An Iterative LQR Method for Addressing Model Uncertainty in the Mars Entry Problem

by

Allegra Farrar

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

Masters of Aeronautics and Astronautics

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2022

Author . . . .
Department of Aeronautics and Astronautics
May 17, 2022

Certified by . . . .
Richard Linares
Associate Professor
Thesis Supervisor

Accepted by . . . .
Jonathan How
Chairman, Department Committee on Graduate Theses

An Iterative LQR Method for Addressing Model Uncertainty in the Mars Entry Problem

by

Allegra Farrar

Submitted to the Department of Aeronautics and Astronautics
on May 17, 2022, in partial fulfillment of the
requirements for the degree of
Masters of Aeronautics and Astronautics

Abstract

Since the moon landing in 1969 sounded the proverbial shotgun inciting efforts to expand the frontiers of space exploration, there has been unparalleled effort to enhance the technologies for doing so. While human presence in Earth orbit has boomed over the past decade, crewed planetary missions have yet to reach desired goals. Set by NASA as the future destinations beyond Earth orbit, Mars presents significant challenges to the entry, landing and descent sequence. Missions including sample return and human exploration require precise landing accuracy. Additionally, entry vehicle dynamics and atmospheric parameters at time of flight are hard to predict. This along with the advancement of mission objectives invoke the need for a reliable, robust, and computationally reasonable method with certifiable guarantees of safe landing.

Therefore, this paper presents a closed-loop trajectory optimizer capable of incorporating the atmospheric models and navigational data uncertainty for the nonlinear dynamics of hypersonic entry by applying an iterative Linear-Quadradic-Regulator (iLQR). iLQR is an efficient and powerful method for trajectory optimization derived from Differential Dynamic Programming (DDP) principles, which have been applied successfully in cases of robotic movement to locally improve upon a single trajectory through second-order convergence for a local optimal trajectory. iLQR takes this method a step further by iteratively linearizing the system dynamics, converging to determine an optimal trajectory by minimizing the performance cost and the uncertainty in the dynamics model.

To demonstrate its effectiveness, the algorithm will be tested against a series of realistic simulations to test the model performance against mission requirements, such as high altitude and precision landing. Results show an efficient data-driven algorithm capable of learning how to successfully control a 40 ton crewed-scale spacecraft for Mars entry under dynamical uncertainties in the state model. Additionally, given system performance parameters, the covariance, or landing accuracy, of the final position can be determined from the algorithm and the results can be used to determine safe parameter ranges that achieve the desired accuracy.

Thesis Supervisor: Richard Linares
Title: Associate Professor

Acknowledgments

I would like to sincerely thank the support system that made it possible to complete the presented work. First and foremost I would like to thank the Department of Aeronautics and Astronautics at MIT for providing the platform to complete this research. I would also like to thank members of the Astrodynamics, Robotics, and Controls Lab (ARCLab) and Space Systems Laboratory (SSL) for their support and guidance over the last couple of years.

Specifically, I would like to give acknowledgement to my thesis advisor, Dr. Richard Linares for his support of this work and guidance throughout this process. I would also like to acknowledge Dr. Bryce Doerr and Dr. Keenan Albee for their guidance in formulating many of the topics presented in this thesis.

I would also like to thank those who reviewed this work and supported it's development at various stages. Your diligence has not gone unnoticed.

This work is supported by the National Science Foundation

I would like to dedicate the completion of this work to my family whose support continues to propel me to new heights ... we're going to reach the stars together!

This Master's thesis has been examined by a Committee of the Department of Aeronautics and Astronautics as follows:

Jonathan P. How . . . .
R. C. Maclaurin Professor of Aeronautics and Astronautics
Chair, Graduate Program Committee

Richard Linares. . . . .
Boeing Career Development Professor of Aeronautics and Astronautics
Thesis Advisor

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

The future of advancing missions to Mars will require 1) greater precision landing accuracy, 2) higher elevation landing ability (+ 1 - 4 km MOLA[1]), and 3) larger entry mass capability $\geq$ 40 mT [6, 9]. While achieving these objectives alone is a challenging task, the inherent limitations to Mars entry, descent, and landing (EDL) apply additional strain on achieving goals for future exploration.

Detailed in Fig. 1-1, entry is the primary and most critical stage of the EDL sequence. Entry flight takes place once the spacecraft interfaces with the planetary atmosphere, taken as approximately 125 km above the Martian surface for this application. The main challenge lying at the foundation of any atmospheric entry problem is managing *hypersonic flight*, typically categorized by the vehicle achieving Mach numbers greater than 5. During hypersonic flight, the aerodynamics and gas dynamics exhibit physical flow phenomena that generate concerns for acceleration, pressure, and heating levels of the vehicle,[2] increasing as the Mach number increases. These concerns based on atmospheric conditions establish inherent limitations to the entry phase, particularly for Mars.

[1] MOLA stands for Mars Orbiter Laser Altimeter, an instrument used to measure Mars topography in 2001. This is used to indicate surface elevation with respect to a nominal reference defined by the topography. Approximate range taken from Ref. [6] and [29].

[2] See Ref. [1] for a more detailed explanation on hypersonic flow

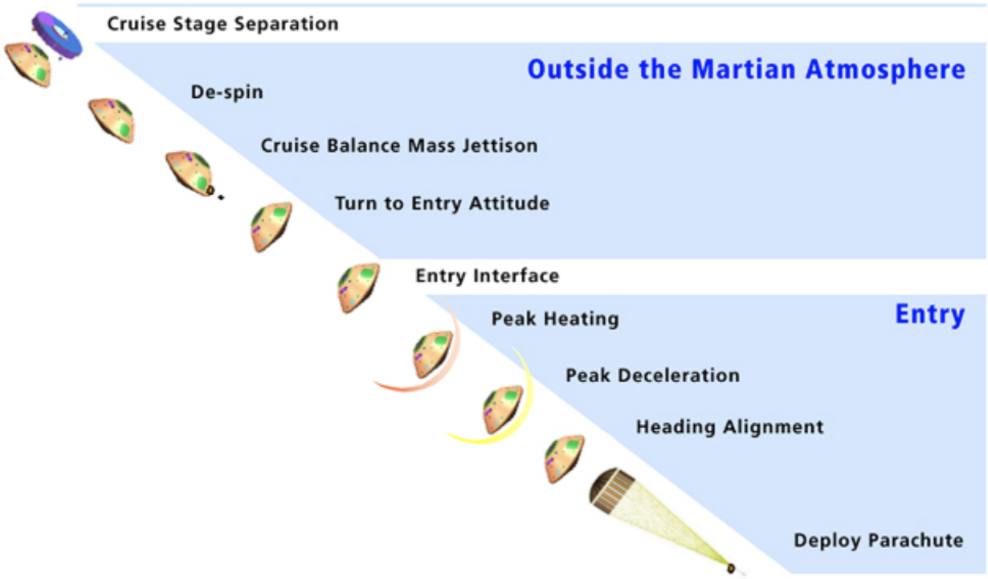

Figure 1-1: Transition and Entry Phases for MSL EDL [29][3]

The Martian atmosphere is known for its considerably low density, with approximately 1/100th of Earth's atmospheric thickness [6], causing vehicle deceleration to occur at critically low altitudes. This presents concerns for having enough altitude above the surface to complete the following descent and landing stages of the EDL sequence without crashing and terminating the mission. For context, the Mars System Laboratory (MSL) mission, which landed in 2011 and was one of the largest scale missions to date, saw an entry Mach of 24 [46]. This vehicle had to be slowed to Mach 2 prior to initiating the descent phase of the EDL sequence using parachute deployment. Standard Newtonian physics indicate that with increased entry mass, this entry speed will be larger and more difficult to slow at an altitude above the critical 10 km region. However, while the atmosphere on Mars is thin in comparison to the well-charted Earth atmosphere, it is still thick enough to cause concerns for the heating and pressure levels seen during entry flight [6]. This limitation becomes significantly more concerning when considering a crewed flight.

Additionally, the mountainous Martian terrain establishes landing limitations on the mission. Previous Viking and Mars Exploration Rover (MER) missions landed

[3] Full sequence shown in Appendix A, Fig. A-1, A-2, and A-3

at low elevation sites no greater than -1.3 MOLA [6]. MSL was the first mission to demonstrate a high elevation landing, followed by Mars 2020 [32]. For advanced scientific missions expected to explore other locations on the surface and considering the sensitivity of safely landing a crewed mission, future technology must demonstrate higher elevation landing capability of at least +1 MOLA given the terrain of targeted landing locations. This compounded with the thin atmosphere condition advances the constraints of the problem.

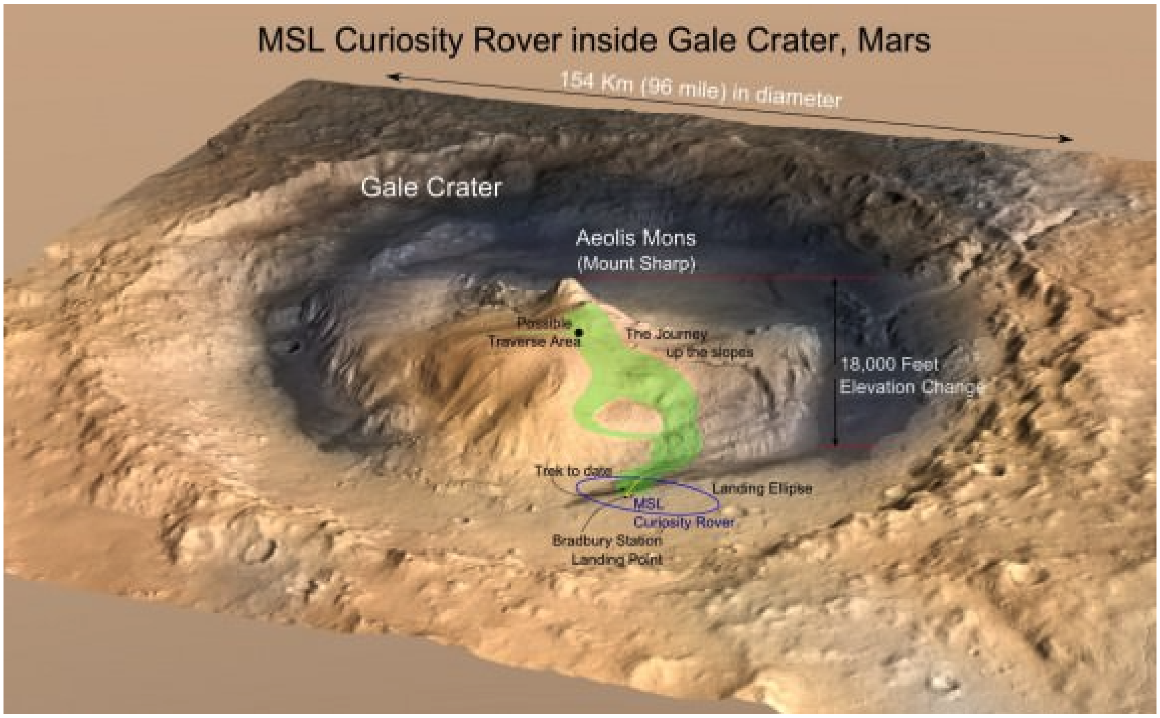

Figure 1-2: 3D Representation of MSL Landing Site[4]

Further inhibiting the goals for successful entry are the uncertainties impacting the guidance and control. These uncertainties stem from potential wind and dust storms threatening to cause severe deviations in trajectory from a nominal sequence, as well as uncertain aerodynamic coefficients and atmospheric modeling within the guidance and control system [24]. Coinciding with these uncertain elements, simulating hypersonic flight conditions for ground-based testing is an extremely challenging task.

[4] MRO image of Gale Crater. Credits: NASA/JPL, illustration, T.Reyes. Source: https://www.universetoday.com/138424/curiosity-lasted-2000-days-mars-triple-original-mission-plan/

The characteristics of hypersonic flow are dependent on how fast a vehicle is traveling within a given flow medium based on Mach number, which makes it impossible to duplicate all aspects of the flight environment during one test. Limitations based on high enthalpy[5] levels, accurate air chemistry and test times on the order of projected flight times have required test facility designs based on targeting certain knowledge points than the entire breadth of the flight envelope.[6] This limitation adds to the gap of accurate modeling and computation of vehicle conditions such as coefficients of lift and drag. Furthermore, simplifications such as the linearization and discretization of equations of motion made as part of the mathematical model create deviations from the true system causing added uncertainty [24].[7]

The competing objectives of properly managing the problem uncertainty while adhering to the necessary constraints for successful future missions lies at the center of this work's motivation for developing a robust guidance system capable of addressing these challenges.

1.1.1 Guidance Methods

The Mars entry guidance problem is that of motion planning in that the goal is to get the spacecraft from initial atmospheric entry to proper terminal conditions for initiating the next phases of the EDL process. Given spacecraft communication during entry is lagged beyond the ability of human intervention and conditions are severe enough to render a failed mission if guidance is not executed properly, the chosen method to get the spacecraft from planetary capture to arrival at the target landing site becomes as critical as the mission itself. As a result, guidance methods take on the approach of aerodynamically maneuvering the entry vehicle to achieve optimal flight conditions, or in other words, optimal control. To instruct the execution

[5] Enthalpy is a thermodynamic quantity representing the heat content of a system based on chemical processes

[6] NASA Ames Arc jet facility typically used to test thermal protection systems (TPS) for entry vehicles and NASA Langley Research Center wind tunnels are capable of reaching hypersonic conditions, but not on the order of the Mars entry interface. See Ref. [1] for more details on hypersonic testing and facilities.

[7] Discussed in greater detail in Ch. 2

of in-flight control, trajectory planning can be done through [18, 44]:

1. *Reference tracking* where nominal trajectory conditions are planned offline and the on-board controller attempts to retain the nominal path by correcting trajectory deviations.

2. *Path planning* where the trajectory is computed online from the current state to the desired target state or state range by applying a predictive model

Prior to MSL, Viking and subsequent lander missions (1997 Mars Pathfinder, 2004 Mars Exploration Rovers and 2008 Pheonix) conducted unguided ballistic trajectories with zero lift [29]. As part of demonstrating improvements in landing accuracy, higher elevation landing, and other advancements in mission capabilities, MSL was the first mission to utilize bank angle controlled entry guidance for reference tracking inherited from Apollo mission designs [7, 32].

