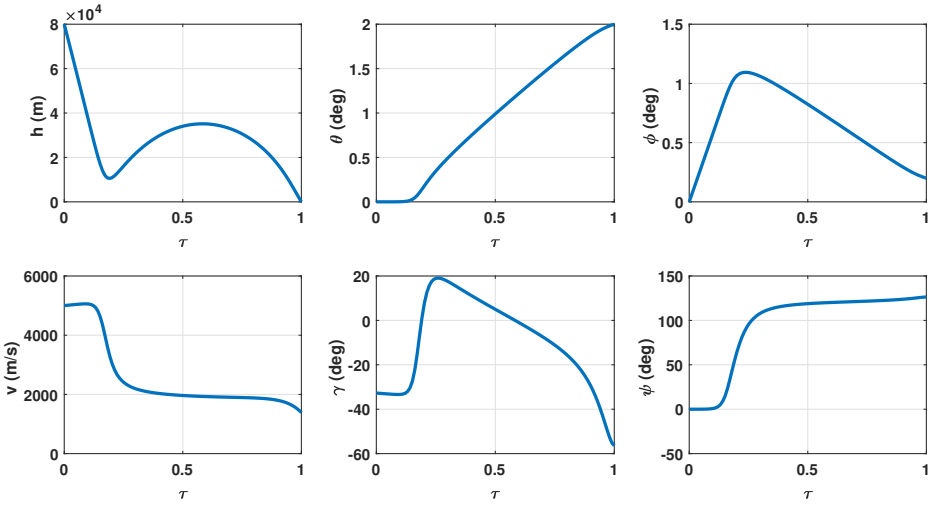


Figure 3.1: State trajectories resulted from the open-loop solution

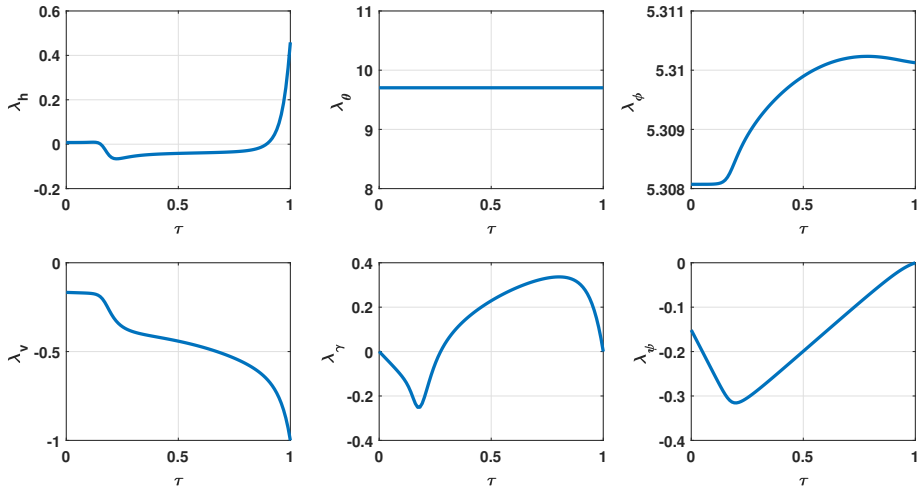


Figure 3.2: Costate trajectories resulted from the open-loop solution

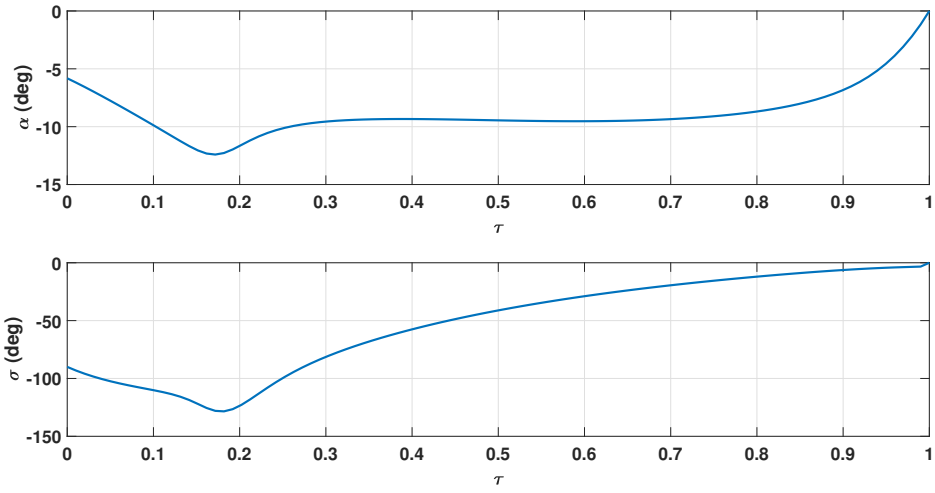


Figure 3.3: Control trajectories resulted from the open-loop solution

Chapter 4
Optimal Feedback Control Design for Hypersonic Reentry Vehicles

The last chapter presented the open-loop solution for a hypersonic reentry trajectory optimization problem. In this chapter, a new method based on both the open-loop solution and recently developed sparse collocation method [32–37] is provided to obtain optimal feedback control law for the same problem. To find the optimal feedback control, one needs to solve the HJB equation with terminal constraints, which is a challenging task. In the proceeding, an introduction to the sparse collocation method is provided and next the proposed method is introduced, and finally the results are presented.

4.1 Sparse Collocation Method

Consider the problem of approximating a function $V(\mathbf{x})$ with a finite series expansion. In this respect, the following truncated expansion is considered as an approximation of $V(\mathbf{x})$:

$$\tilde{V}(\mathbf{x}) = \sum_{i=1}^{m} c_i \phi_i(\mathbf{x}) \tag{4.1}$$

where the c_i's are unknown coefficients and the ϕ_i's are known basis functions,

generally, chosen to be orthogonal polynomial basis functions. Equation (4.1) can be written in a matrix form as:

$$\tilde{V}(\mathbf{x}) = \mathbf{C}^T \mathbf{\Phi}(\mathbf{x}) \tag{4.2}$$

Where $\mathbf{C} = [c_1, c_2, ..., c_m]^T$, and $\mathbf{\Phi}(\mathbf{x}) = [\phi_1(\mathbf{x}), \phi_2(\mathbf{x}), ..., \phi_m(\mathbf{x})]^T$. The error function then can be defined as:

$$e(\mathbf{x}) = V(\mathbf{x}) - \tilde{V}(\mathbf{x}) \tag{4.3}$$

Notice that the error is a result of using the truncated form of the function series. Our goal is to find the unknown coefficients which minimize the error function in equation (4.3). A good way of minimizing the error is to use weighted residuals method which simply try to minimize the integral of the error multiplied by a weight function over the domain:

$$\int_\Omega \Psi_j(\mathbf{x}) e(\mathbf{x}) d\mathbf{x} = 0 \qquad j = 1, 2, \ldots, m \tag{4.4}$$

where Ω represents the domain and $\mathbf{\Psi}(\mathbf{x}) = [\Psi_1(\mathbf{x}), \Psi_2(\mathbf{x}), ..., \Psi_m(\mathbf{x})]^T$ is a set of weight functions which are linearly independent. The weighted residual method is actually trying to set the projection of the error function onto the selected test functions (weight functions) to be zero.

Based on the choice of test functions the method of solving the problem will be different. Taking the test functions same as the basis functions leads to the Galerkin method [28, 61, 73]. The Galerkin projection method provides a system of equations for the unknown coefficients by projecting the error function onto the same basis functions that are used to construct the approximated function in equation (4.1) and setting these projections equal to zero:

$$\int_\Omega \phi_j(\mathbf{x}) e(\mathbf{x}) d\mathbf{x} = 0 \qquad j = 1, 2, \ldots, m \tag{4.5}$$

Using this method leads to m independent equations which can be solved to find the m unknown coefficients. The least squares problem can be formulated by choosing the test functions as $\Psi_j(\mathbf{x}) = \frac{\partial e(\mathbf{x})}{\partial c_j}$:

$$\int_\Omega \frac{\partial e(\mathbf{x})}{\partial c_j} e(\mathbf{x}) d\mathbf{x} = \frac{\partial}{\partial c_j} \int_\Omega e^2(\mathbf{x}) d\mathbf{x} = 0 \qquad j = 1, 2, \ldots, m \tag{4.6}$$

Note that for the function approximation problem, both the least square and Galerkin method lead to a same solution. This is due to the fact that for this special problem $\frac{\partial e(\mathbf{x})}{\partial c_j} = \phi_j(\mathbf{x})$.

If the test functions are set to be dirac-delta functions at specific points within the domain ($\Psi_j(\mathbf{x}) = \delta(\mathbf{x} - \mathbf{x}_j)$), the weighted residual method leads to the collocation method:

$$\int_\Omega \delta(\mathbf{x} - \mathbf{x}_j) e(\mathbf{x}) d\mathbf{x} = e(\mathbf{x}_j) = 0 \qquad j = 1, 2, \ldots, m \tag{4.7}$$

In all types of weighted residual methods, the solution of the original fitting problem is converted into a problem of solving a linear set of m equations to find m unknown coefficients, but in the first two methods (Galerkin, and Least squares) multi-dimension projection integrals have to be computed. The compution of multi-dimension integrals is expensive especially when the state-space dimension or order of basis functions are large [61]. In addition, the collocation method rather than integration process, tries to make the zero error at specially chosen collocation points.