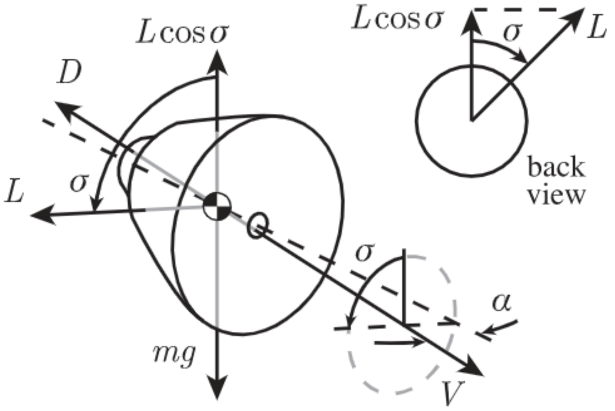

Figure 1-3: Bank Angle Control Forces [37]

As shown in Fig. 1-3, control input commands through bank angle orient the vehicle around the velocity vector to control the cross range of the entry trajectory. For bank angle control, the goal is to manipulate the bank angle to guide the vehicle along a trajectory terminating at conditions suitable for descent and landing.

These methods have allowed for Mars mission success to-date; however, building up to a crewed-scale mission, or even more elaborate robotic and scientific missions, will undoubtedly require advancements beyond the current state-of-the-art. Thankfully, advances since the dawn of the space age in fields such as robotics, control theory and machine learning have allowed for optimal control techniques with capabilities to address critical needs for Mars entry, and planetary entry at large.

1.1.2 Robust Entry Guidance

Since the late 1950s, optimal control theory has been explored and developed with great prominence [22]. Starting with the robotics field, many techniques have filtered through to aerospace operations such as hypersonic aircraft, reusable rocket recovery, satellite constellations, and more. Optimal control has become particularly beneficial to advancing technological capabilities of mechanical systems due to its ability for handling robust operation.

Robust optimal control approaches to entry guidance have been studies extensively with the dawn of NASA's Artemis program and ambitions for sending humans to Mars [6, 35]. The idea is to generate a guidance and control solution adaptable enough to handle multiple sources of uncertainty while still ensuring guaranteed success in completing mission objectives. For classification, this topic is represented in the form of *open-loop* and *closed-loop*.

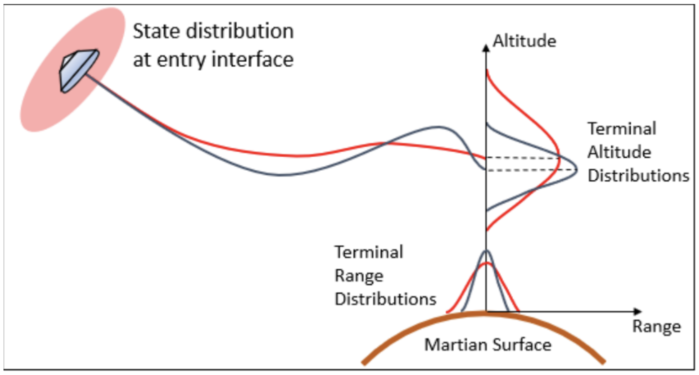

Figure 1-4: Nominal Deviation Sources for Entry Guidance taken from [10]

In the previous section, it was noted that trajectory planning for entry could be classified by reference tracking (offline planning) and path planning (online planning). Open-loop solutions for entry guidance typically fall under the category of reference tracking where the developed nominal trajectory is based solely on initial conditions and optimized around a cost function. As part of making these solutions robust against uncertainty, constraints on flight conditions and terminal bounds can be applied to further optimize the nominal path. Closed-loop solutions, however, can be applied to both reference tracking and path planning methods and are more widely used for optimal control of complex systems given their adaptability. Also classified as dynamic programming, closed-loop optimization apply stability and convergence guarantees necessary for highly uncertain systems. These solutions also allow for building on the robustness of the model with probabilistic constraints [10]. It has been noted in literature [18] that closed-loop guidance will be required to achieve landing accuracy within ± 10 km of the target for advanced robotic and crewed missions. For EDL analysis and design, pre-flight identification of potential deviations from expected flight conditions becomes essential to verifying mission success. Aside from making systems adaptable through closed-loop guidance, additional computational measures of developing an expected trajectory envelope provide performance guarantees necessary for the nature of the problem. This concept is known as *reachable sets*.

Reachable sets establish bounds and provide verification on terminal entry conditions including the landing uncertainty ellipse. Additionally, these sets help predict potential real-time performance amid potential error to the modeled nominal conditions. To determine these sets for error analysis, *uncertainty quantification* (UQ) methods can be applied.

There are several methods to performing UQ. Among these methods, there is a trade-off between computational expense and quantification accuracy [10]. Particularly, linear covariance lies on the more efficient side of the spectrum; however, its performance scales inversely with the nonlinearity of the system dynamics. Alternatively, Monte Carlo (MC) simulation is known for its precise computation of uncertainty bounds, but is computationally taxing to perform. Other methods have been developed to strike balance between the two extremes [26, 14, 10].

Numerous work has been presented in literature to address robust entry guidance for Mars entry. Lu et al. [24] presents a robust, closed-lop control approach by tracking a reference drag versus velocity profile, similar to space shuttle guidance, where both angle of attack and bank angle are controlled. Similarly, Cianciolo et al. [9] performs dual direction control by modulating the angle of attack and bank angle and applying the control method to a crewed-scale mission design. However, this method only targets reducing errors in the open-loop formulation. Kluever [18], however, investigates closed-loop guidance in comparison to developing open-loop nominal trajectory solutions. In doing so they propose an adjustable altitude profile for online predictive guidance based on MSL design. Ridderhof [37] also targets a closed-loop guidance method for addressing external disturbances and model uncertainties by implementing a stochastic-based control approach for an MSL-like entry scenario. Work also has been done to improve reference trajectory planning through uncertainty modeling for analysis on large-scale, crewed mission design potential [10, 15, 29].

1.2 Contributions of Thesis

Adequately addressing the challenges of entry, particularly that of Mars, involves methods of both optimization and uncertainty analysis to construct a guidance method robust enough to ensure a safe and successful landing.

While several approaches specialize in one of the major domains in Fig. 1-5 with even some cross-disciplinary work, few approaches straddle all three of these domains to express operational guarantees for more extreme entry cases with high-mass and precision landing needs while respecting the strict constraints of the problem. Majority of the proposed optimal control approaches focus on enhancing flight capabilities for crewed design in the open-loop, reference trajectory formulation or seek to demonstrate adaptability to uncertainty and potential disturbances by applying closed-loop techniques to previously flown mission deigns. To the author's knowledge, demonstration of an open- and closed-loop optimal guidance method to generate a reference trajectory with adaptive capabilities has yet to be shown with a large-scale crewed mission design.

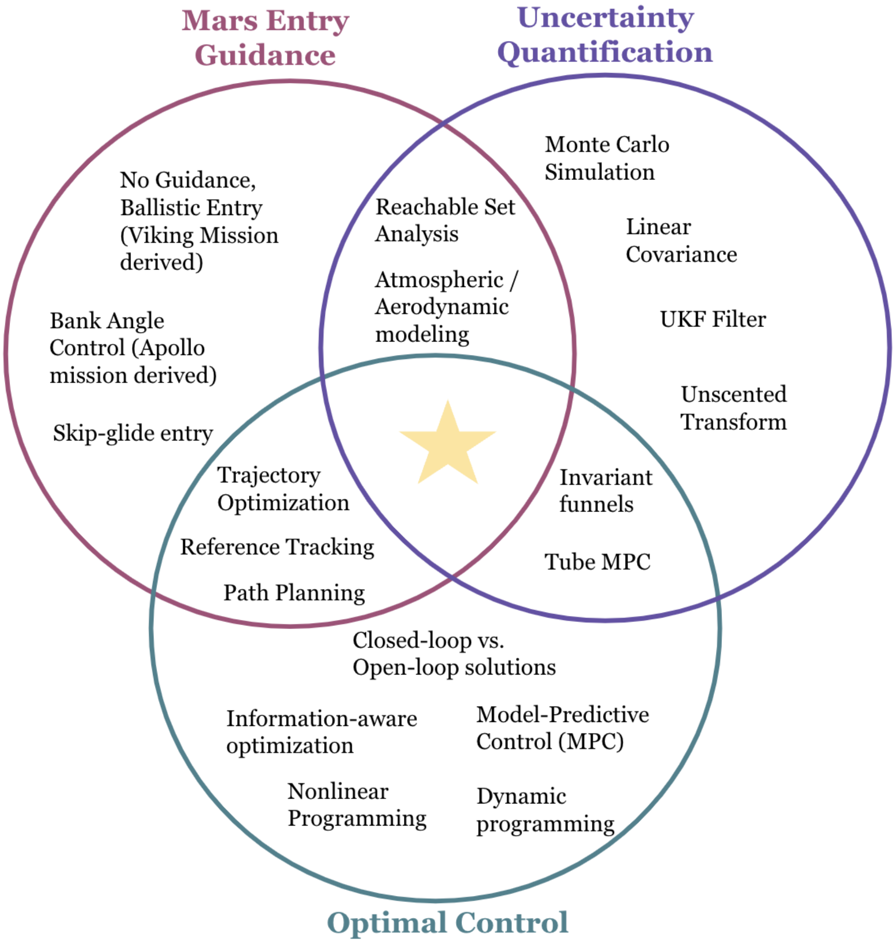

Figure 1-5: Outline of Topics Related to Thesis Work

Inspired by the computational structure in [12] and the approach outlined in [39], this approach performs two-tiered reference trajectory optimization to improve on the uncertainty bound range at terminal conditions and therefore, demonstrate greater capabilities in precision landing amid strict state and control constraints and highly nonlinear system dynamics. The goal is to present a computational infrastructure capable of supporting an autonomous, decision-making spacecraft operating large-scale, critical missions in the dynamic Mars entry environment.

1.3 Thesis Organization

Ch. 2 will start by backtracking through some of the major topics addressed in this introduction to provide an overview of considerations for motion planning, trajectory optimization and control theory.

Ch. 3 provides a more detailed look at optimal control methods and explores information-aware optimization as a form of robust optimal control by reviewing methods pertinent to the approach for this work.

Ch. 4 breaks down the methodology adopted for the presented work detailing the necessary equations, system modeling, and computation considerations.

Ch. 5 presents the generated results outlining performance capabilities with the applied method along with supporting discussion.

Ch. 6 concludes by reflecting on the impact of the presented work and important takeaways. This chapter also outlines considerations for extending this work to future applications

Chapter 2

Literature Review

2.1 Planning

Planning is an essential process to nearly all aspects of life. From everyday scheduling tasks to more complicated jobs such as organizing air traffic through a major city and even determining the trajectory of a spacecraft (which, ironically, is what this thesis covers). The more complex the task, the more involved the planning must be. However, what happens when plans must be completed and executed without human intervention? While not a concern for most planning needs, this is a question widely explored by the robotics field.

Many robotic applications investigate methods for developing automated mechanical systems with sensing, operational, and computational capabilities [20]. In other words, autonomous systems. This form of planning involves converting a high-level request (i.e. take the garbage to the trash) to low-level motion instructions that can be computationally implemented (i.e, move to trash location, pick up trash, carry trash, move to trash can, etc.) [20]. In doing so, this provides opportunity to extend solving problems beyond human capability through developing comprehensive and efficient algorithms for robust implementation [20, 41].

This chapter will start by giving an overview of topics addressed in this thesis (motion planning, trajectory optimization, and control theory) and then go on to address pertinent work related to these subjects.

2.2 Motion Planning

As mentioned previously, planning comes in all forms and is especially important when involving motion. In most cases, our brains exist as an essential component to determining basic movement strategy. To navigate from home to work, or even from the kitchen to the bathroom, there must be some decision process to get from point A to point B. While supporting equipment may aid in the completion of the task (i.e., car, feet, GPS navigation, eye sight, etc.), there must be some direct human intervention to determine the path and the method of execution. Therefore achieving autonomous movement of a mechanical system (robot, vehicle, spacecraft, etc.) requires establishing some internal guidance for moving from one position to the next based on its capabilities. This need promotes the discipline of motion planning.

The motion planning process can be best understood by looking at a basic grid-world scenario (see Fig. 2-1) explored widely in the robotics field.

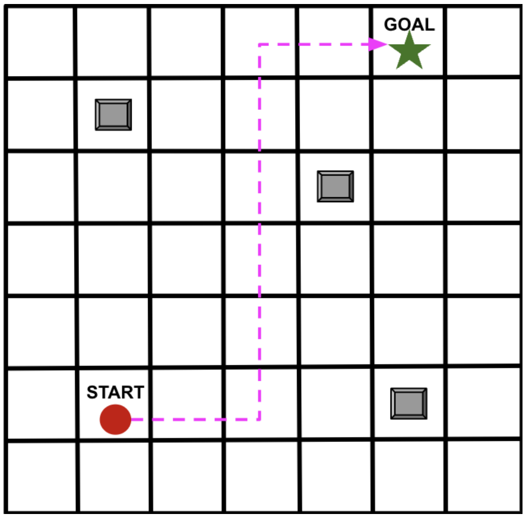

Figure 2-1: Grid World Motion Planning Scenario

While the basic goal is to progress from the start to goal location, based on the developed algorithm, motion planning allows mechanical systems to handle environments with varying levels of uncertainty, obstacle avoidance, unknown initial conditions, disturbances, and more, without requiring direction from an external operator. Subsequently, proper motion planning is an essential piece to developing autonomous function.

As discussed by LaValle [20], There are five key ingredients to setting up any motion planning problem:

1. **State space**: A state space captures all the possible scenarios of the agent. When considering a robot operating in a simple grid world, for example, the state space would include all the available grid locations. However, if the system allowed for information on the robots location, where it's facing and how fast it's going, the state space would be expanded to include all available positions, orientations and velocities. As a result, state spaces can be discrete (finite or countably infinite) or continuous (uncountably infinite).

2. **Time**: Time is used to guide all motion planning decisions. This can be done *explicitly* by defining the system in terms of time or *implicitly* by matching time the the succession of the motion sequence.