4.1.1 Collocation Method

It has been discussed that collocation method forces the error to be exactly zero at chosen collocation points. Consider that there exist N collocation points available over the domain, therefore error function must be be zero at N points, which means the approximated function must be equal to true function at these points:

$$e(\mathbf{x}_i) = 0 \longrightarrow V(\mathbf{x}_i) = \mathbf{C}^T \mathbf{\Phi}(\mathbf{x}_i) \qquad i = 1, 2, \ldots, N \tag{4.8}$$

which leads to the following linear system of equations:

$$\bar{\mathbf{V}} = \begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ V_N \end{bmatrix} = \begin{bmatrix} \phi_1(\mathbf{x}_1) & \phi_2(\mathbf{x}_1) & \dots & \phi_m(\mathbf{x}_1) \\ \phi_1(\mathbf{x}_2) & \phi_2(\mathbf{x}_2) & \dots & \phi_m(\mathbf{x}_2) \\ \vdots & \vdots & \dots & \vdots \\ \phi_1(\mathbf{x}_N) & \phi_2(\mathbf{x}_N) & \dots & \phi_m(\mathbf{x}_N) \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_m \end{bmatrix} = \mathbf{AC} \tag{4.9}$$

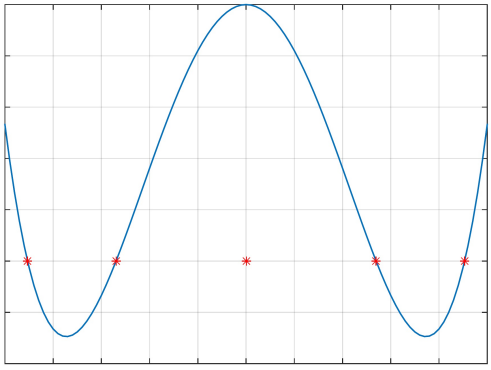
Solving the collocation equations (4.9), first requires a set of collocation points, and then a complete set of basis functions, since the number of collocation points N and the number of basis functions m can be different one may face to the following cases:

1. $\mathbf{m < N} \longrightarrow$ **Over-Determined System:** $\longrightarrow$ No Solution
2. $\mathbf{m = N} \longrightarrow$ **Square System:** $\longrightarrow$ Unique Solution
3. $\mathbf{m > N} \longrightarrow$ **Under-Determined System:** $\longrightarrow$ Infinitely Many Solutions

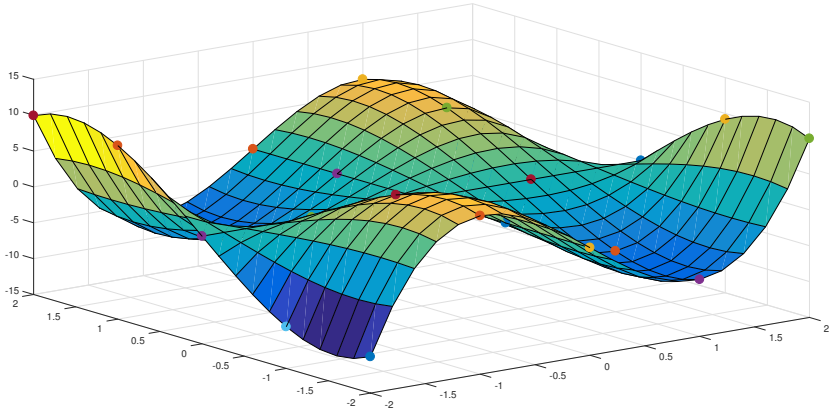
From the above, it is desired, the system of equations to be a square system or at least Under-Determined System, therefore it has to be guaranteed that the number of collocation points must be equal or less than the number of basis functions($N \leq m$). One may say, by increasing/decreasing the collocation points a square system can be obtained, but actually the equality in number of collocation points and the number of basis functions is not a viable solution because the growth rate in collocation points and basis functions are different, therefore the main challenge lies in choosing appropriate collocation points and the basis functions. To handle the curse of dimensionality challenge, it is desired to use minimal set of collocation points and also minimal set of basis functions.

In $1-D$, using the Gaussian quadrature points as collocation points with the Lagrange interpolation polynomials as basis functions is the optimal choice, in this case a $N-1$ degree Lagrange polynomial is required for N collocation points, but as the dimension of the system is increased, for both the basis functions and collocation points a tensor product is required, for example, in $1-D$ for 5 collocation points a 4^{th} order Lagrange polynomial must be used, but in $2-D$ the number of collocation points are $5 \times 5 = 25$ which needs a 8^{th} order Lagrange polynomial. A comparison

between using the lagrange interpolation in $1-D$ and $2-D$ is shown in 4.1(a). This increase in the order of the polynomial has a consequence is called the Gibbs phenomenon [74] which causes incorrect interpolation and also large oscillation in the boundaries.



(a) 4^{th} order Lagrange polynomial:$1-D$



(b) 8^{th} order Lagrange polynomial:$2-D$

Figure 4.1: Lagrange interpolation polynomials

As shown above, the performance of the collocation method is affected directly by the choice of the collocation points and the basis functions.

In this thesis the recently developed Conjugate Unscented Transform (CUT) is used to develope the minimal set of collocation points [75].

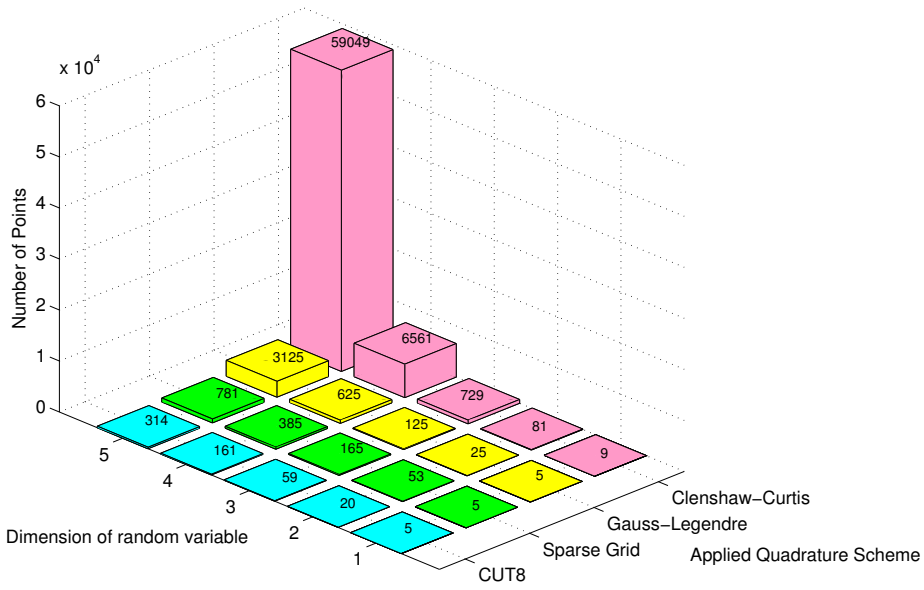


Figure 4.2: Comparison of quadrature schemes

4.1.2 Collocation Points Generation

The appropriate collocation points selection leads to a well-conditioned linear system of equations. In this work, the CUT methodology is used to produce the desired minimal set of collocation points [76]. Choosing the collocation points to be CUT points is guaranteed to lead to an *under-determined* system. This is due to the fact that the growth of CUT points is slower than the equivalent Gaussian quadrature points as well as combinatorial growth of the basis functions. Figure 4.2 shows a comparison of the CUT method with other equivalent order quadrature methods. More details about the CUT methodology can be found in Ref [38, 75, 77–79].

4.1.3 Optimal Selection of Basis Functions

As discussed a good performance in collocation method depends on the selection of basis functions, and in addition to number of collocation points the minimum number of basis functions is desired. By using the CUT collocation points it would be guaranteed that the system of equations is *under-determined*, but still a minimal set of basis functions with the least degree is desired to avoid the expensive computational effort. In this regard, a new sparse optimization based approach has been used to select the proper basis functions which is best suited for the given

collocation points and the system dynamic.

The introduced approach solves the collocation equations (4.9) by minimizing the l_1-norm of the vector $\mathbf{C}$. Ideally, l_0-norm of the coefficient vector should be minimized because l_0-norm corresponds to the cardinality of the vector $\mathbf{C}$, but the it leads to a optimization problem which is non-convex. Minimizing l_1-norm makes the coefficients to be close to zero therefore it is a good approximation of l_0-norm optimization, and also it is convex. Therefore to find the minimumal set of basis functions following sparse convex optimization problem must be solved iteratively:

$$\min_{\mathbf{C}} \quad \|\mathbf{WC}\|_1 \tag{4.10}$$

$$\text{Subject to :} \quad \bar{V} = \mathbf{AC} \tag{4.11}$$

Equation (4.10) minimizes the l_1-norm of coefficient vector, where $\mathbf{W}$ is a diagonal weighting matrix, and equation (4.11) represents the collocation constraints. Initially, the weight matrix is assumed to be diagonal matrix with diagonal term being proportional to the order of the basis functions. This makes sure that higher order monomials are penalized more in the optimization process. However, to derive the small coefficient values to zero, one can iteratively solve the aforementioned l_1-norm optimization problem with weight matrix being inversely proportion to the corresponding coefficient values from previous iteration:

$$\begin{gathered} \mathbf{W} = diag[W_1,\ W_2,\ \ldots, W_m] \\ W_l{}^{(iter)} = \frac{1}{(c_l{}^{(iter-1)} + \varepsilon)}\ , \qquad l = 1,\ 2,\ \ldots,\ m \end{gathered} \tag{4.12}$$

where the ε is a small number to avoid from the numerical issues when the coefficient is close to zero. The iterative l_1-norm minimization problem has the tendency to produce the solution corresponding to l_0-norm minimization, i.e., the sparse solution [80].

4.2 Optimal Feedback Solution

As discussed in chapter 2, in the presence of the terminal manifold constraint the value function at final time is only defined on the terminal manifold. In other words, the value function at the final time is defined only at those points where terminal constraint is satisfied:

$$\begin{aligned} V(\mathbf{x}(t_f), t_f) = \Phi(\mathbf{x}(t_f), t_f) \quad & \boldsymbol{\Psi}(\mathbf{x}(t_f), t_f) = 0 \\ V(\mathbf{x}(t_f), t_f) = \infty \qquad\quad & \boldsymbol{\Psi}(\mathbf{x}(t_f), t_f) \neq 0 \end{aligned} \tag{4.13}$$

The Lagrange multiplier, $\boldsymbol{\nu}$, is introduced to account for terminal manifold constraint. This results in the following augmented boundary condition:

$$V(\mathbf{x}(t_f), t_f) = -x_4(t_f) + \boldsymbol{\nu}^T \boldsymbol{\Psi}(\mathbf{x}(t_f), t_f) \tag{4.14}$$

Due to the terminal constraints on state vector, one has to consider the cost-to-go function to be function of system states as well as the Lagrange multipliers, $\boldsymbol{\nu}$. The motivation of making this assumption is the appearance of $\boldsymbol{\nu}$ in the boundary condition, and it is along the lines of Bryson's sweep method [17].