3. **Action space**: The action space sets the actions to alter the state. In subjects of control theory and robotics, the terms *inputs* and *controls* are used to identify the actions and operator that executes the plans, respectively. In this case, a *controller* or *control system* is what drives the changes in state of the trajectory and a *control sequence* is a specified set of inputs that produce a given trajectory. These terms will be addressed throughout the thesis as optimal control is covered in greater depth. Additionally, methods of selecting actions include random sampling to build a value function or selecting from nature.

4. **Initial and Goal State**: Planning problems typically involve navigating from a set initial state to a specified goal state or goal state within a set of desired goals.

5. **Criteria**: The criteria of a motion planning problem involves some objective to guide the motion form the initial to goal state. This criteria includes 1) *feasibility* - is it possible to find a plan, and 2) *optimality* within the feasible plans - is there a plan that improves the performance in a defined way.

The following sections of this chapter will review motion planning as it relates to trajectory optimization and control theory.

2.3 Trajectory Optimization

Trajectory optimization is a powerful motion planning tool applied to goal-oriented robotic motion [40] with a focus on optimal solutions for nonlinear and continuous *dynamical systems*. Seeking to to find a state-control sequence that reduces an applied cost function, this type of optimization can be considered a form of finite-horizon optimal control that focuses on problems with high dimensional systems, large state spaces, or the need to be very accurate [41, 17]. These problems typically include systems with many degrees of freedom (DOF) (i.e., robotics and vehicle control) invoking highly nonlinear dynamics and therefore, requiring stronger computational tools to address the complexity of the system while utilizing a motion planning algorithm to chart a path from an initial point to final states.

Unlike optimization in the parameterized world (such as the grid-world environment discussed in the previous section), trajectory optimization focuses on minimizing the cost function, J, for an arbitrary vector function, $\mathbf{f}(x(t), u(t))$. This significantly expands the state space making these type of problems more challenging to solve [17]. This thesis focuses on continuous dynamical systems where the time, state space, and control is continuous. This form is commonly seen in robotics and aerospace applications.

To better understand the set-up of these types of problems, we can define them in terms of the five key components to motion planning.

1. **State Space ($\mathbf{x}$)**: Defined by a system of differential equations, $\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u})$ known as the *dynamics*, to implicitly represent the continuous state space.

2. **Control Sequence ($\mathbf{U}$)**: Set of control inputs to the system dynamics. The optimal control sequence is the set of inputs, $\mathbf{U}^*(\mathbf{x})$, that minimizes the total cost.[1]

[1] This is what allows trajectory optimization solutions to address optimal control problems. More

3. **Goal**: Guiding from initial state to goal state or goal within set of goals states.

4. **Optimization**: Minimize the cost provided by a specified function $J\left(x, u\right)$.

These systems can be discretized to form an approximation to the continuous-time form

$$\mathbf{x}_{i+1} = f(\mathbf{x}_i, \mathbf{u}_i)$$

to expand on modeling and implementation capabilities. The above formulation models propagating the state space of the dynamics forward in time through incremental time steps.

Basic trajectory optimization tends to be an open-loop planning system producing a *locally optimal* solution in which there are no convergence guarantees for every formulation of the problem. This becomes a concern for situations in which a solution must be found given the provided input and problem constraints. This is what *dynamic programming* addresses.

Dynamic Programming

Unlike standard optimization techniques, dynamic programming focuses on developing an *optimal policy*. Rather than computing the control for a single optimal trajectory, this technique computes the optimal control for *every* point in the state space for a more considerable approach to determining an optimal trajectory (see Fig. 2-2). This process is what drives closed-loop solutions for trajectory optimization.

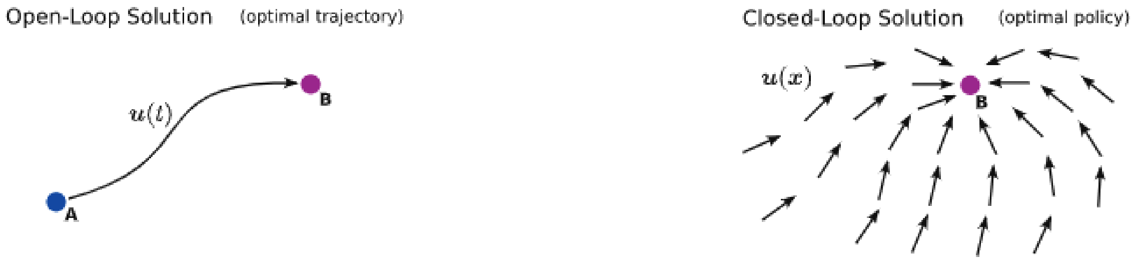

Figure 2-2: Open-loop vs. Closed-loop Trajectory Optimization Solutions [17]

on this in Sec. 2.4

In part of developing formulations for locating an optimal trajectory, these methods can also be applied to design control input.

2.4 Control Theory

Control theory is the study of designing inputs to physical systems defined by differential equations. Its applications are expansive and include mechanical systems (cars, aircraft, automatic machinery, etc.), electrical systems (circulation pumps, noise filters, compressors, etc.), and even reaching to fields of chemistry and sociology [20]. Typical representations include *feedback* design which enables a controller to build its performance off of the system response, otherwise known as a *feedback controller.*

Feedback Control

Aligned with dynamic programming, feedback controllers use some form of sensor capability onboard the mechanical system to measure the system output and compute the control input. See Fig. 2-3 for standard block diagram.

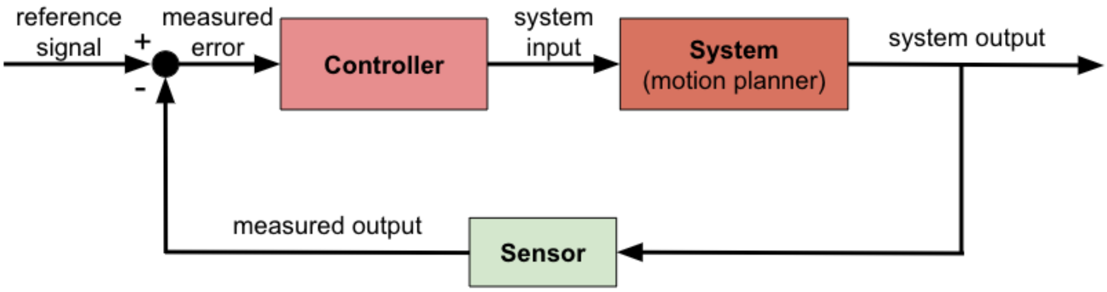

Figure 2-3: Feedback Controller Structure

While feedback controllers are challenging to implement and adapt to complex problem formulations, they provide greater stability to the overall control design by executing based on system response. However while robust to disturbances, standard control design does not implement optimality goals for strategically selecting the control input to improve system response. Combining goals of trajectory optimization and control theory leads directly to the discipline of optimal control.

Chapter 3

Optimal Control

The limitations of space, and the human navigation thereof, leads to problems requiring constrained optimization solutions [5]. These solutions typically generate a *nonlinear program* problem where either the constraints or the objective function are nonlinear. Optimal control solutions can then be applied to extend nonlinear programming (NLP) to an infinite number of variables [5] for solutions with mandatory, verifiable, and sufficient [25].

The overall goal of optimal control is to develop a controller capable of taking a system from some initial state to some desired final state while also providing the best possible performance based on some targeted *measure of performance*, defined by a cost function, J, for this work. Most problem involving motion planning and control theory, especially for robotics applications, tend to focus on the optimal criteria.

Trajectory optimization takes part in control theory through developing optimal control solutions. *The dynamics*, modeled as a set of differential equations for dynamical systems, determine how the system changes in response to applied control. Therefore, for optimal control solutions, trajectory optimization not only optimizes the plan, but also the control input which informs the optimal plan.

Additionally, taking trajectory optimization from a planning method to an online control method introduces the domain of Model-Predictive Control (MPC), known for its dominant capabilities in developing real-time optimal control solutions. MPC has the capability of working with complex problems while avoiding the computational issues known to dynamic programming for problems with large state spaces.

3.1 Information-Aware Planning and Control

As part of NLP and optimal control, information-aware planning considers trajectory optimization of the closed-loop form or dynamic programming.[1] This topic includes machine learning techniques from artificial intelligence (such as reinforcement learning) which improve on a problem or scenario through building a data knowledge base, as well as Markov decision processes (MDPs) for partially observable states (POMPDPs).[2] To expand on the topics surrounding this thesis, we will start by reviewing methods for information-aware planning and their trade-offs in computation and performance.

3.1.1 Direct vs. Indirect Methods

Methods for solving trajectory optimization problems can mostly be categorized as *direct* or *indirect*. While direct methods discretize the entire continuous-time trajectory optimization problem (dynamics, constraints, etc.) to convert to a NLP prior to optimization, indirect methods find the optimal solution first, then discretize the resulting conditions [17]. As with any computational technique, there are trade-offs related to the optimization goals. Indirect methods stand out mainly for their accuracy and reliable error estimate [17]. However, in comparison to direct methods, indirect methods have reduced convergence guarantees and require more precise initial guesses [17].

As an extension to indirect and direct trajectory optimization, shooting methods simulate an approximate trajectory to optimize around [17]. This form tends to work best for problems where dynamics must be calculated accurately but the control space is fairly simple [5]. This applies directly to trajectory optimization for entry spacecraft given the complex, nonlinear dynamics yet low order of control inputs (i.e.,

[1] Discussed at length in Sec. 2.3 of this thesis

[2] See Ref. [12] for more information on information-aware planning techniques

only firing thruster for control execution) [17].[3]

3.1.2 Differential Dynamics Programming

Differential Dynamic Programming (DDP) is a form of direct shooting that computes the optimal trajectory through backwards propagation of the optimal control [17]. Due to computation of 2^{nd}-order and cross partial derivatives, NLP's tend to grow quadratically with the number of decision variables creating a computationally expensive solution method [3]. In response, DDP has demonstrated strong quadratic convergence abilities compared to standard Newton's method[4] [40] allowing for a NLP method capable of operating on the exact value of the state as opposed to a linear approximation [31].

For this method, dynamic constraints do not need to be specified given the dynamics model is integrated within the optimization. However, for these formulations control is inherently unconstrained [40]. While this is beneficial to allowing for real-time, fast computation (given that the controller has full range of the search space for devising an optimal solution), it presents an issue for problems where the control can only operate within specified bounds. Tassa's approach in Ref. [40] seeks to expand on this limitation by applying control boundary constraints expanding the application to problems where the control is limited in kinematics such as steering a vehicle or joint rotation of a humanoid robot.

3.1.3 Iterative Linear-Quadratic Regulator

As part of the family of DDP principles, iterative Linear-Quadratic Regulator is a Gauss-Newton approximation that takes the optimization a step further by only requiring computation of the first-order derivatives as opposed to classic DDP methods that require up to second-order derivatives [40, 39]. This allows the computational cost of this optimal controller to be more efficient and applicable to systems with nonlinear dynamics models (as with the model for this application).

[3] See Ref. [17] and [5] for more information on trajectory optimization methods.

[4] See Ref. [25] and [5] for a more in-depth description on Newton's method

iLQR bases its operation off the Linear-Quadratic Regulator (LQR); a respected concept in optimal control with far-reaching application for trajectory optimization and widely used in robotics. LQR functions as a common feedback controller that operates based on two major assumptions:

1. Known, linear system dynamics expressed in the form:

   $$\dot{\mathbf{x}} = \mathrm{A}\mathbf{x} + \mathrm{B}\mathbf{u}$$

   Where $\mathbf{x}$ and $\dot{\mathbf{x}}$ are the state and state time derivative, $\mathbf{u}$ is the control input, and **A** and **B** are matrix coefficients that contain the effects of state and control input on the derivative of the state.

2. Quadratic cost function of the form:

   $$J = \int_0^\infty \left(\mathbf{x}^T \mathrm{Q}\mathbf{x} + \mathbf{u}^T \mathrm{R}\mathbf{u}\right) dt$$

   Where Q and R are weights applied to the cost function (penalties) for note being at the target state while applying some control signal. A high Q value prioritizes getting to the target state ASAP, and a higher R prioritizes a small control input for getting to the target state.

LQR plans an optimal trajectory sequence (for both state and control) where controller 1) linearizes the dynamics, 2) evaluates the cost function around a current point in space, and 3) computes the feedback gain. This drives the system from an initial state to a target state based on the system model and cost function. However, the assumption that the dynamics are linear and consistent for all states causes LQR to optimize assuming the approximated linear dynamics computed at the current time hold for all time [16].

To avoid this locally linear assumption That causes LQR solutions to only be valid at the current point in time when the system dynamics are nonlinear, iterative LQR (iLQR) expands on the LQR method to optimize at each time step by iteratively reusing LQR to refine the trajectory and eventually, converge to an optimal solution [38, 39, 41].

Each iteration of the iLQR controller can be categorized by the following three steps:

1. **Dynamics Rollout and Derivatives**: To establish a starting reference trajectory, a dynamics rollout is completed by integrating over the system of equations. For this work, Runge-Kutta integration is applied to obtain this trajectory starting from an initial state x_0. Then, with the reference trajectory, $\{\mathbf{x},\ \mathbf{u}\}$, the first-order derivatives for the state space and cost function are computed.

2. **Backward Pass**: From the reference trajectory, starting from the last state and iterating backwards, the controller determines the optimal control sequence for this trajectory. This is performed continuously until a successful sequence is found.[5]

3. **Forward Pass**: The control sequence from the backward pass is then propagated forward in time to develop the optimal trajectory.

By running this sequence over an iteration and continuously recomputing the dynamics, the solution is simultaneously refined allowing for nonlinear dynamics and a non-quadratic cost function for this implementation. This method allows for closing the loop on the trajectory optimization sequence while still permitting performance efficiency and control of a system with complex nonlinear dynamics making this method aptly beneficial to the Mars entry problem.

The iLQR closed-loop optimizer has found great success in optimizing and properly emulating the movement of humanoid robotics and other complex systems with many degrees of freedom and highly nonlinear dynamics such as unmanned aerial vehicles (UAV). Several approaches, and formulations thereof, have applied iLQR to improve on the trajectory optimization capabilities for these systems.

[5] See Ch. 4 for details on how the success of the backwards pass is gauged.