Hence, a truncated series expansion is assumed to approximate the value function:

$$V(\mathbf{x}, \boldsymbol{\nu}, t) = \sum_{i=1}^{m} c_i(t)\phi_i(\mathbf{x}, \boldsymbol{\nu}) = \mathbf{c}^T(t)\boldsymbol{\Phi}(\mathbf{x}, \boldsymbol{\nu}) \tag{4.15}$$

Where $\mathbf{c}(t) = [c_1(t)\ c_2(t)\ ...\ c_m(t)]^T$ is a vector of unknown coefficients which change along the time, and $\boldsymbol{\Phi}(\mathbf{x}, \boldsymbol{\nu}) = [\phi_1(\mathbf{x}, \boldsymbol{\nu})\ \phi_2(\mathbf{x}, \boldsymbol{\nu})\ ...\ \phi_m(\mathbf{x}, \boldsymbol{\nu})]^T$ is a vector of known basis functions of state vector and the lagrange multipliers.

The proposed approach is based on open-loop solution. The key point of this method is the definition of the costate vector, which is the sensitivity of value function with respect to the state vector. In other words, the costate vector is the gradient of the value function with respect to the state vector:

$$\boldsymbol{\lambda} = \frac{\partial V(\mathbf{x}(t), t)}{\partial \mathbf{x}} \tag{4.16}$$

The above equation is the key equation in the following proposed methodology, Equation (4.16) links the open-loop solution to the optimal value function. The optimal open-loop solution provides the gradient of optimal value function (co-state) along an optimal trajectory. The open-loop solution provides the co-state trajectory only for a single initial condition, with the iterative numerical methods also the Lagrange multipliers corresponding to the terminal constraint can be obtained along this optimal trajectory [17, 81]. As it discussed before, these methods provide the open-loop solution which should be solved for every different initial condition. If the open-loop solution be determined for whole the state space domain then one can just use the interpolation to find the optimal control law and avoids resolving the TPBVP.

The approximated value function (4.15) can be substituted into the optimal co-state equation (4.16) instead of HJB equation. The following equation is the base of a method to find the gradient of the value function in terms of the state, and the Lagrange multipliers:

$$\boldsymbol{\lambda}^*(t|\mathbf{x}^*(t), \boldsymbol{\nu}^*) = \frac{\partial \Phi(\mathbf{x}, \boldsymbol{\nu})}{\partial \mathbf{x}}^T \mathbf{c} \tag{4.17}$$

The co-state is not a explicit function of states and Langrange multiplier, the optimal co-state trajectory $\boldsymbol{\lambda}^*(t)$ is indicated by corresponding optimal state trajectory $\mathbf{x}^*(t)$ and the optimal Lagrange multiplier $\boldsymbol{\nu}^*$.

The main idea is to match the optimal costates from the open-loop solution with gradient of the value function. To modify the sparse collocation method to use for optimal control problem with terminal constraint one can determine the value function itself from the open-loop solution and match with value function expansion (4.15).

As it's stated the value function is same as the cost-to-go function, the value of cost-to-go function at a time instance t represents the cost for the rest of the trajectory to reach to the final time t_f, which can be represent as $V(\mathbf{x}, t) = J(\mathbf{x}, t \rightarrow t_f)$, for the optimal control problem (3.8), the cost function which is desired o be

minimized only includes the final cost term, therefor there is no cost for $t < t_f$, hence along a optimal trajectory the value function is constant and equals to the final cost which is negative of the final velocity:

$$V^*(\mathbf{x}, t) = -x_4(t_f) \tag{4.18}$$

4.2.1 Proposed Methodology

For a specific initial condition $\mathbf{x}_0$, the open-loop solution of the optimal control problem (3.8) can be obtained same as the method in chapter 3, this open-loop solution provides the optimal state and co-state trajectories, $\mathbf{x}^*(t)$, $\boldsymbol{\lambda}^*(t)$, the optimal Lagrange multiplier $\boldsymbol{\nu}^*$ which is constant along the optimal trajectory, and finally the optimal value function $V^*(\mathbf{x}^*(t), t)$ which is also constant along this optimal trajectory. Now, one can use CUT quadrature points in the initial state space domain as the initial condition and obtain the open-loop solution for all of them.

After computing the open-loop solution for N CUT points, one needs to solve the sparse approximation problem to find an expression for the cost-to-go function. The optimal control can be derived from the cost-to-go function by utilizing the fact that co-state vector is the gradient of the cost-to-go function with respect to the state vector. By discretizing the normalized time variable τ from (3.20), at each time step τ_k, there would be N independent equations to satisfy, which correspond to reproducing the exact cost-to-go function from the open-loop solution.

$$V^*(\mathbf{x}_j, \boldsymbol{\nu}_j, \tau_k) = \boldsymbol{\Phi}^T(\mathbf{x}_j, \boldsymbol{\nu}_j)\mathbf{c}(\tau_k), \quad j = 1, 2, ..., N \tag{4.19}$$

Since the computation of closed-loop solution requires the knowledge of $[\lambda_v, \lambda_\gamma, \lambda_\psi]$, which are only co-states appear in optimal control expression (3.18), (3.19), also the following $3N$ collocations constraints will be added, to reproducing the exact values of $[\lambda_v, \lambda_\gamma, \lambda_\psi]$ at collocation points.

$$\lambda_v(\mathbf{x}_j, \boldsymbol{\nu}_j, \tau_k) = \frac{\partial \boldsymbol{\Phi}(\mathbf{x}_j, \boldsymbol{\nu}_j)^T}{\partial x_4}\mathbf{c}(\tau_k)$$

$$\lambda_\gamma(\mathbf{x}_j, \boldsymbol{\nu}_j, \tau_k) = \frac{\partial \boldsymbol{\Phi}(\mathbf{x}_j, \boldsymbol{\nu}_j)^T}{\partial x_5}\mathbf{c}(\tau_k) \tag{4.20}$$

$$\lambda_\psi(\mathbf{x}_j, \boldsymbol{\nu}_j, \tau_k) = \frac{\partial \boldsymbol{\Phi}(\mathbf{x}_j, \boldsymbol{\nu}_j)^T}{\partial x_6}\mathbf{c}(\tau_k)$$

$$j = 1, 2, ..., N$$

The unknown coefficient vector at each time step $\mathbf{c}(\tau_k)$ can be found by solving the iterative l_1-norm optimization method (2.8) - (2.10), subject to the $4N$ collocation constraints (4.19) - (4.20)

Since the final time t_f and state variable γ at initial time are free, the sparse collocation method can be used directly to find an expression of them in terms of initial state condition, in this regard the following series expansion is performed for them to pose two separate sparse collocation problem for each of them:

$$t_f = \sum_{i=1}^{m_{t_f}} c_{t_f i}\phi_i(\mathbf{x}_0) = \mathbf{c}_{t_f}^T \boldsymbol{\Phi}(\mathbf{x_0}) \tag{4.21}$$

$$\gamma_0 = \sum_{i=1}^{m_{\gamma_0}} c_{\gamma_0 i}\phi_i(\mathbf{x}_0^*) = \mathbf{c}_{\gamma_0}^T \boldsymbol{\Phi}(\overline{\mathbf{x}}_0) \tag{4.22}$$

Where $\mathbf{c}_{t_f}$, and $\mathbf{c}_{\gamma_0}$ are vectors of unknown coefficients, and $\boldsymbol{\Phi}(\mathbf{x_0})$, and $\boldsymbol{\Phi}(\overline{\mathbf{x}}_0)$ are known vectors of basis functions in terms of initial condition $\mathbf{x}_0$, and $\overline{\mathbf{x}}_0 = [h_0\ \theta_0\ \phi_0\ v_0\ \psi_0]^T$. one can match both the t_f, and γ_0 corresponding to the optimal trajectory for each CUT initial condition from the open-loop solution with the approximated final time in (4.21), and approximated γ_0 in (4.22), to construct the N collocation constraints which must be satisfied while minimizing the l_1-norm of the unknown coefficient vectors $\mathbf{c}_{t_f}$, and $\mathbf{c}_{\gamma_0}$.

Furthermore, the Lagrange multiplier $\boldsymbol{\nu} = [\nu_h,\ \nu_\theta,\ \nu_\phi]^T = [\lambda_h(t_f),\ \lambda_\theta(t_f),\ \lambda_\phi(t_f)]^T$

are also function of initial conditions because by changing the initial condition a new optimal trajectory would be produced which has a new Lagrange multiplier and it's constant along the trajectory, and hence, also a series expansion is performed for $\boldsymbol{\nu}$:

$$\begin{aligned}
\nu_h &= \sum_{i=1}^{m_{\nu h}} c_{\nu_h i}\phi_i(\mathbf{x}_0) = \mathbf{C}_{\nu_h}^T \mathbf{\Phi}(\mathbf{x_0}) \\
\nu_\theta &= \sum_{i=1}^{m_{\nu \theta}} c_{\nu_\phi i}\phi_i(\mathbf{x}_0) = \mathbf{C}_{\nu_\theta}^T \mathbf{\Phi}(\mathbf{x_0}) \\
\nu_\phi &= \sum_{i=1}^{m_{\nu \phi}} c_{\nu_\phi i}\phi_i(\mathbf{x}_0) = \mathbf{C}_{\nu_\phi}^T \mathbf{\Phi}(\mathbf{x_0})
\end{aligned} \tag{4.23}$$

The unknown coefficient for each series expansion are computed by solving an iterative l_1-norm optimization problem of (2.8) - (2.10). In summary, the main steps of the proposed algorithm can be enumerated as follows:

1. Generate CUT points in the domain of interest for state initial conditions.
2. Solve open-loop optimal control problem for each initial condition generate by CUT collocation points.
3. Discretize the normalized time variable $\tau \in [0\ 1]$
4. At each time step τ_k, match the optimal value function and co-state variables from the open-loop solutions with approximated variables to construct the collocation constraints and solve the l_1-norm optimization problem of (2.8) - (2.10) to solve for the unknown coefficients, $\mathbf{c}(\tau_k)$.
5. Match t_f, γ_0, ν_h, ν_θ, and ν_ϕ from the open-loop solutions with the approximated variables to construct the collocation constraints and solve the l_1-norm optimization problem of (2.8) - (2.10) to solve for the unknown coefficients coefficients, $\mathbf{C}_{t_f}$, $\mathbf{C}_{\nu_h}$, $\mathbf{C}_{\nu_\theta}$, $\mathbf{C}_{\nu_\phi}$ and $\mathbf{C}_{\gamma_0}$.
6. Producing the closed-loop solution by integrating the state equations (3.1) Using the optimal control which is calculated from using the equation (4.20)

in conjunction with equations (3.18), and (3.19).

4.3 Simulation and Results

To validate the developed approach, the hypersonic maneuver is considered to make maximum impact energy of the glide payload at the target [7] and is of significant importance to the mission of conventional prompt global strike (CGPS). Table 3.3 lists the boundary conditions for the considered maneuver and the performance index is given by (3.8). For feedback controller design purposes, the initial position of the vehicle is assumed to be uncertain with following bounds:

$$\begin{aligned} 79.5\ km \leq h_0 \leq 80.5\ km \\ -0.5^\circ \leq \theta_0 \leq 0.5^\circ \\ -0.005^\circ \leq \phi_0 \leq 0.005^\circ \end{aligned} \tag{4.24}$$

With regard to the approach outlined in the previous section, 59 CUT points in the aforementioned three dimension domain were generated. For each CUT point, the open-loop solution was found while using DIDO and bvp4c as outlined in chapter 3. The value function at each time instant is considered to be a 4^{th} order polynomial function of six-dimensional state vector $\mathbf{x}$, and three dimensional Lagrange multiplier $\boldsymbol{\nu}$, associated with terminal constraints, resulting in a total of 715 basis functions. To avoid explosion in polynomial expression of value function, $\mathbf{x}$, and $\boldsymbol{\nu}$ are normalized into the interval $[-1,\ 1]$. Defining different transformation matrices at each node to normalize $\mathbf{x}$, and $\boldsymbol{\nu}$ causes numerical issues while integrating the state equations, because for any time instant other than the preselected nodes the transformation matrix must be interpolated. Any small error in interpolation can make the variables to be out of the interval $[-1,\ 1]$ which leads to an explosion in the polynomial function during the time. In addition, defining a unique transformation matrix for all the nodes is not possible because of the large variation in the variables along the time. To overcome these issues, the following change of variable is used:

$$\begin{aligned}
\boldsymbol{\Delta}\mathbf{x}(\tau_k) &= \mathbf{x}(\tau_k) - \mathbf{x}_{mid}(\tau_k) \\
\boldsymbol{\Delta}\boldsymbol{\lambda}(\tau_k) &= \boldsymbol{\lambda}(\tau_k) - \boldsymbol{\lambda}_{mid}(\tau_k) \\
\boldsymbol{\Delta}V(\tau_k) &= V(\tau_k) - V_{mid}(\tau_k) \\
\boldsymbol{\Delta}\boldsymbol{\nu} &= \boldsymbol{\nu} - \boldsymbol{\nu}_{mid}
\end{aligned} \tag{4.25}$$

Where $k = 1, \cdots, 100$ and the index $[.]_{mid}$ refers to the middle point in the initial condition domain box. From (4.24) the middle point can be found as $[h_0, \theta_0, \phi_0]^T = [80\ km,\ 0,\ 0]^T$ which is also one of the CUT points, so the open-loop solution for it, is available. Equation (4.25) shows that the new variable at each node is made by reducing the value of the same variable at the same node from the open-loop solution of middle point. With the new variables it would be possible to define only one transformation matrix to normalize them, but still there is a problem in the last node, because three of the states are fixed at the final node to the corresponding new variables are zero, and it makes numerical issues in the next step's optimization problem. Finally, at the first 95 nodes (Part 1) the changed variables are used, and the original variables are used for the last 5 nodes (Part 2).

Table 4.1 shows the number of unknowns and equations, and the transformation matrices are used to normalize variables at two different parts of the trajectory. After solving the optimization problem, the unknown coefficients at each node were found. Fig. 4.3 shows the coefficients corresponding to the monomials appear in the value function structure in part 1. Fig. 4.4 shows the coefficients corresponding to the monomials appear in the ΔV structure in part 2. Monomials with different order are shown separately in both Fig. 4.3, and Fig. 4.4 to emphasize the high nonlinearity order of the value function.

To generate the optimal feedback control, different order polynomial series for $t_f, \nu_h, \nu_\theta, \nu_\phi$, and γ_0 are considered and an iterative sparse optimization problem is solved to find unknown polynomial coefficients, $\mathbf{C}_{t_f}$, $\mathbf{C}_{\nu_h}$, $\mathbf{C}_{\nu_\theta}$, $\mathbf{C}_{\nu_\phi}$ and $\mathbf{C}_{\gamma_0}$.

Table 4.2 lists the number of basis functions for each polynomial expansion considered and the maximum and minimum error in evaluating these variables for all 59 CUT runs. From Table 4.2, one can infer that the polynomials approximation

Table 4.1: Information of approximation problem for the value function.

Approximated Variable	Basis Function Dictionary	Number of Unknowns	Number of Equations
$\mathbf{\Delta}V$(First 95 nodes)	Polynomials up to Degree 4	715	4 x 59 = 236
Transformation Matrix= $diag$[1.4, 119, 191.5, 4, 4.9, 5.2, 49.8, 0.27, 0.16]			
V(Last 5 nodes)	Polynomials up to Degree 4	715	4 x 59 = 236
Transformation Matrix= $diag$[1.05, 28.6, 239.3, 0.6, 1, 0.45, 49.8, 0.27, 0.16]			

Table 4.2: Number of basis functions and accuracy of polynomial series expansion for different variables

Approximated Variable	Basis Function Dictionary	Number of Unknowns	Number of Equations	Max Error	Min Error
t_f	Polynomials up to degree 6	924	59	7.98×10^{-7}	1.27×10^{-9}
γ_0	Polynomials up to degree 10	3003	59	1.4×10^{-9}	1.58×10^{-11}
ν_h	Polynomials up to degree 6	924	59	4.9×10^{-8}	3.98×10^{-10}
ν_θ	Polynomials up to degree 6	924	59	3.08×10^{-6}	3.25×10^{-10}
ν_ϕ	Polynomials up to degree 6	924	59	7.6×10^{-8}	1.14×10^{-10}

of these variables can produce the true values with very good accuracy for the CUT points.

Furthermore, 6 random initial condition (cf. Table 4.3) are generated from the prescribed bounds of (4.24). A control solution is generated for these random points by using the polynomial series expansion and is compared with solving open-loop problem for each of these random initial condition. Table 4.4 and table 4.5 lists the value for various variables as obtained from the series expansion (closed-loop solution) and the open-loop solution. It should be mentioned that vehicle's altitude at polynomial series computed final time may not be exactly zero and in that case, a zero order hold is assumed on the control profile and state equations are integrated till the vehicle hit the ground. Comparing the time of the hitting and corresponding value function with the final time and value function from the open-loop solution shows the excellent accuracy of the closed-loop solution. It is clear that the series solution can reproduce the optimal open-loop solution with less than 0.5% error.

Figs. 4.5(a)-4.5(c) show the state trajectories for the nominal initial condition and two random initial conditions. The solid blue line represents the results of open-

Table 4.3: Different random initial conditions

Initial Condition	$h(m)$	$\theta(rad)$	$\phi(rad)$
Random 1	79760	-0.0017	-7.25 $\times 10^{-5}$
Random 2	80445	0.007	-7.86 $\times 10^{-5}$
Random 3	80050	0.0014	6.445 $\times 10^{-5}$
Random 4	80013	-0.0026	2.13 $\times 10^{-5}$
Random 5	79684	-0.0066	-4.54 $\times 10^{-5}$
Random 6	79764	-0.0056	7.17 $\times 10^{-5}$

Table 4.4: Comparing the final time and value function from open-loop and closed-loop solution for different random initial conditions

Initial Condition	$t_f(s)$			Value Function		
	Open loop	Closed loop	closed-loop (Hitting the ground)	Open loop	Closed loop	closed-loop (Hitting the ground)
Random 1	160.56	160.02	159.87	-1401.3	-1393.8	-1402.1
Random 2	126.2	126.2	127.17	-1327.7	-1374.9	-1329.8
Random 3	148.62	148.58	148.73	-1376.6	-1385	-1377.5
Random 4	164.48	164.33	164.41	-1406.3	-1411.5	-1407.5
Random 5	180.36	182.08	182.78	-1432.6	-1471.9	-1433.2
Random 6	176.67	174.87	174.4	-1425.4	-1399.5	-1426.7

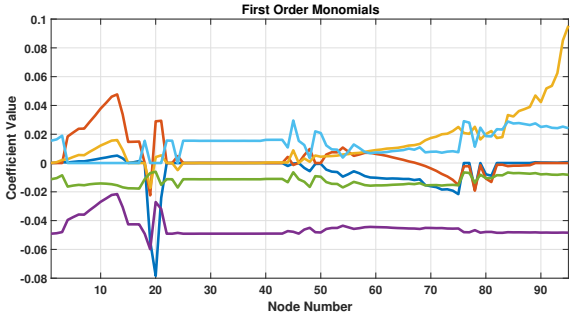
loop solution and the red dash-line corresponds to the results of closed-loop solution. Figs. 4.6(a)-4.6(c) show the optimal co-states plots for these initial conditions and Figs. 4.7(a) and 4.7(b) show the corresponding optimal control trajectories. The blue circles correspond to the open-loop solution and red markers represent the closed-loop solution. As expected, the optimal trajectories obtained from open-loop solution and the polynomial series solution match well with each other.