Li et al. [22] explores improving biological movement capabilities for robotic systems by applying iLQR to a 2- and 6-link muscle arm model. Results demonstrated improvements on efficiency and accounting for variations and disturbances in the movement compared to other existing optimal control algorithms applied to similar problems.[6] To more explicitly model these disturbances within the optimizer, iterative Linear-Quadratic Gaussian (iLQG) methods which incorporate control-dependent noise to achieve similar effects of an energy cost [42, 43]. This method has shown success for humanoid robotics completing complex tasks such as transitioning from positions of sitting on the ground to standing [39].

Additionally, over the past five years there has been demonstrated success applying iLQR to autonomous driving vehicles [34, 45] where, similar to Mars entry, there are strict control constraints and high levels of uncertainty. Solutions have addressed model predictive capability for obstacle avoidance with uncertainty awareness for the dynamic environment.

Chen et al. [8] also applies Neural Networks to demonstrate a reinforcement learning architecture capable of refining iLQR optimization without prior information on the dynamics model, a scenario more representative of real-world applications (autonomous driving, computer vision, etc.).

In [10], Noyes explores closed-loop reference trajectory optimization for Mars entry as part of their doctoral thesis. In this approach, a closed-loop optimizer is overlaid with an open-loop optimizer to achieve stronger stability and convergence guarantees on the reference trajectory while maintaining the problem constraints. Alternative to the methods presented in Ch. 1, Noyes focuses on expanding the control-limited DDP approach presented by [40] to demonstrate robust entry performance for MSL- (3 mT in mass) and SRL-class (5 mT in mass) missions. The improvements made on the DDP algorithm demonstrate improved entry flight path angle as well as reference control and feedback gains presenting this as a viable approach to closing the loop on entry trajectory optimization.

[6] See ref. [19], [13], and [4] for more work on applying iLQR for improving bio-inspired robotic performance.

While this thesis takes a similar approach of stacking open- and close-loop optimization to develop a stronger reference trajectory, and therefore, more robust entry guidance, this work focuses more on directly implementing iLQR to avoid the computational cost of taking second-order derivatives of the nonlinear dynamics. Additionally, to build on Noyes's results, this work extends the reference trajectory optimization for guidance assessment to more challenging mission scenarios with greater mass requirements (20-40 mT) while still adhering to strict terminal conditions. For more detail on how iLQR is implemented for this approach, see Ch.4 of this thesis.

Chapter 4

Methodology

As mentioned in Ch. 1, the competing objectives of adhering to mission constraints while accounting for uncertain model and flight conditions makes Mars entry guidance a challenging problem to solve. Additional goals of advancing mission scale to incorporate high-mass and low-lift entry vehicle design for precise, high-elevation landing invokes the need for a powerful guidance method capable of meeting these challenges.

The presented approach targets reference trajectory optimization to build a motion planning model robust against strict formulation constraints and adaptable nominal deviations. To achieve these capabilities, this program utilizes a two-tiered, open- and closed-loop trajectory optimization approach with uncertainty analysis on reachable terminal states to address feasibility in landing future large-scale missions on Mars. The structure of this program is presented as:

1. Open-loop nominal trajectory for reference tracking using NLP solver.
2. Closed-loop trajectory optimizer that builds off the open-loop solution using an iLQR controller tailored to this problem.
3. Reachable set analysis performed via Monte Carlo (MC) simulation to compute error estimation on terminal states.

Applying an open-loop solution allows for a baseline nominal trajectory constructed within state and control bounds to build the closed-loop solution off of.

Unlike most reference tracking methods which only compute an open-loop solution, this approach overlays a closed-loop, iLQR optimizer for enhanced performance. Additionally, optimization through a closed-loop formulation provides benefits of performance stability and convergence guarantees no provided in the open-loop. Finally, uncertainty quantification on reachable terminal states based on deviations from the nominal model conditions allows for better understanding of performance capabilities for such high-stakes missions.

The following sections will review the problem formulation in detail including considerations for the dynamics model, entry guidance conditions and modeling of vehicle conditions as well as environmental conditions. This chapter will also review methodology for constructing solutions in the open- and closed-loop forms along with the MC simulation setup.

4.1 Problem Formulation

4.1.1 Atmospheric Model

Table 4.3 below outlines the necessary constants related to Mars used in this work.

Mars Parameters	
Surface Acceleration due to Gravity, g_0 [m/s^2]	3.71
Surface Density, ρ_0 [kg/m^3]	0.0158
Equatorial Radius, R_M [km]	3396.2
Gravitational Constant, μ_M [km^3/s^2]	42,840
Scale Height, H [km]	9354

Table 4.1: Martian Planetary Conditions

An exponential atmospheric model shown in the following equation was used to map the density of the Martian atmosphere.

$$\rho = \rho_{ref} \exp\left(\frac{h_{ref} - h}{H}\right) \tag{4.1}$$

In this model, the reference altitude is taken at the surface ($h_{ref} = 0$ km) and a reference surface density of $\rho_{ref} = 0.0158$ kg/m^3. The results of this model over the entry altitude are presented in Fig. 4-1. As mentioned in the introduction of this thesis, the Martian atmosphere is significantly thinner than that of Earth making entry vehicle deceleration far more difficult than previous crewed missions. This phenomenon is represented in Fig. 4-2.

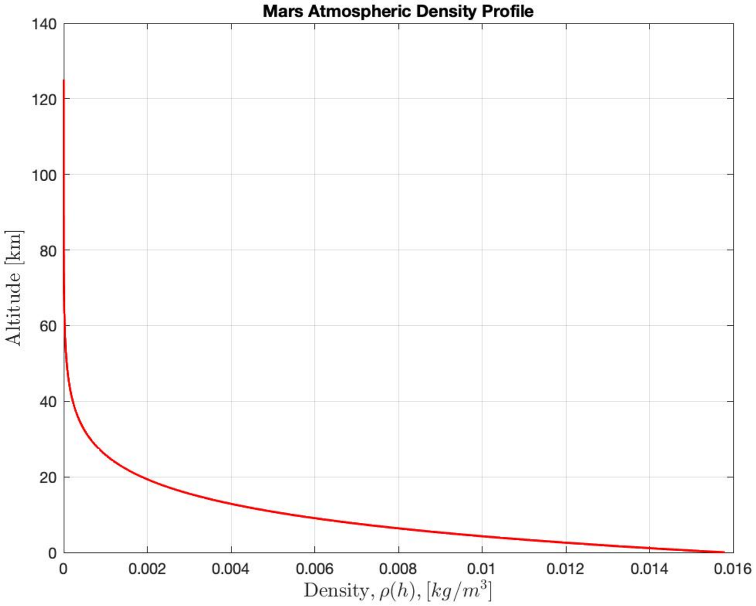

Figure 4-1: Martian Density Profile

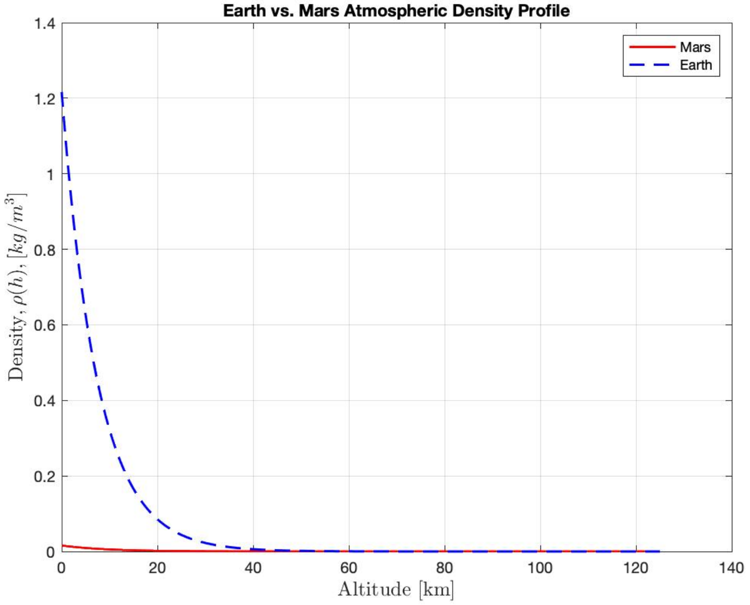

Figure 4-2: Earth and Mars Atmospheric Density Comparison

However, given that thermodynamic and fluid dynamic effects cause the atmosphere to be nonuniform, a more representative model provided by NASA based on the temperature and pressure calculations for the stratosphere and troposphere, respectively, are used for this approach.[1]

For $h > 7000$ m:

$$T = -23.4 - .00222 * h \tag{4.2a}$$

$$p = .699 * \exp\left(-.00009 * h\right) \tag{4.2b}$$

For $h < 7000$ m:

$$T = -23.4 - .00222 * h \tag{4.3a}$$

$$p = .699 * \exp\left(-.00009 * h\right) \tag{4.3b}$$

In this case, density was calculated with the following equation:

$$\rho = \frac{p}{0.1921 * (T + 273.1)} \tag{4.4}$$

[1] Equations 4.2 and 4.3 are provided by NASA Glenn Research Center website: https://www.grc.nasa.gov/www/k-12/airplane/atmosmrm.html

4.1.2 Entry Equations of Motion

For this paper the Mars EDL sequence is defined as an entry vehicle entering the Martian atmosphere at an altitude of 125 km and terminating prior to the terminal descent phase where a parachute, inflatable deployment, or some other terminal descent strategy is enacted. The 3-DOF dynamics that correspond with this sequence are listed below [21].

$$\dot{h} = v \sin\gamma \tag{4.5a}$$

$$\dot{v} = -\frac{D}{m} - g\left(h\right) \sin\gamma \tag{4.5b}$$

$$\dot{\gamma} = \frac{L}{mv} \cos\sigma + \left(\frac{v}{r} - \frac{g(h)}{v}\right) \cos\gamma \tag{4.5c}$$

$$\dot{\theta} = \frac{v \cos\gamma \sin\psi}{r \cos\gamma} \tag{4.5d}$$

$$\dot{\lambda} = \frac{v}{r} \cos\gamma \cos\psi \tag{4.5e}$$

$$\dot{\psi} = \frac{v}{r} \sin\psi \cos\gamma \tan\lambda + \frac{L}{mv \cos\gamma} \sin\sigma \tag{4.5f}$$

Although 3-DOF dynamics are formulated with assumptions such as stationary atmosphere and non-rotating planet, this method is suitable for system analysis and design given that most entry guidance relies on translational motion [21]. Additionally, given the instantaneous nature of the entry sequence these assumptions are acceptable for this work.

Furthermore, by simplifying the equations of motion to a longitudinal model, the target state is formulated in terms of downrange replacing the longitude and latitude integration shown with the following equations [21]. This is best suited for dynamical analysis.

$$\dot{h} = v \sin\gamma \tag{4.6a}$$

$$\dot{v} = -\frac{D}{m} - g\left(h\right) \sin\gamma \tag{4.6b}$$

$$\dot{\gamma} = \frac{L}{mv}\mathrm{u} + \left(\frac{v}{r} - \frac{g(h)}{v}\right)\cos\gamma \tag{4.6c}$$

$$\dot{s} = \cos\gamma \tag{4.6d}$$

The above longitudinal dynamics are formulated with bank angle, σ being the control input ($\mathrm{u} = \cos\sigma$) and using the following aerodynamic expressions representing gravitational, drag, and lift forces, respectively:

$$g(h) = \frac{\mu}{(h + R_M)} \tag{4.7a}$$

$$D = \frac{1}{2}C_D\rho v^2 A \tag{4.7b}$$

$$L = \frac{1}{2}C_L\rho v^2 A \tag{4.7c}$$

4.1.3 Entry Guidance Considerations

As discussed in literature [23], entry guidance in the aerospace field is the process of steering a hypersonic vehicle through a planetary atmosphere by providing initial conditions to achieve the terminal conditions. For the purposes of this paper, entry guidance takes place from atmospheric entry on Mars (h = 125 km) to the point of parachute or inflatable deployment, defined by a Mach number $\in [1.4, 2.2]$ which translates velocities $\in [320, 540]$ m/s at termination of the entry phase.

Provided the Martian atmosphere is thick enough to induce significant heating levels on an entry vehicle while thin enough to not provide adequate terminal descent velocity from drag alone [6], entry guidance techniques must consider maximizing the terminal altitude while still reducing velocity to terminate within the specified entry limits.

Of the closed-loop, optimal control work applied to Mars entry trajectories, few studies investigated the capabilities for large entry vehicle design with significantly higher ballistic coefficients and lower lift-to-drag ratios in comparison to previously flown payload landings on the Martian surface.

Vehicle Model

To verify the optimization setup, this study will use conditions from the Mars Science Laboratory (MSL) mission. MSL successfully landed the largest payload on Mars using bank angle control, making way for greater capabilities for successfully landing future missions including the most recent Mars 2020 lander mission which takes after the MSL entry design [32]. The MSL vehicle design is standard with blunt-body, 70^o cone aeroshell designs (shown in Fig. 4-3) flown on the Apollo missions previous Mars lander missions. Therefore, the MSL vehicle configuration, standard with the blunt-body aeroshell design, along with the initial and terminal entry conditions will be used to confirm a proper optimization setup.

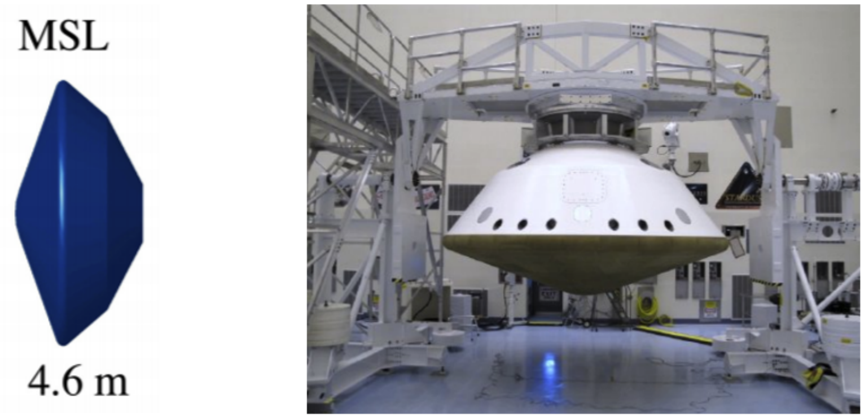

Figure 4-3: MSL Vehicle Design

Building off the MSL verification, a 40 mT (40,000 km) blunt-body vehicle design consistent with parameters projected by Mall et al. [27] will be applied to the presented computational framework to test the limits of the approach by implementing this large-scale design. Specifics on vehicle parameters are presented in Table 4.2.