Finally, 59 open-loop solutions corresponding to 59 CUT initial conditions are used to generate performance based reachability sets. The CUT algorithm provides

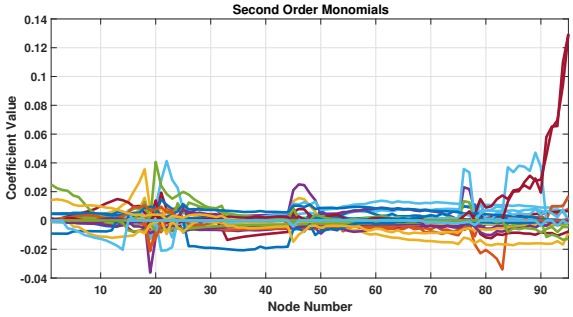
Table 4.5: Comparing Lagrange multipliers and γ_0 from open-loop and closed-loop solution for different random initial conditions

Initial Condition	$\gamma_0(rad)$		ν_h		ν_θ		ν_ϕ	
	Open loop	Closed loop	Open loop	Closed loop	Open loop	Closed loop	Open loop	Closed loop
Random 1	-0.5427	-0.5479	0.4632	0.4637	9.89	9.77	4.58	4.73
Random 2	-0.7085	-0.7087	0.4424	0.4425	6.84	6.83	10.37	10.38
Random 3	-0.5944	-0.5949	0.4551	0.4551	9.47	9.45	6	6
Random 4	-0.5299	-0.5312	0.4649	0.4650	9.92	9.89	4.2	4.24
Random 5	-0.4744	-0.4617	0.4743	0.4731	9.9	10.034	3.015	2.75
Random 6	-0.4865	-0.4998	0.4716	0.4728	9.94	9.76	3.23	3.54

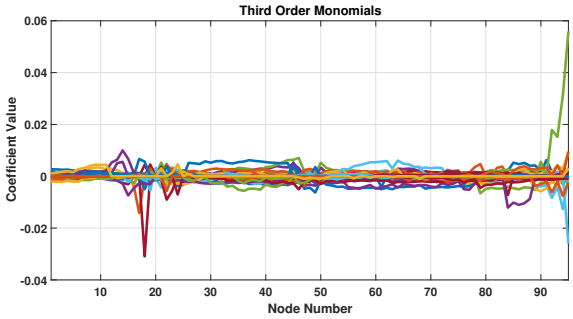
non-product cubature points, which are used to evaluate mean and variance of all possible trajectories which vehicle can have if initial condition corresponds to prescribed bounds of (4.24) and control policy is generated from (3.18) and (3.19). Fig. 4.8 shows the mean trajectory (blue line) and $\pm 3\sigma$ bounds (green lines) along with few trajectories (red dots) initiating from prescribed initial condition bounds. As expected the uncertainty in final values for h, θ and ϕ is nearly zero as feedback control law derive these states to fixed final values (cf. Table 3.3). These plots once again show the effectiveness of the sparse collocation method to generate optimal feedback laws.



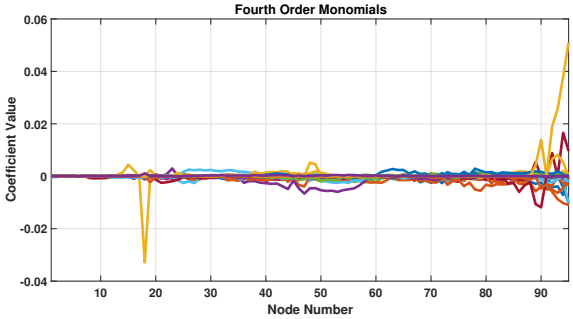
(a) First order monomials



(b) Second order monomials

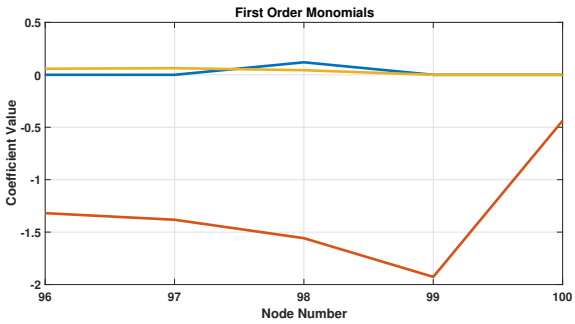


(c) Third order monomials

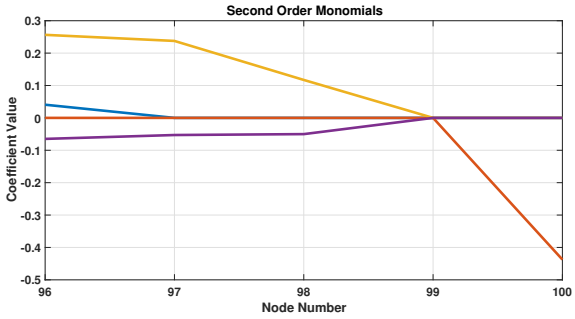


(d) Fourth order monomials

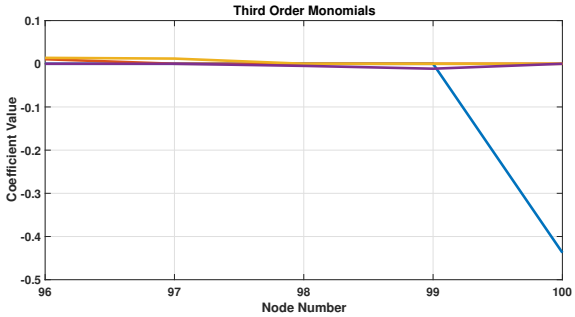
Figure 4.3: Evaluated coefficients of approximated polynomial for ΔV in the first 95 nodes



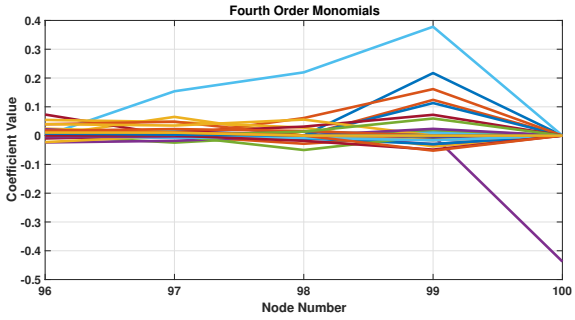
(a) First order monomials



(b) Second order monomials

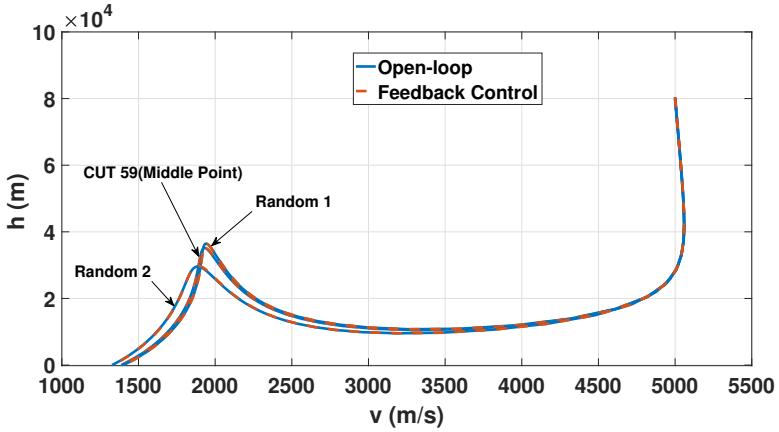


(c) Third order monomials

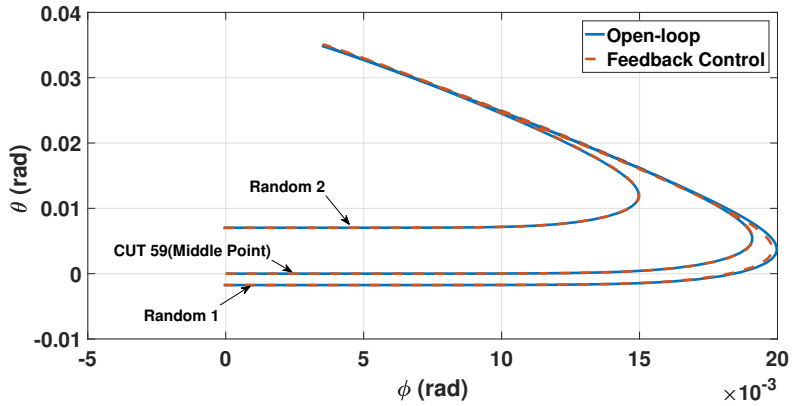


(d) Fourth order monomials

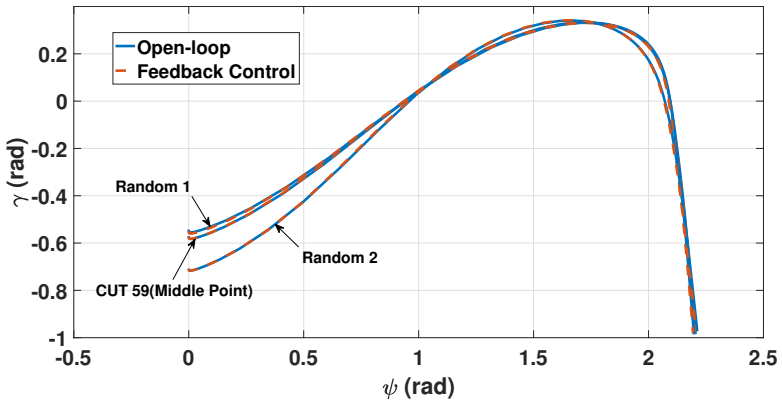
Figure 4.4: Evaluated coefficients of approximated polynomial for V in the last 5 nodes