Constraints

As part of determining the success of the calculated trajectory, values for dynamic pressure $\bar{q}$, heat rate $\dot{Q}$, and peak acceleration a_n (in Earth g's) were computed with equations 4.8a-c:

Parameter	Vehicle	
	MSL	Blunt Body
Mass, m [kg]	3151	40,000
Nose Diameter, D_n [m]	4.5	10
Surface Area, A [m^2]	15.9	78.54
Ballistic Coefficient, β [kg/m^2]	121	300
Lift-to-Drag Ratio, L/D	0.24	0.6

Table 4.2: Vehicle Parameters [9, 27, 30]

$$\bar{q} = \frac{1}{2}\rho v^2 \tag{4.8a}$$

$$\dot{Q} = K_q \left(\frac{\rho}{r_n}\right)^N v^M \tag{4.8b}$$

$$a_n = \frac{\sqrt{L^2 + D^2}}{m * g_{0,E}} \tag{4.8c}$$

Where m is the vehicle mass and L and D are lift and drag forces, respectively. All remaining constants are outlined in Table 4.3.

K_q	1.903E-07
r_n	0.6
N	0.5
M	3
Earth Gravitational Const. $g_{0,E}$ [m/s^2]	9.81

Table 4.3: Entry Flight Parameter Constants

Condition	Values		
	Nominal	Min. Value	Max. Value
Dynamic Pressure, $\bar{q}$ [kPa]	12	14.1	16.0
Heat rate, $\dot{Q}$ [km]	1000	1950	2339
G-loading, g^s [km]	5	11.9	13.4

Table 4.4: Entry Flight Parameter Bounds [46]

For the flight parameter constraints presented in table 4.4, nominal values are taken as the set limits for a crewed-scale entry presented in literature [6, 10]. However, the minimum and maximum values shown in the table reflect peak ranges seen during the MSL lander mission, reflecting realistic ranges seen during an executed flight of maximum scale.

Boundary Conditions

Boundary constraints on the state variables allow for proper nominal reference trajectory modeling in the open-loop as well as analysis on acceptable generated trajectories in the closed-loop. Table 4.5 outlines the standard state constraints used in this approach along with the initial entry values.

State	**Values**		
	Initial	Min. Terminal	Max. Terminal
Altitude, h [km]	125	4	–
Velocity, v [m/s]	5500	320	540
Flight Path Angle, γ [deg]	-15	–	–
Downrange, s [km]	0	0	900

Table 4.5: Entry Boundary Conditions [33, 37]

4.2 Open-Loop Formulation (CasADi)

The goal of the open-loop optimizer is to directly apply the boundary conditions and problem constraints required for this problem for the closed-loop optimizer to track and improve upon. This avoids having to explicitly develop a non-quadratic cost function that incorporates several constraints and performing complex derivatives on a highly non-quadratic cost function which is computationally more expensive.

4.2.1 Cost Function

Given that high-mass mission designs will cause a significant challenge for reducing vehicle speeds at high enough altitudes for parachute deployment, the open-loop formulation models these constraints by optimizing around maximum terminal altitude while constraining the terminal velocity and altitude limits

$$\max h(t_f) \tag{4.9}$$

Subject to:

$$h(t_0) = h_0 \tag{4.10a}$$

$$v(t_0) = v_0 \tag{4.10b}$$

$$h(t_f) \geq h_{max} \tag{4.10c}$$

$$v(t_f) \leq v_{max} \tag{4.10d}$$

It is important to note that for the formulation of the open-loop solution for this approach, entry flight constraints presented in Table 4.4 were not directly applied to the model, but instead overlaid on solutions to analyze the feasibility of the method based on the results.

4.2.2 Nonlinear Program Solver

CasADi is an open-source tool that permits nonlinear optimization and algorithmic differentiation making it suitable for this type of optimal control problem10. Given that the Opti modeling class works in discrete time, the Runge-Kutta method is implemented to solve the equations of motion given the set constraints [2]. Table 4.6 shows a full breakdown of the CasADi setup.

Parameter	**Value**
NLP Solver	iPOPT
Modeling Class	Opti
ODE Method	Runge-Kutta
Discretized Points	50
Max Iterations	8000

Table 4.6: CasADi Solver Setup

4.3 Closed-Loop Formulation (iLQR)

Discussed at length in Ch. 3, iLQR has demonstrated notable success in robotics and other fields as a closed-loop method for optimizing trajectories of mechanical systems with highly nonlinear dynamics while respecting control bounds. To understand how this method generates optimal solutions, the formulation implemented for this approach is outlined in the following section.

4.3.1 iLQR Structure

System Model

Following the trajectory optimization set-up outlined in Ch. 2, the *state space* is defined by discrete-time dynamics modeled by a generic function $\mathbf{f}$,

$$\mathbf{x}_{i+1} = \mathbf{f}\left(\mathbf{x}_i, \mathbf{u}_i\right) \tag{4.11}$$

models the progression of the state space $\mathbf{x} \in \mathbb{R}^n$ from i to $i+1$, in the discretized time sequence, given some control input $\mathbf{u} \in \mathbb{R}^m$. Simultaneously, a cost value is computed summing over the running cost, l (or cost at each time step), and adding the final cost, l_f. This develops from starting at some initial state $\mathbf{x}_0$ and applying a control sequence $\mathbf{U} \equiv \{\mathbf{u}_0, \mathbf{u}_1, ..., \mathbf{u}_{N-1}\}$ in the form:

$$J_0\left(\mathbf{x}, \mathbf{U}\right) = \sum_{i=0}^{N-1} l\left(\mathbf{x}_i, \mathbf{u}_i\right) + l_f\left(\mathbf{x}_N\right)$$

Where $\mathbf{x}_i$ for $i > 0$ is computed via equation 4.11 and some specified $\mathbf{x}_0$ when $i = 0$. The goal is to develop some optimal control sequence that minimizes this overall cost and optimizes the solved trajectory.

Dynamics Rollout

With initial state, x_0, and an initialized control sequence, $\mathbf{U}$, simulate the system by performing a rollout of the dynamics to obtain the trajectory through the state space, $\mathbf{X}$, for a complete trajectory $\{\mathbf{X}, \mathbf{U}\}$.

Backward Pass

To initiate the backwards pass, the following expansion coefficients are computed in the model.

$$Q_{\mathbf{x}} = \ell_{\mathbf{x}} + \mathbf{f}_{\mathbf{x}}^{\top} V'_{\mathbf{x}} \tag{4.12a}$$

$$Q_{\mathbf{u}} = \ell_{\mathbf{u}} + \mathbf{f}_{\mathbf{u}}^{\top} V'_{\mathbf{x}} \tag{4.12b}$$

$$Q_{\mathbf{xx}} = \ell_{\mathbf{xx}} + \mathbf{f}_{\mathbf{x}}^{\top} V'_{\mathbf{xx}} \mathbf{f}_{\mathbf{x}} + V'_{\mathbf{x}} \cdot \mathbf{f}_{\mathbf{xx}} \tag{4.12c}$$

$$Q_{\mathbf{uu}} = \ell_{\mathbf{uu}} + \mathbf{f}_{\mathbf{u}}^{\top} V'_{\mathbf{xx}} \mathbf{f}_{\mathbf{u}} + V'_{\mathbf{x}} \cdot \mathbf{f}_{\mathbf{uu}} \tag{4.12d}$$

$$Q_{\mathbf{ux}} = \ell_{\mathbf{ux}} + \mathbf{f}_{\mathbf{u}}^{\top} V'_{\mathbf{xx}} \mathbf{f}_{\mathbf{x}} + V'_{\mathbf{x}} \cdot \mathbf{f}_{\mathbf{ux}} \tag{4.12e}$$

With the expansion coefficients in equation 4.12, the open-loop term and feedback gain term are computed as $\mathbf{k} = -Q_{\mathbf{uu}}^{-1} Q_{\mathbf{u}}$ and $\mathbf{K} = -Q_{\mathbf{uu}}^{-1} Q_{\mathbf{ux}}$, respectively. From the expansion coefficients, the quadratic value of the model at time i is given by the following set of expressions:

$$\Delta V(i) = \frac{1}{2} Q_{\mathbf{u}} Q_{\mathbf{uu}}^{-1} Q_{\mathbf{u}} \tag{4.13a}$$

$$V_{\mathbf{x}}(i) = Q_{\mathbf{x}} - Q_{\mathbf{u}} Q_{\mathbf{uu}}^{-1} Q_{\mathbf{ux}} \tag{4.13b}$$

$$V_{\mathbf{xx}}(i) = Q_{\mathbf{xx}} - Q_{\mathbf{xu}} Q_{\mathbf{uu}}^{-1} Q_{\mathbf{ux}} \tag{4.13c}$$

Forward Pass

After completing the backward pass, a new trajectory is computed by a forward pass through discretized time, i:

$$\hat{\mathbf{x}}(1) = \mathbf{x}(1) \tag{4.14a}$$

$$\hat{\mathbf{u}}(i) = \mathbf{u}(i) + \mathbf{k}(i) + \mathbf{K}(i)\left(\hat{\mathbf{x}}(i) - \mathbf{x}(i)\right) \tag{4.14b}$$

$$\hat{\mathbf{x}}(i+1) = \mathbf{f}\left(\hat{\mathbf{x}}(i), \hat{\mathbf{u}}(i)\right) \tag{4.14c}$$

A regularization term μ is applied to the local cost to account for the non positive-definite second-order Hessian's computed on the cost:

$$\tilde{Q}_{\mathbf{uu}} + \mu \mathbf{I}_m \tag{4.15}$$

Following the regularization, modifications via a Levenberg-Marquardt parameter adds a quadratic cost around the control sequence for more conservative steps. However, regularization can apply unnecessary to control deviations. The following equations are implemented to penalize deviations in the state as opposed to the control:

$$\tilde{Q}_{\mathbf{uu}} = \ell_{\mathbf{uu}} + \mathbf{f}_{\mathbf{u}}^{\top}\left(V'_{\mathbf{xx}} + \mu \mathbf{I}_n\right)\mathbf{f}_{\mathbf{u}} + V'_{\mathbf{x}} \cdot \mathbf{f}_{\mathbf{uu}} \tag{4.16a}$$

$$\tilde{Q}_{\mathbf{ux}} = \ell_{\mathbf{ux}} + \mathbf{f}_{\mathbf{u}}^{\top}\left(V'_{\mathbf{xx}} + \mu \mathbf{I}_n\right)\mathbf{f}_{\mathbf{x}} + V'_{\mathbf{x}} \cdot \mathbf{f}_{\mathbf{ux}} \tag{4.16b}$$

$$\mathbf{k} = \tilde{Q}_{\mathbf{uu}}^{-1}\tilde{Q}_{\mathbf{u}} \tag{4.16c}$$

$$\mathbf{K} = \tilde{Q}_{\mathbf{uu}}^{-1}\tilde{Q}_{\mathbf{ux}} \tag{4.16d}$$

The impact of this approach with respect to other control-based regularization is that **K** forces the new trajectory closer to the old trajectory improving on robustness [39].

Finally, value updates from equations 4.13a-c are performed with the following calculations:

$$\Delta V(i) = \frac{1}{2}\mathbf{k}^\top Q_{\mathbf{uu}}\mathbf{k} + \mathbf{k}^\top Q_{\mathbf{u}} \tag{4.17a}$$

$$V_{\mathbf{x}}(i) = Q_{\mathbf{x}} + \mathbf{K}^\top Q_{\mathbf{uu}}\mathbf{k} + \mathbf{K}^\top Q_{\mathbf{u}} + Q_{\mathbf{ux}}^\top \mathbf{k} \tag{4.17b}$$

$$V_{\mathbf{xx}}(i) = Q_{\mathbf{xx}} + \mathbf{K}^\top Q_{\mathbf{uu}}\mathbf{K} + \mathbf{K}^\top Q_{\mathbf{ux}} + Q_{\mathbf{ux}}^\top \mathbf{K} \tag{4.17c}$$

As part of the formulation developed in [39], a line-search parameter is, $0 < \alpha \leq 1$ introduced to better tune the solution. This parameter is integrated with the control update as follows:

$$\hat{\mathbf{u}}(i) = \mathbf{u}(i) + \alpha\mathbf{k}(i) + \mathbf{K}(i)\left(\hat{\mathbf{x}}(i) - \mathbf{x}(i)\right) \tag{4.18}$$

4.3.2 Cost Function

For the closed-loop formulation, the cost function is modified from that applied to the open-loop solution to incorporate a running cost term represented in the following form:

$$J_i = \sum_{j=i}^{N-1} \ell(\mathbf{x}_j, \mathbf{u}_j) + \ell_f(\mathbf{x}_N) \tag{4.19}$$

where at each iteration,

$$\ell(\mathbf{x}_j, \mathbf{u}_j) = \left(\mathbf{x} - \mathbf{x}^*\right)^\top Q\left(\mathbf{x} - \mathbf{x}^*\right) + \mathbf{u}^\top R\mathbf{u} \tag{4.20}$$

And the $\mathbf{x}^*$ term denotes the reference trajectory tracked from the open-loop solution.

4.4 Uncertainty Quantification (MC Simulation)

To understand the robustness of the proposed computation program against inevitable uncertainties to nominal modeling, MC simulation are implemented to determine reachable sets from a uniform initial condition distribution. This approach was chosen given the fairly high frequency of the closed-loop optimizer (1 sec - 45 sec depending on the implementation), ease of computational implementation, and error estimation accuracy. Table 4.7 outlines the max disturbance to the distribution determined from previous modeling in literature [33, 37, 46].

Initial State	σ
h_0	1000 m
v_0	50 m/s
γ_0	0.1^o
s_0	1500 m

Table 4.7: MC Simulation σ Values Applied to Normal Distribution [2]

[2] The σ symbol used here represents standard deviation consistent with statistics representation. This should not be confused with the σ symbol used to represent the bank angle control input applied to the entry equations of motion expressed in Sec. 4.1.2 of this chapter.

Chapter 5

Results

5.1 MSL Verification

To demonstrate the capabilities of the guidance approach presented in this work prior to scaling to a crewed design, the system was tested against MSL-derived flight conditions outlined in literature [6, 35, 46] to establish performance capabilities against the current state-of-the-art.[1]

As outlined in Ch. 4, this system constitutes a tiered approach to reference trajectory optimization by establishing a baseline open-loop solution and building-up using a closed-loop formulation. The following sections outline results from tier 1 - implementing the open-loop NLP solver in CasADi, and then tier 2 - referencing the open-loop solution to construct closed-loop trajectory optimization using the iLQR method.

[1] While Mars 2020 is the most recent and highest-performing Mars mission to date, MSL condition were used for this work given that literature on MSL is more extensive and the achieved mission objectives are comparable.

5.1.1 Tier 1: Open-loop Solution

For NLP solvers, establishing reasonable and predictive initial guesses is just as important as the formulation itself. Table 5.1 compares the initial guesses to the final conditions solved by the programmer.[2]

State	**Initial Guess**	**Final Condition**
h	11.8 km	12.8 km
v	854 m/s	488 m/s
γ	360^o	346^o
s	616 km	626 km
t_f	–	231.1 sec
run time	7.6 sec	

Table 5.1: Open-Loop NLP Solver Values for MSL Design

The final state values demonstrate considerable adherence and improvement on the initial guesses formulated for the problem with velocity terminating at 488 m/s (Mach 2) and altitude at 17.1 km, both of which are sufficient for parachute deployment. Additionally, from the flight profile for each state, shown in Fig. 5-1, the trajectory does not dip below 12.9 km avoiding interference with the surface.

From the state and control response in Fig. 5-1 and 5-2, the system switches to full-lift control ($\sigma = 0^o$) around 100 seconds causing the g-load 18 g's, 2 g's above recorded MSL conditions [46].

[2] Guesses are established as a combination of terminal entry conditions recorded for MSL and Mars 2020. Additionally, some modification, particularly for expected velocity, were made to allow for feasibility of the solution and set constraints

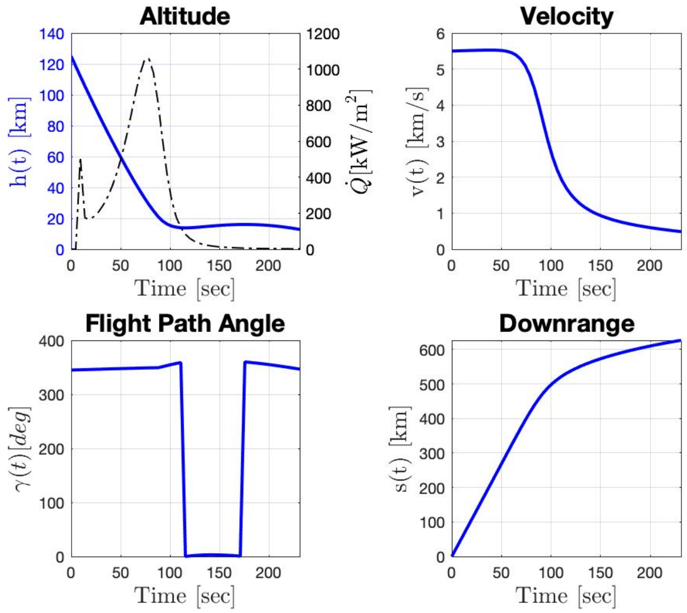

Figure 5-1: Open-Loop State Dynamics for MSL Conditions

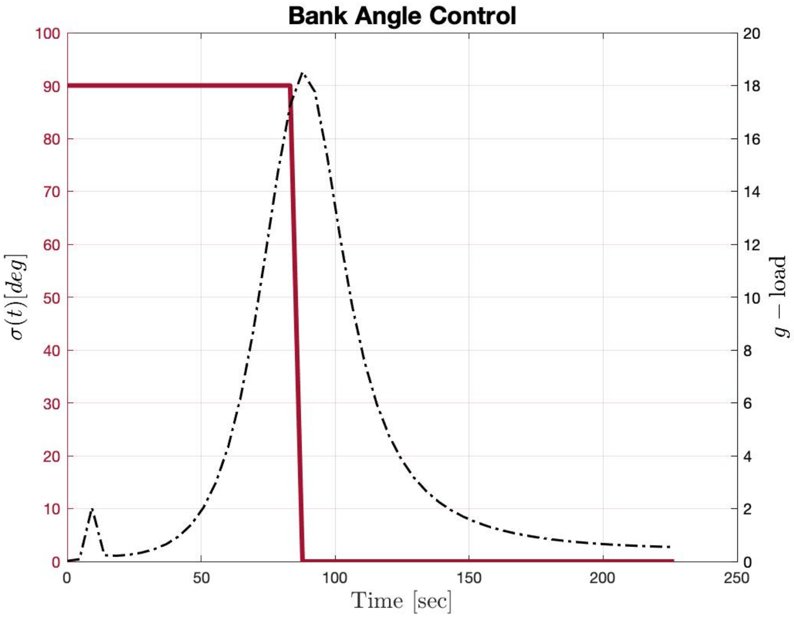

Figure 5-2: Open-Loop Control for MSL Conditions

5.1.2 Tier 2: Closed-Loop Solution

Building off of tier 1 by tracking the reference trajectory through the energy optimal cost function, the following results demonstrate performance of the closed-loop iLQR optimal controller.

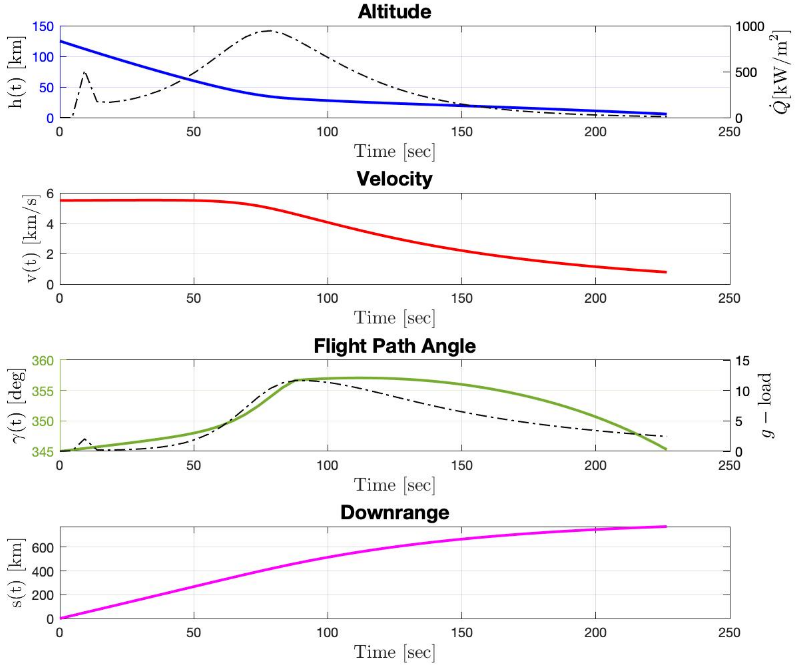

Figure 5-3: iLQR Closed-Loop State Dynamics for MSL Conditions

Fig. 5-3 and 5-4 demonstrate the state and control performance, respectively. For this closed-loop performance, tuning of the cost function and alpha parameter became extremely important to ensure adequate tracking of the reference trajectory. For this and the following test case, tuning was selected by trial and error based on values that provided the most optimal performance.

As seen from the results in Table 5.2, terminal conditions deviate from the reference trajectory causing mainly altitude and velocity values to fall outside of required ranges for parachute deployment. It is important to note that position landing accuracy for this and subsequent cases is determined from the approximate terminal downrange from the MSL mission.

While the results do not precisely track the terminal conditions from the open-loop optimizer, they do however demonstrate smoothing on the overall trajectory (each of the states) improving on the flight conditions presented in Fig. 5-5 and outlined in Table 5.3. This effect can also be seen from Fig. 5-6 which maps the entry

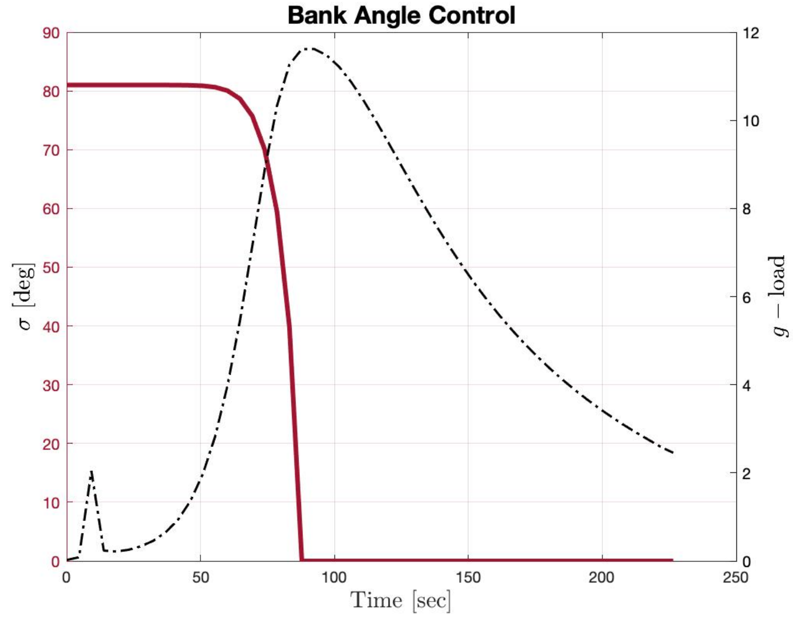

Figure 5-4: iLQR Closed-Loop Control for MSL Conditions

altitude against the velocity. This graph shows that the trajectory relies heavily on low-altitude conditions to slow the vehicle over 4000 m/s. However, the work still demonstrates significant capability in establishing an optimal trajectory with improved performance for heat rate, dynamic pressure, and g-loading all maximizing below the lower-bound limit derived from MSL.

State	**Initial Condition**	**Final Condition**	**Nominal Final Condition**
h	125 km	5.8 km	$>$ 6 km
v	5500 m/s	788 m/s	$<$ 520 m/s
γ	-15^o (345 ^{o})	345^o	–
s	0 km	773 km	616 km
t_f	–	231.1 sec	–
run time	2.28 sec		

Table 5.2: Initial and Terminal Conditions for iLQR Closed-Loop Solution with MSL Design

Overall, the two-tiered optimization approach for Mars entry trajectory demonstrates capabilities on developing proper trajectory design. While the MSL verification test showed deviations from expected terminal conditions, these values still lie within the same order of magnitude without considering maneuvering techniques

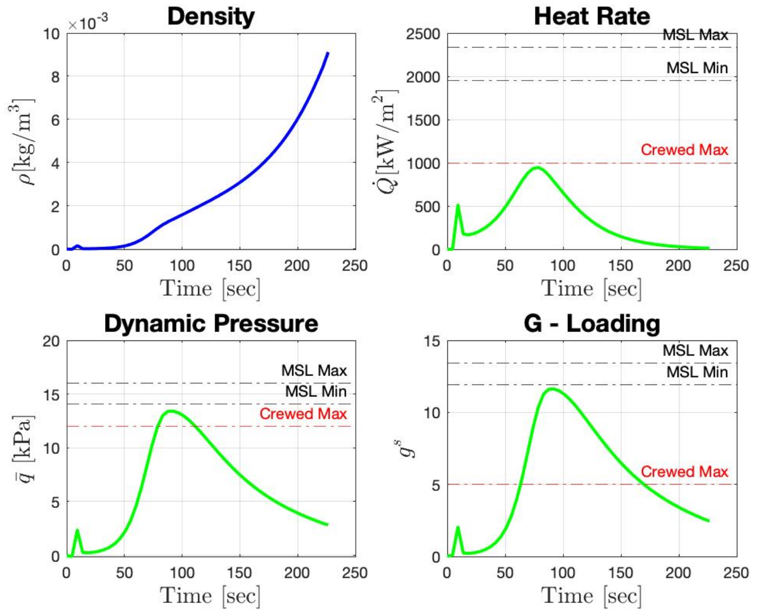

Figure 5-5: iLQR Entry Flight Conditions for MSL

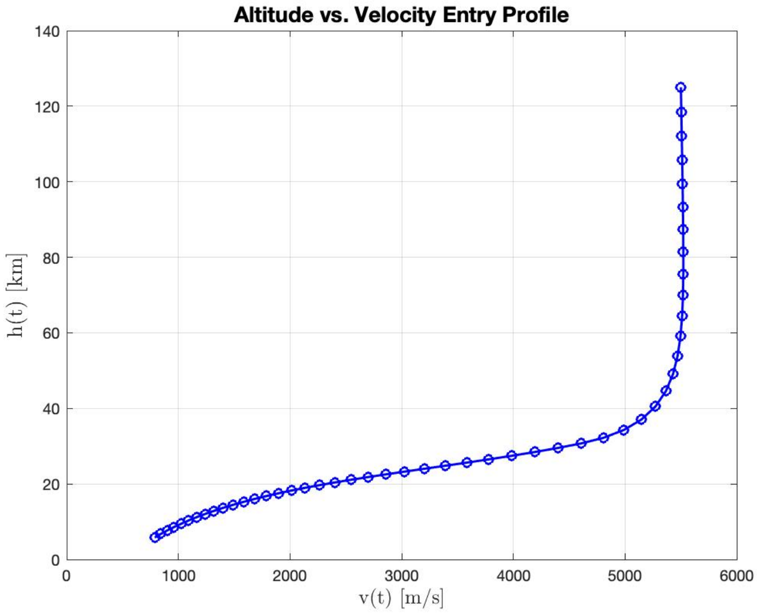

Figure 5-6: iLQR Entry Profile for MSL Conditions

applied to MSL entry guidance to achieve those values. Additionally, the significant reduction of entry flight conditions below prompts further investigation on how this approach approach matches up with a larger scale mission on the order of a crewed mission design.

	Open-loop	Closed-loop
$\bar{q}$ [kPa]	21.4	13.4
$\dot{Q}$ [kW/m^3]	1057.7	944.6
a_n [g$_0$]	18.5	11.6

Table 5.3: Max Flight Conditions from NLP Solver

5.2 Crewed-Scale Design

Part of the motivation for this thesis is to build up capabilities in entry guidance to establish a viable method for potential crewed-landing missions. Detailed extensively in Ch. 1 of this thesis, the prospect of landing humans or even larger-scale missions on the Martian surface will require guidance technology robust to uncertainty and disturbances from nominal guidance schemes. The results presented in the following sections reviews performance of the proposed method with respect to a large-scale, crewed entry vehicle design. The objective is to demonstrate viable performance that significantly slows the entry vehicle while maintaining high altitudes.