(a) h - v

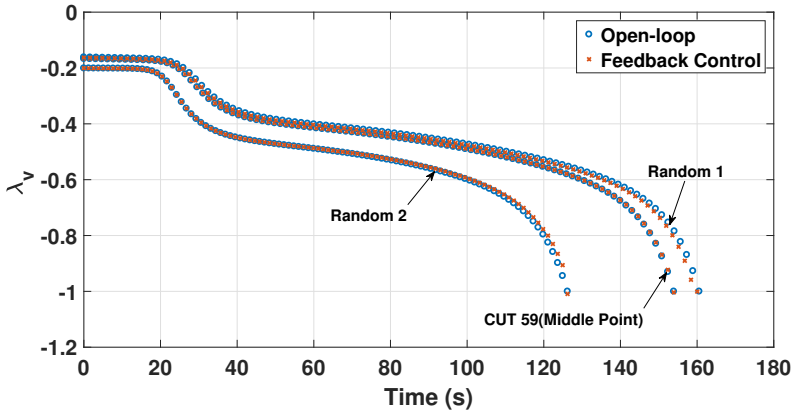


(b) θ - ϕ

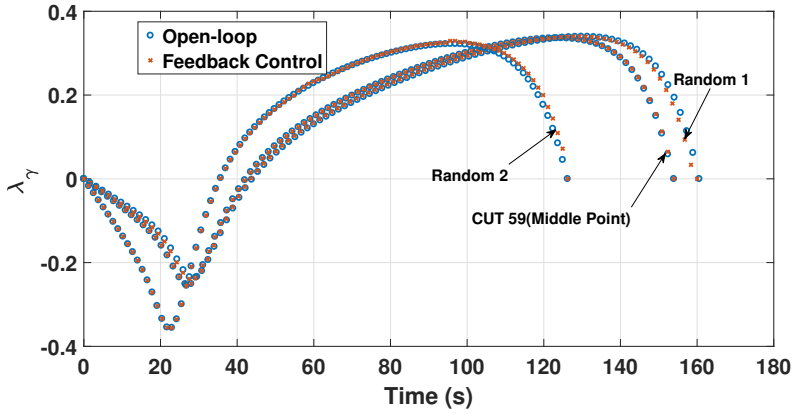


(c) γ - ψ

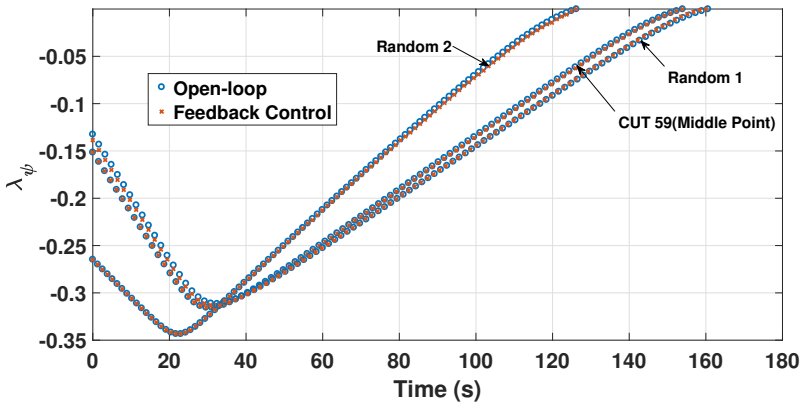
Figure 4.5: Comparing open-loop and closed-loop state trajectories



(a) λ_v - t

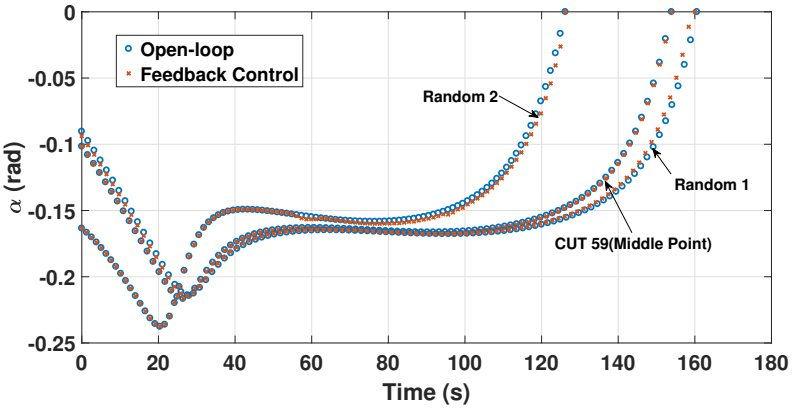


(b) λ_γ - t

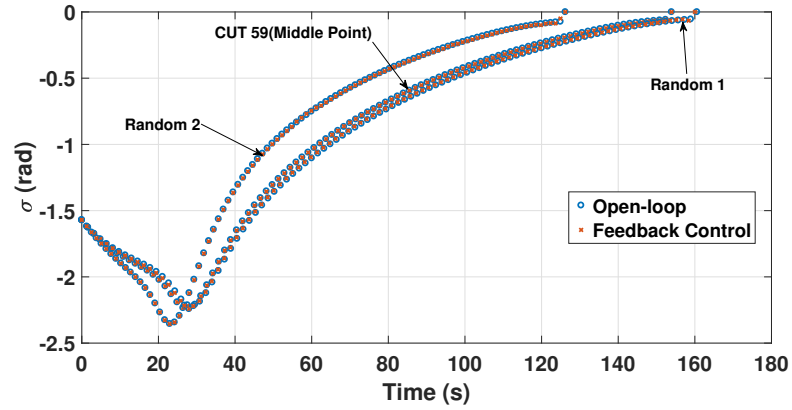


(c) λ_ψ - t

Figure 4.6: Comparing open-loop and closed-loop costate trajectories

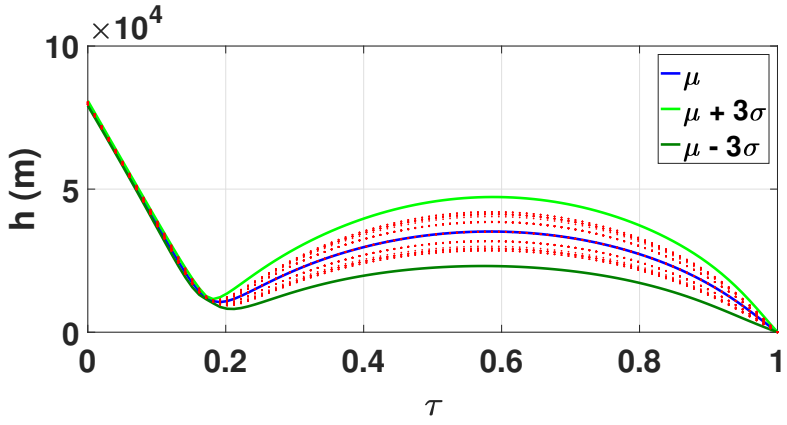


(a) α - t

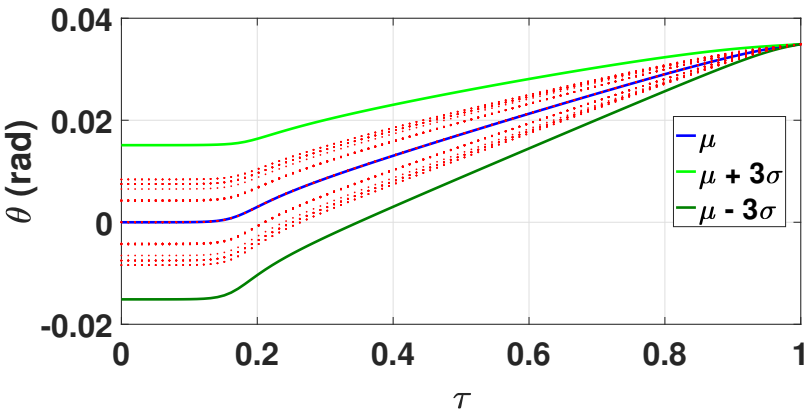


(b) σ - t

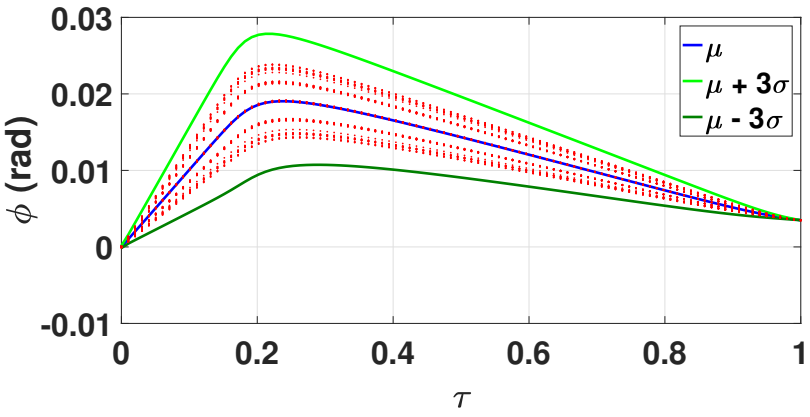
Figure 4.7: Comparing open-loop and closed-loop control trajectories



(a) h - τ

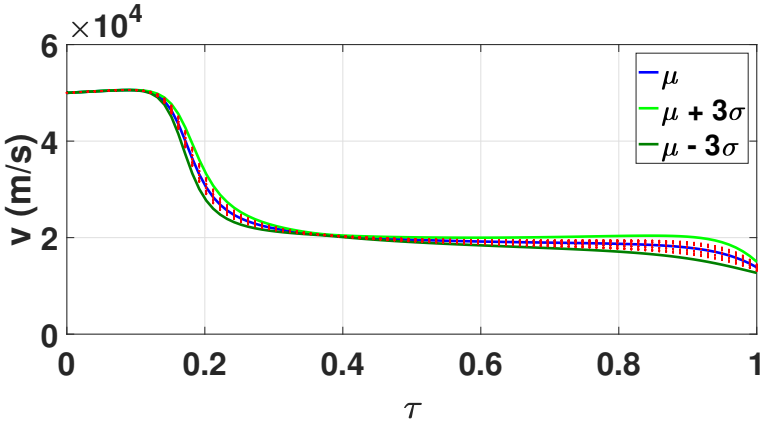


(b) θ - τ

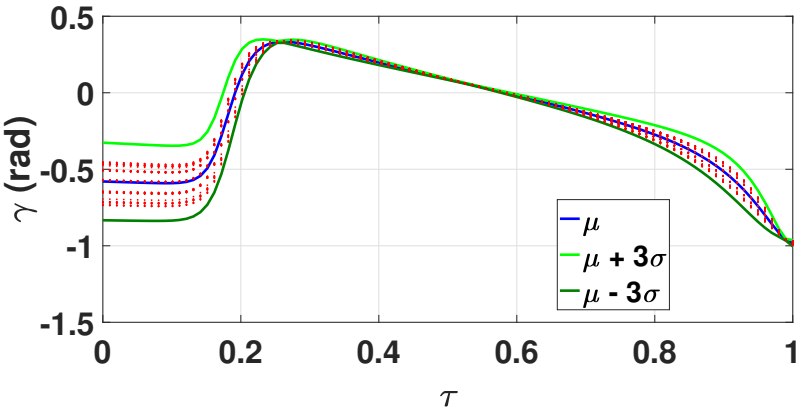


(c) ϕ - τ

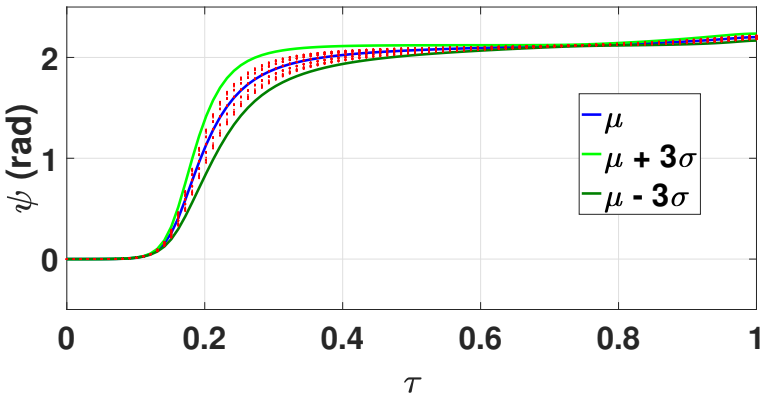
Figure 4.8: States uncertainty over time



(a) v - τ



(b) γ - τ



(c) ψ - τ

Figure 4.8: States uncertainty over time (cont.)

Chapter 5
Conclusions

The goal of this thesis was to establish a method to obtain the optimal feedback control law for hypesonic reentry flight. The guidance and trajectory planning for a hypersonic reentry vehicle is a challenging task because of the high order nonlinearity of the system equation. This problem can be posed as an optimal control problem. The open-loop solution for this optimal control problem has been addressed by many researchers, but little research has been done in the optimal feedback control design for this problem. The main shortcoming of the open-loop solution is its sensitivity to the initial condition uncertainties and state perturbations during the trajectory, and hence for any small deviation from the optimal trajectory the open-loop solution has to be resolved.

This work utilizes a sparse collocation-based approach to generate optimal feedback control laws for the hypersonic re-entry problem. The main contribution of this work is to extend the recently developed sparse collocation method to handle the non-affine control variables in system dynamics as well as extending the approach for free final time problems. The developed approach fruitfully benefits from recent advances in non-product quadrature methods known as the *Conjugate Unscented Transformation (CUT)* and l_1-norm optimization to generate optimal feedback laws from a finite number of open-loop solutions. The main advantage of the proposed solution is that it does not require any a-priori information or assumption about the optimal control profile. Furthermore, the methodology outlined here can also be used to generate performance-based reachability sets. Simulation studies presented in the paper provide a basis for optimism for the effectiveness of this approach for optimal on-board closed-loop guidance in hypersonics applications. The presented method can derive the optimal closed-loop solution from only a

few open-loop solution. The process of designing the optimal feedback control for guidance is independent of system dynamics and open-loop solutions, therefore the computation process of open-loop solutions can be done off-line, this property makes the closed-loop solution available in real time conditions.

The dynamical model considered in the problem did not take into account the actuator dynamics and the variation of aerodynamic coefficients with flight conditions. However, the developed methodology is generic in nature and can be used in conjunction with high fidelity dynamical model. Furthermore, the hypersonic reentry maneuver can consist of many complex features which depends on the desired mission, so adding these features to the problem structure can be done in the future works based on this thesis. For example, one can consider different constraints such as maximum heat transfer rate or bounded control constraints. Also, in this thesis a single phase mission has been considered which can be extended in future research to multi-phase missions like way-points tracking problem or problems with no-fly zone constraint. The proposed methodology is based on matching the costate variables from the open-loop solution and closed-loop solution, but most of the methods for solving the open-loop problem can't provide the costate variables, and it requires to mix multiple methods to derive the costate variables, which can be a challenging task.

Potential future work can also involve the design of feedback laws under model parameters uncertainty. Like the state uncertainty, multiple open-loop solutions can be generated according to the CUT method in the domain of model parameters. These open-loop solutions can be utilized in the sparse collocation framework to derive optimal gains as a function of unknown parameters. A thorough numerical analysis is also required with regard to the implementation of derived control laws in real-time using an on-board computer.

References

[1] Grant, M. J. (2012) *Rapid simultaneous hypersonic aerodynamic and trajectory optimization for conceptual design*, Ph.D. thesis, Georgia Institute of Technology.

[2] Anderson Jr, J. D. (2006) *Hypersonic and high-temperature gas dynamics*, American Institute of Aeronautics and Astronautics.

[3] Bryson, A. E. (1962) *Optimum lateral turns for a re-entry glider*, Institute of the Aerospace Sciences.

[4] Vinh, N. X. (1982) "Optimal trajectories in atmospheric flight," in *Space Mankind's Fourth Environment*, Elsevier, pp. 449–468.

[5] Chavez, F. R. and D. K. Schmidt (1994) "Analytical aeropropulsive-aeroelastic hypersonic-vehicle model with dynamic analysis," *Journal of Guidance, Control, and Dynamics*, **17**(6), pp. 1308–1319.

[6] Grant, M. J. and R. D. Braun (2014) "Rapid indirect trajectory optimization for conceptual design of hypersonic missions," *Journal of Spacecraft and Rockets*, **52**(1), pp. 177–182.

[7] Mansell, J. R. and M. J. Grant (2018) "Adaptive Continuation Strategy for Indirect Hypersonic Trajectory Optimization," *Journal of Spacecraft and Rockets*, **55**(4), pp. 818–828.

[8] Sigthorsson, D., P. Jankovsky, A. Serrani, S. Yurkovich, M. Bolender, and D. B. Doman (2008) "Robust linear output feedback control of an airbreathing hypersonic vehicle," *Journal of guidance, control, and dynamics*, **31**(4), pp. 1052–1066.

[9] Wei, X., L. Liu, Y. Wang, and Y. Yang (2018) "Reentry Trajectory Optimization for a Hypersonic Vehicle Based on an Improved Adaptive Fireworks Algorithm," *International Journal of Aerospace Engineering*, **2018**.

[10] Wilcox, Z., W. MacKunis, S. Bhat, R. Lind, and W. Dixon (2010) "Lyapunov-based exponential tracking control of a hypersonic aircraft with

aerothermoelastic effects," *Journal of guidance, control, and dynamics*, **33**(4), pp. 1213–1224.

[11] Xu, B. and Z. Shi (2015) "An overview on flight dynamics and control approaches for hypersonic vehicles," *Science China Information Sciences*, **58**(7), pp. 1–19.

[12] Zhang, K. and W. Chen (2011) "Reentry vehicle constrained trajectory optimization," in *17th AIAA International Space Planes and Hypersonic Systems and Technologies Conference*, p. 2231.

[13] Zhao, J. and R. Zhou (2013) "Reentry trajectory optimization for hypersonic vehicle satisfying complex constraints," *Chinese Journal of Aeronautics*, **26**(6), pp. 1544–1553.

[14] Schmidt, D. K. (1997) "Optimum mission performance and multivariable flight guidance for airbreathing launch vehicles," *Journal of Guidance, Control, and Dynamics*, **20**(6), pp. 1157–1164.

[15] Dreyfus, S. (1965) *Dynamic Programming and Calculus of Variations*, Academic Press.

[16] Bertsekas, D. P. (2000) *Dynamic Programming and Optimal Control, vols I and II*, Cambridge.

[17] Bryson, A. E. and Y. C. Ho (1965) *Applied Optimal Control*, Taylor and Francis.

[18] Cloutier, J. R. and D. T. Stansbery (1999) "State-Dependent Riccati Equation Techniques," in *Proceedings of the American Control Conference*, pp. 893–898.

[19] Park, C. and D. J. Scheeres (2004) "Solutions of Optimal Feedback Control Problems with General Boundary Conditions Using Hamiltonian Dynamics and Generating Functions," in *Proceedings of the American Control Conference*, Dayton, OH, pp. 679–684.

[20] Xin, M., S. N. Balakrishnan, D. T. Stansbery, and E. J. Ohlmeyer (2004) "Nonlinear Missile Autopilot Design with θ - D Technique," *Journal of Guidance, Control, and Dynamics*, **27**(3), pp. 406–417.

[21] Deprit, A. (1969) "Canonical Transformations Depending on a Small Parameter," *Celestial Mechanics and Dynamical Astronomy*, **1**, pp. 12–30.

[22] Sassano, M. and A. Astolfi (2012) "Dynamic Approximate Solutions of the HJ Inequality and of the HJB Equation for Input Affine Nonlinear Systems," *IEEE Transactions on Automatic Control*, **57**(10), pp. 2490 – 2503.

[23] Albrekht, E. G. (1962) "On the Optimal Stabilization of Nonlinear Systems," *Journal of Applied Mathematics and Mechanics*, **25**(5), pp. 1254–1266.

[24] Carrington, C. K. and J. L. Junkins (1986) "Optimal Nonlinear Feedback Control for Spacecraft Attitude Maneuvers," *Journal of Guidance Control and Dynamics*, **9**(1), pp. 99–107.

[25] Vadali, S. R. and R. Sharma (2006) "Optimal Finite-Time Feedback Controllers for Nonlinear Systems with Terminal Constraints," *Journal of Guidance, Control, and Dynamics*, **29**(4), pp. 921–928.

[26] Grüne, L. (1997) "An adaptive grid scheme for the discrete Hamilton-Jacobi-Bellman equation," *Numerische Mathematik*, **75**(3), pp. 319–337.

[27] Saridis, G. N. and C. S. Lee (1979) "An Approximation Theory of Optimcal Control for Trainable Manipulators," *IEEE Transactions Systems, Man and Cybernetics*, **SMC-9**(3), pp. 152–159.

[28] Beard, R. and T. McLain (1998) "Successive Galerkin Approximation Techniques for Nonlinear Optimcal and Robust Control," *International Journal of Control*, **71**(5), pp. 717–748.

[29] Boulbrachene, M. and M. Haiour (2001) "The finite element approximation of Hamilton-Jacobi-Bellman equations," *Computers & Mathematics with Applications*, **41**(7-8), pp. 993–1007.