To address the objective, the following results apply the same approach analyzed in the previous section to a mission design of crewed scale. The major indicators for a mission of this caliber rely on stronger limitation to maximum levels for entry flight conditions along with more restricting terminal altitude and landing location error.

5.2.1 Tier 1: Open-loop Performance

To formulate the nominal trajectory in the first tier of the program, the NLP solver developed and demonstrated in the MSL verification analysis is applied with modified constraints. Working off the same initial guesses established for the MSL verification test, Table 5.4 compares these guesses to the terminal conditions for this crewed case.

While terminal velocity increased to the upper-bound of accepted parachute deployment demonstrated in Fig. 5-7, all calculated nominal values lie within the desired ranges expressed in Table 4.5 and improve on the initial guesses demonstrating validity in the first tier of the approach. Additionally, open-loop control shown in Fig. 5-8 is consistent with bank angle modulation constraints between 0 and 90 degrees.

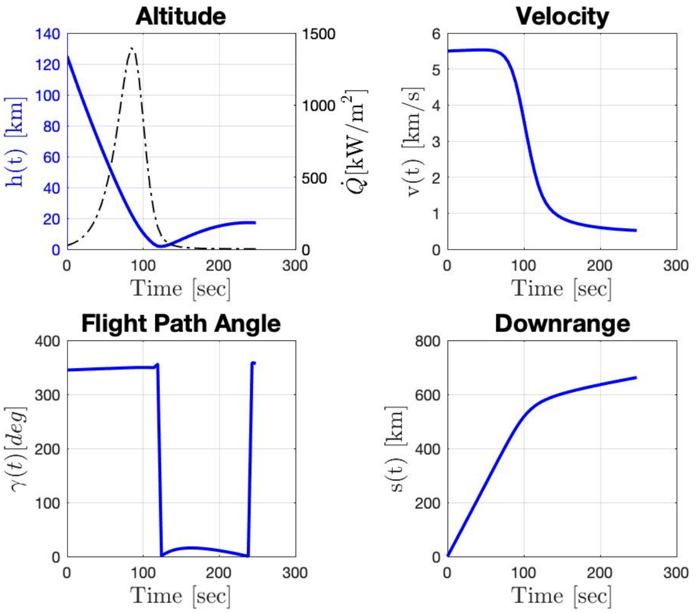

Figure 5-7: Open-loop State Dynamics for Crewed Design

State	**Initial Guess**	**Final Condition**
h	11.8 km	17.1 km
v	854 m/s	520 m/s
γ	360^o	357^o
s	616 km	662 km
t_f	–	247.93 sec
run time	44.5 sec	

Table 5.4: Open-Loop NLP Solver Values

However, the abrupt change in lift vector direction causes high g-loading, plotted against the control in the same figure.

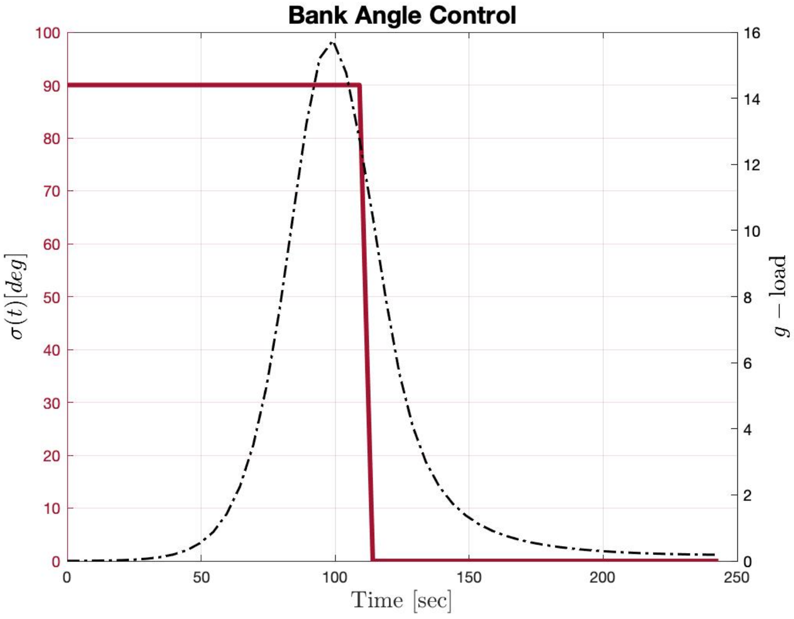

Figure 5-8: Open-loop Control for Crewed Design

5.2.2 Tier 2: Closed-loop Performance

Building up to the second tier of the approach, state and control weights were tuned in the energy optimal cost function to better constrain the closed loop terminal values and perform better tracking of the boundaries set in the open-loop formulation.

State	Initial Condition	Final Condition	Nominal Final Condition
h	125 km	5.82 km	$>$ 6 km
v	5500 m/s	621.5 m/s	$<$ 520 m/s
γ	-15^o (345 ^{o})	340.2^o	–
s	0 km	635.3 km	616 km
t_f	–	247.93 sec	–
run time	45.67 sec		

Table 5.5: Initial and Terminal Conditions for iLQR Closed-Loop Solution with Crewed-Scale Design

The results outlined in Table 5.5 demonstrate close tracking to nominal conditions. By finely tuning the line-search parameter, α, for the backward pass search of the iLQR method, results for the crewed design were able to track nominal bounds more consistently producing a terminal velocity over 100 m/s less than the computed value

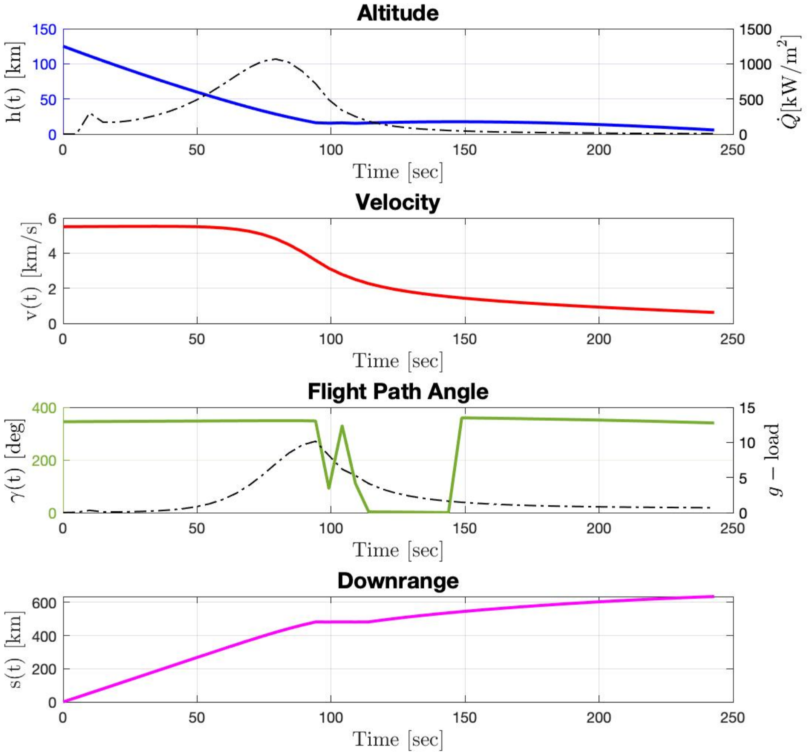

Figure 5-9: iLQR Closed-loop State Dynamics for Crewed Design

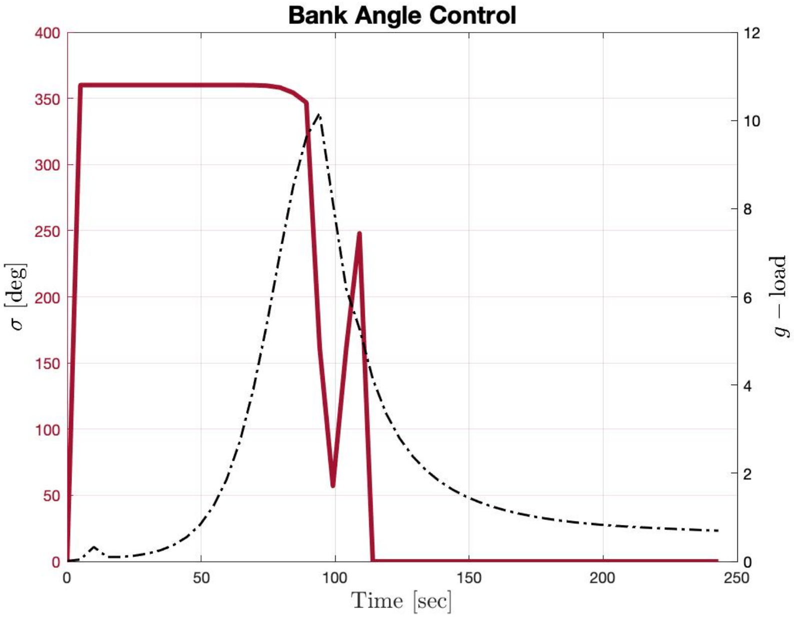

Figure 5-10: iLQR Closed-loop Control for Crewed Design

for MSL verification shown in Fig 5-9. Additionally, the control input shown in Fig. 5-10 is able reduce g-loading for a less abrupt modulation of the lift vector. Finally, the entry profile presented by Fig. 5-12 demonstrates that the trajectory was capable of remaining at higher altitudes, closer to 20 km, for longer periods of time while the vehicle slowed to terminal conditions. While these improvements came at the computational expense of run time increasing the length an order of magnitude over that for MSL verification, these results strongly demonstrate the iLQR optimal controller's ability to apply stable, closed-loop control to complex nonlinear systems.

	Open-loop	Closed-loop
$\bar{q}$ [kPa]	39.7	25.6
$\dot{Q}$ [kW/m^3]	1397.5	1067.4
a_n [g$_0$]	15.7	10.2

Table 5.6: Max Crewed Flight Conditions from NLP Solver

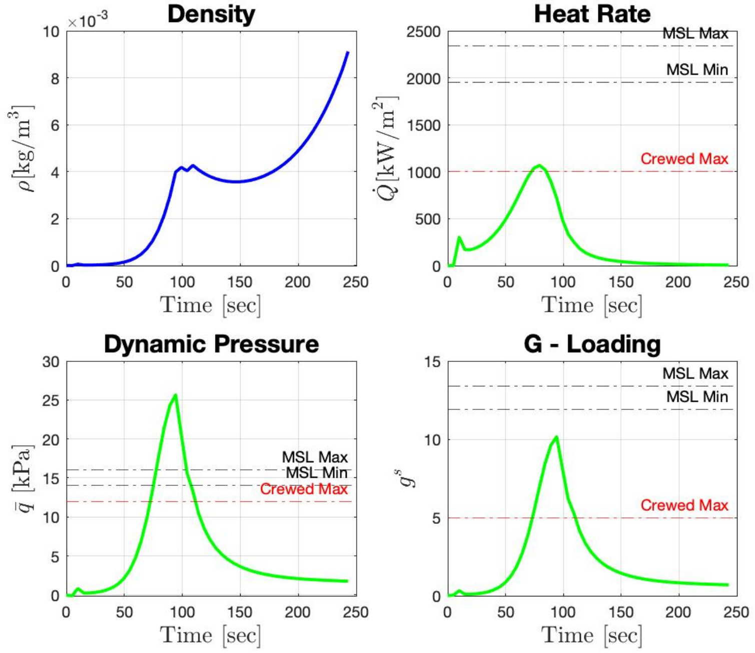

Figure 5-11: iLQR Entry Flight Conditions for Crewed Design

Furthermore, dynamic pressure, heat rate and g-loading entry flight conditions were significantly improved upon from the open-loop formulations seen in Fig. 5-11. While the overall maximum values are too large to conduct a crewed mission, all except for dynamic pressure were reduced below MSL limits. This demonstrates potential for improvement on flight conditions with the provided approach. More work on respecting these flight limitation should be done by looking into potential vehicle material and other design characteristics.

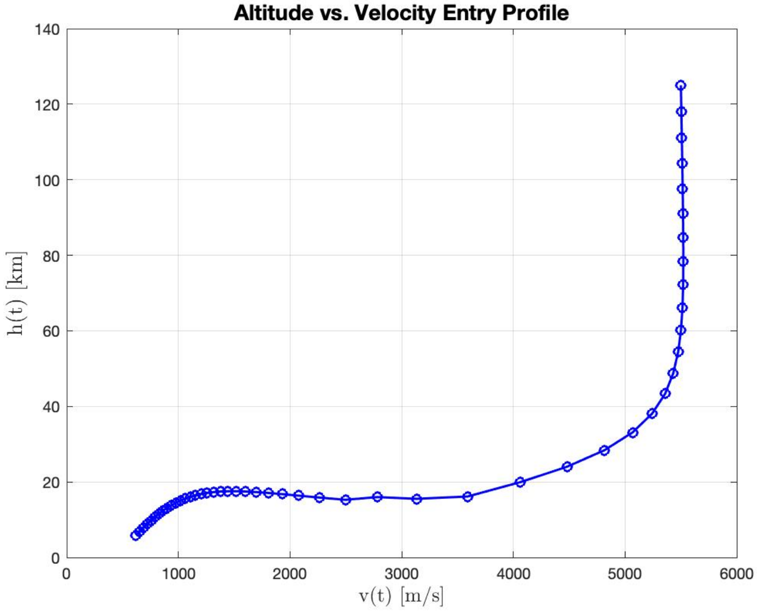

Figure 5-12: iLQR Entry Profile for Crewed Design

5.2.3 Reachable Set Analysis

As presented in Ch. 1 Reachable Sets are a form of uncertainty analysis commonly applied to trajectory design. Understanding the potential terminal states from variations to nominal initial conditions is imperative for ensuring flight readiness of the approach given that at any point during Mars Entry wind gusts, dust storms and even inaccuracies in the guidance model from true flight conditions can disturb the optimal trajectory plan. In accordance, the following results perform analysis on reachable sets using Monte Carlo simulation.