[30] Abu-Khalaf, M. and F. L. Lewus (2005) "Nearly Optimal Control Laws for Nonlinear Systems with Saturating Actuators Using a Neural Network HJB Approach," *Automatica*, **41**(5), pp. 779–791.

[31] Osher, S. and R. Fedkiw (2003) *Level Set Methods and Dynamic Implict Surfaces*, Applied Mathematical Sciences.

[32] Adurthi, N., P. Singla, and M. Majji "Sparse Approximation based Collocation Scheme for Nonlinear Optimal Feedback Control Design," *AIAA Journal of Guidance, Control and Dynamics, In Review.*

[33] Mercurio, M., N. Adurthi, P. Singla, and M. Majji (2016) "A Collocation-Based Approach to Solve the Finite Horizon Hamilton-Jacobi-Bellman Equation," in *2016 American Control Conference.*

[34] Mercurio, M. and P. Singla (2016) "Conjugate Unscented Transform Based Approach to Solve the Fokker-Planck-Kolmogorov Equation," in *AIAA/AAS Astrodynamics Specialist Conference*, p. 5679.

[35] Mercurio, M. (2017) *Sparse Collocation Methods for Solving High Dimension PDES in Estimation and Control of Dynamical Systems*, Ph.D. thesis, State University of New York at Buffalo.

[36] Mercurio, M., P. Singla, and M. Majji (2018) "A Conjugate Unscented Transform-Based Scheme for Optimal Control with Terminal State Constraints," in *2018 Annual American Control Conference (ACC)*, IEEE, pp. 2651–2656.

[37] Mirzaei, M., P. Singla, and M. Majji (2018) "A sparse collocation approach for optimal feedback control for spacecraft attitude maneuvers," in *AAS/AIAA Astrodynamics Specialist Conference, 2017*, Univelt Inc., pp. 219–232.

[38] Adurthi, N., P. Singla, and T. Singh (2018) "Conjugate Unscented Transformation: Applications to Estimation and Control," *Journal of Dynamic Systems, Measurement, and Control*, **140**(3), p. 030907.

[39] Kelly, M. (2017) "An introduction to trajectory optimization: how to do your own direct collocation," *SIAM Review*, **59**(4), pp. 849–904.

[40] Grant, M., I. Clark, and R. Braun (2011) "Rapid simultaneous hypersonic aerodynamic and trajectory optimization using variational methods," in *AIAA Atmospheric Flight Mechanics Conference*, p. 6640.

[41] Betts, J. T. (1998) "Survey of Numerical Methods for Trajectory Optimization," *Journal of Guidance, Control and Dynamics*, **21**(2), pp. 193–207.

[42] Patterson, M. A. and A. V. Rao (2014) "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)*, **41**(1), p. 1.

[43] Rao, A. V. (2009) "A survey of numerical methods for optimal control," *Advances in the Astronautical Sciences*, **135**(1), pp. 497–528.

[44] Elsgolc, L. D. (2012) *Calculus of variations*, Courier Corporation.

[45] Petrov, I. P. (1968) *Variational methods in optimum control theory, Tech. rep.*

[46] Kraft, D. (1985) "On converting optimal control problems into nonlinear programming problems," in *Computational mathematical programming*, Springer, pp. 261–280.

[47] Renes, J. (1978) *On the use of splines and collocation in a trajectory optimization algorithm based on mathematical programming*, NLR.

[48] Betts, J. and I. Kolmanovsky (2002) "Practical methods for optimal control using nonlinear programming," *Applied Mechanics Reviews*, **55**, p. B68.

[49] Elnagar, G., M. A. Kazemi, and M. Razzaghi (1995) "The pseudospectral Legendre method for discretizing optimal control problems," *IEEE transactions on Automatic Control*, **40**(10), pp. 1793–1796.

[50] Fahroo, F. and I. M. Ross (2002) "Direct Trajectory Optimization by a Chebyshev Pseudospectral Method," *Journal of Guidance, Control and Dynamics*, **25**(1), pp. 160–166.

[51] Ross, I. M. and F. Fahroo (2003) "Legendre Pseudospectral Approximations of Optimal Control Problems," *Lecture Notes in Control and Information Sciences*, **295**.

[52] Guo, T., J. Li, H. Baoyin, and F. Jiang (2013) "Pseudospectral methods for trajectory optimization with interior point constraints: Verification and applications," *IEEE Transactions on Aerospace and Electronic Systems*, **49**(3), pp. 2005–2017.

[53] Elnagar, J., M. A. Kazemi, and M. Razzaghi (1995) "The Pseudospectral Legendre Method for Discretizing Optimal Control Problems," *IEEE Transactions on Automatic Control*, **40**(10), pp. 1793–1796.

[54] Williams, P. (2009) "Hermite-Legendre-Gauss-Lobatto direct transcription in trajectory optimization," *Journal of guidance, control, and dynamics*, **32**(4), pp. 1392–1395.

[55] Garg, D., M. A. Patterson, C. Francolin, C. L. Darby, G. T. Huntington, W. W. Hager, and A. V. Rao (2011) "Direct trajectory optimization and costate estimation of finite-horizon and infinite-horizon optimal control problems using a Radau pseudospectral method," *Computational Optimization and Applications*, **49**(2), pp. 335–358.

[56] Fahroo, F. and I. M. Ross (2002) "Direct trajectory optimization by a Chebyshev pseudospectral method," *Journal of Guidance, Control, and Dynamics*, **25**(1), pp. 160–166.

[57] Gill, P. E., W. Murray, M. A. Saunders, and E. Wong (2018) *User's Guide for SNOPT 7.7: Software for Large-Scale Nonlinear Programming*, *Center for Computational Mathematics Report CCoM 18-1*, Department of Mathematics, University of California, San Diego, La Jolla, CA.

[58] Ross, I. M. and F. Fahroo (2001) "A Pseudospectral Transformation of the Convectors of Optimal Control Systems," *IFAC Proceedings Volumes*, **34**(13), pp. 543–548.

[59] Davis, P. J. (1975) *Interpolation and Approximation*, Dover Publication, Inc NewYork.

[60] Beard, R. (1995) "Improving the closed-loop performance of nonlinear systems," *Rensselaer Polytechnic Institute*.

[61] Beard, R., G. Saridis, and J. Wen (1997) "Gelarkin Approximation of the Generalized Hamilton-Jacobi Bellman Equation," *Automatica*, **33**(12), pp. 1042–1048.

[62] Park, C. and P. Tsiotras (2003) "Sub-Optimal Feedback Control Using a Successive Wavelet-Galerkin Algorithm," in *Proceedings of the American Control Conference*, American Automatic Control Council, Dayton, OH, pp. 1926–1931.

[63] Abu-Khalaf, M. and F. L. Lewis (2004) "Nearly Optimal State Feedback Control of Constrained Nonlinear Systems Using a Neural Networks HJB Approach," *IFAC Annual Reviews in Control*, **28**, pp. 239–251.

[64] Osher, S. and J. A. Sethian (1988) "Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations," *Journal of computational physics*, **79**(1), pp. 12–49.

[65] Juang, J.-N., J. D. Turner, and H. M. Chun (1985) "Closed-form solutions for feedback control with terminal constraints," *Journal of Guidance, Control, and Dynamics*, **8**(1), pp. 39–43.

[66] Betts, J. T. (2010) *Practical methods for optimal control and estimation using nonlinear programming*, vol. 19, Siam.

[67] Hargraves, C. R. and S. W. Paris (1987) "Direct trajectory optimization using nonlinear programming and collocation," *Journal of Guidance, Control, and Dynamics*, **10**(4), pp. 338–342.

[68] Elnagar, G. N. and M. A. Kazemi (1998) "Pseudospectral Chebyshev Optimal Control of Constrained Nonlinear Dynamical Systems," *Computational Optimization and Applications*, **11**, pp. 195–217.

[69] Williams, P. (2004) "Jacobi Pseudospectral Method for Solving Optimal Control Problems," *Journal of Guidance, Control, and Dynamics*, **27**(2), pp. 293–297.

[70] Darby, C. L., W. W. Hager, and A. V. Rao (2011) "Direct Trajectory Optimization Using a Variable Low-Order Adaptive Pseudospectral Method," *Journal of Spacecraft and Rockets*, **48**(3), pp. 433–445.

[71] Darby, C. L., D. Garg, and A. V. Rao (2011) "Costate Estimation Using Multiple-Interval Pseudospectral Methods," *Journa of Spacecraft and Rockets*, **48**(5), pp. 856–866.

[72] Ross, I. M. (2015) *A primer on Pontryagin's principle in optimal control*, vol. 2, Collegiate publishers San Francisco, CA.

[73] Saridis, G. N. and C.-S. G. Lee (1979) "An approximation theory of optimal control for trainable manipulators," *IEEE Transactions on systems, Man, and Cybernetics*, **9**(3), pp. 152–159.

[74] Golub, G. H. and C. F. Van Loan (1996) *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, MD, pp. 140–149, 601.

[75] Adurthi, N., P. Singla, and T. Singh (2012) "The conjugate unscented transform—an approach to evaluate multi-dimensional expectation integrals," in *American Control Conference (ACC), 2012*, IEEE, pp. 5556–5561.

[76] Adurthi, N. (2016) *Conjugate Unscented Transform Based Methods for Uncertainty Quantification, Nonlinear Filtering, Optimal Feedback Control and Dynamic Sensing*, Ph.D. thesis, State University of New York at Buffalo.

[77] ——— (2013) *The Conjugate Unscented Transform - A Method to Evaluate Multidimensional Expectation Integrals*, Master's thesis, University at Buffalo.

[78] Adurthi, N., P. Singla, and T. Singh (2012) "The Conjugate Unscented Transform-An Approach to Evaluate Multi-Dimensional Expectation Integrals," Proceedings of the American Control Conference.

[79] Nagavenkat, A., P. Singla, and T. Singh (2013) "Conjugate unscented transform rules for uniform probability density functions," in *Proceedings of the American Control Conference*.

[80] Boyd, S. and L. Vandenberghe (2004, 67-78, 152-159) *Convex Optimization*, Cambridge University Press.

[81] Kirk, D. E. (2012) *Optimal control theory: an introduction*, Courier Corporation.