While the benefit of having a closed-loop solution to perform optimal control is the guaranteed convergence capability, in the context of this formulation consisting of strict constraints that are no explicitly set in the closed-loop setup this trait presents concerns for what terminal conditions are considered *achievable*. In terms of altitude, not only must the solution project a trajectory that does not hit below a 0 km altitude, but also terminate above specified altitude limits for the following phases of EDL. This constraint is consistent with terminal velocity for parachute deployment and terminal downrange for precision landing.

To address this concern, the reachable set formulation presented in the following results performs *rejection sampling* of the Monte Carlo simulations. This allows for an uncertainty quantification method aware of the terminal constraints to construct the available trajectory envelope given the uncertainty distribution. The results from this analysis are presented in the following figures.

Given the computational cost of the iLQR formulation for the crewed design, Monte Carlo simulations scaled on this cost by the number of iterations. Additionally, given the results would be rejecting iterations falling outside constraint bounds, it was necessary run enough iterations to to properly construct the profile. To strike a balance between computational cost and proper result generation, Table 5.7 outlines the considerations adapted for this approach.

State	Target[3]	Min. Terminal	Max. Terminal	Mean	Std.
Altitude, h [km]	5.82	0	–	8.72	4.95
Velocity, v [m/s]	621.5	–	1000	719.7	143.8
Downrange, s [km]	635.3	500	700	646.7	14.8

Table 5.7: MC Rejection Sampling for the Crewed Case

The bounds for this method were determined based on simulations performed without sampling. Although they allow for terminal conditions outside of target entry ranges, they lie close enough to the conditions to allow additional measures to constrict them back to nominal while remaining broad enough to allow for a representative set. Additionally, given the computational cost of the MC simulation, 500 samples were taken of the nominal initial conditions to form proper analysis within a capable time-frame. The results of this approach are presented below.

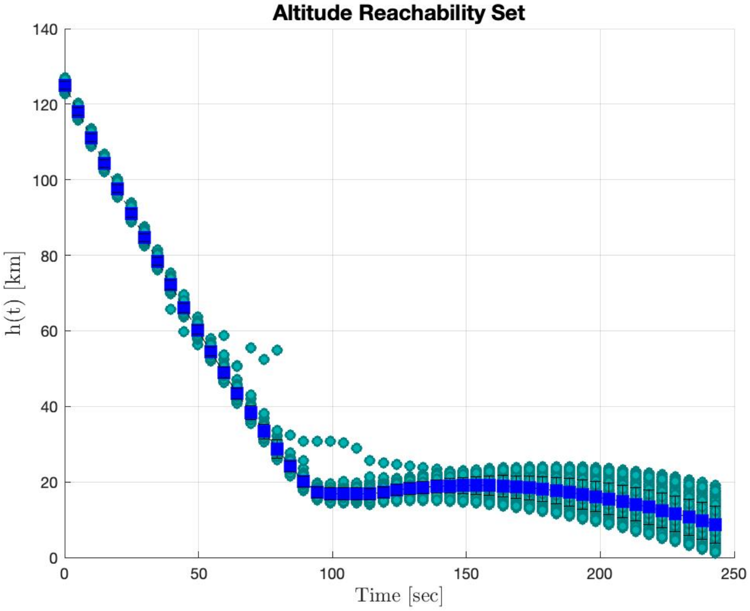

Figure 5-13: Altitude Reachable Set with Rejection Sampling for Crewed Design

Fig. 5-13 represents the altitude spread from normally distributed initial conditions where $h_0 \sim \mathcal{N}(0, 1000\,m)$. As mentioned previously, the closed-loop form does no account for strict state or control constraints, but instead, utilizes tuning parameters to track nominal values in it's performance. Therefore, samples for altitude that terminated below or equal to 0 km (signifying a vehicle crash) were rejected when de-

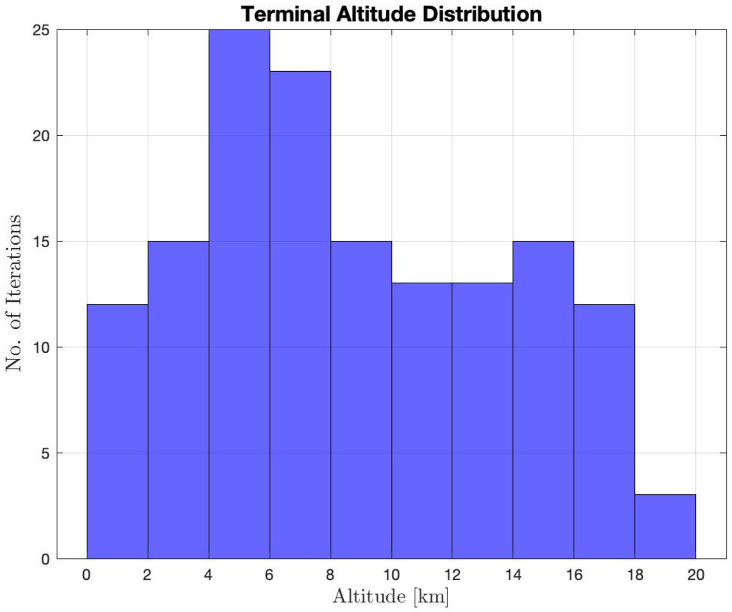

Figure 5-14: Altitude Terminal Range with Rejection Sampling

termining the spread. However, values below the target altitude range for parachute deployment were kept to demonstrate, for all *feasible* trajectories, how many of those trajectories are viable given the problem constraints. Fig. 5-14 represents the terminal altitude spread among the accepted iterations demonstrating a large concentration of terminal altitudes between 4 and 10 km, which is the ideal range for completing before parachute deployment.

Unlike altitude, the reachable set for the velocity state, shown in fig. 5-15, demonstrates concentration on the higher end of the optimal bound for parachute deployment with a distribution of $v_0 \sim \mathcal{N}(0, 50\, m/s)$. While parachute deployment calls for terminal altitudes $\in [320, 540]$ m/s, terminal altitudes from the accepted MC simulation concentrate between 500 and 700 m/s seen in Fig. 5-16. Nonetheless, the spread is still tuned enough to track the nominal terminal conditions.

Downrange, however, was the more difficult of the state conditions to tune. While altitude and velocity rejected less than half their samples, downrange had to reject close to three quarters of its samples due to certain terminal condition regressing away from the target. MC simulation on downrange applied a distribution of $s_0 \sim \mathcal{N}(0, 1500\, m)$. While significantly more iterations were rejected for downrange,

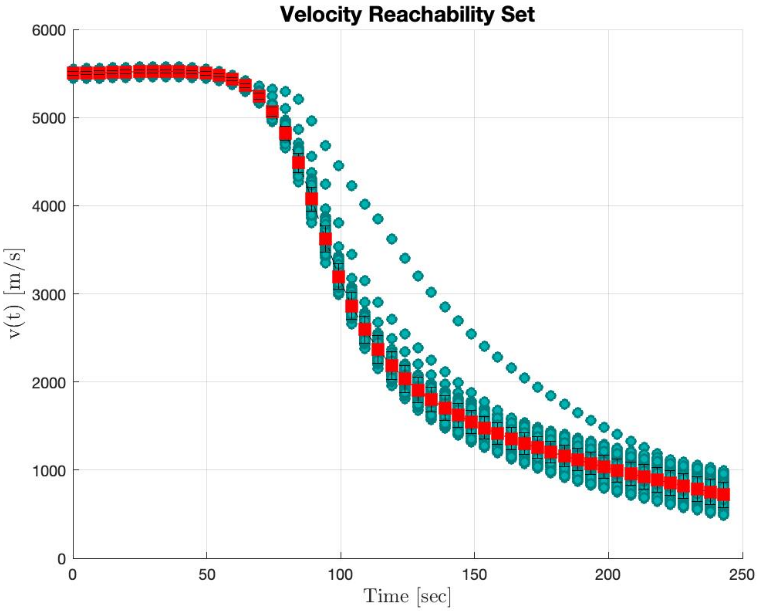

Figure 5-15: Velocity Reachable Set with Rejection Sampling for Crewed Design

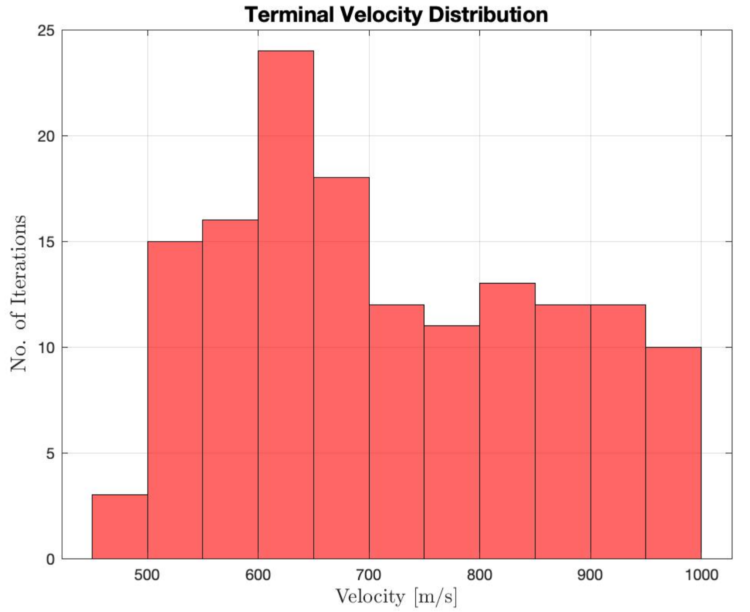

Figure 5-16: Velocity Terminal Range with Rejection Sampling

the spread of reachable sets shown in Fig. 5-17 track closely to projected terminal downrange of approximately 600km from MSL and Mars 2020 missions. Additionally, the range of accepted terminal conditions shown in Fig. 5-18 demonstrate capability for all iterations to lie between the 600 to 700 km range.

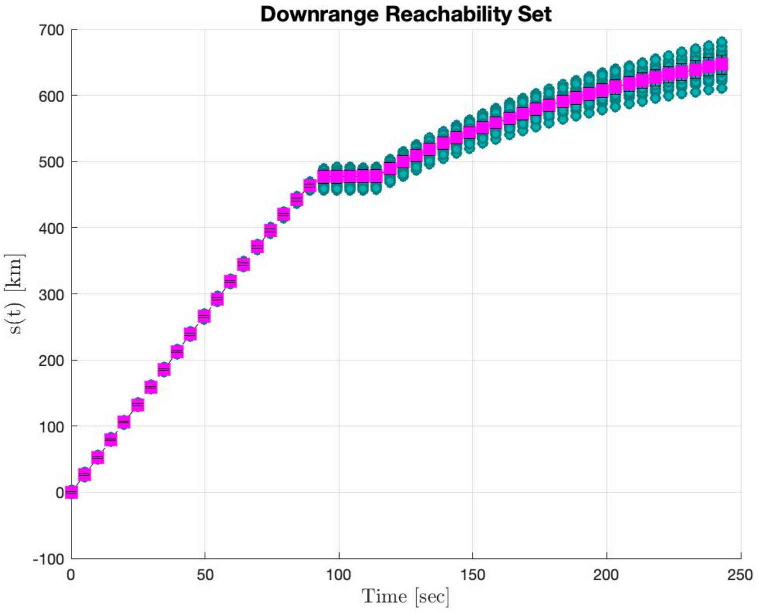

Figure 5-17: Downrange Reachable Set with Rejection Sampling for Crewed Design

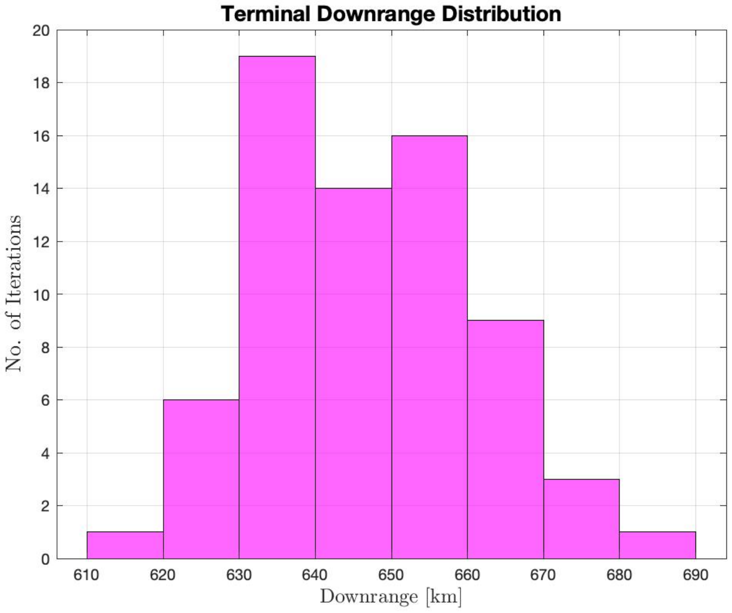

Figure 5-18: Downrange Terminal Range with Rejection Sampling

However, the issues encountered for downrange versus altitude and velocity reveal the constraints to tuning in the closed-loop formulation. Numerous tested cases revealed that any combination of tuning the state and control error, as well as the line-search parameter, only permit true nominal tracking of one state the expense of other(s). Yet, even with the tuning constraint, the reachable sets demonstrated above for the complex crewed entry scenario adequately account for constraining the terminal conditions, through iLQR tuning and rejection sampling, for an uncertainty range applicable to the nature of the problem.

Chapter 6

Discussion

Overall, while the guidance method presents some restrictions to meeting mission objectives, it still adequately demonstrates closed-loop entry guidance capability through reference trajectory formulation and reachable set analysis. This determination becomes significantly applicable when considering the complex nature of the crewed design highlighted in this approach. To provide an review on the explored methods, the following sections outline considerations to the approaches based on the achieve results.

6.1 Crewed-scale Application

A 40 mT, blunt-body and low-lift entry vehicle design was chosen as a full-scale method for testing the limitations of this approach. As discussed in Ch. 1, few closed-loop guidance approaches target designs beyond those for previously flown mission. As outlined in the results, reference tracking in the closed form does not allow for strict adherence constraints on state variables. However, iLQR high-performance ability for nonlinear system operating in dynamic environments caused it to stand out as a viable approach for this problem. While the results are representative of improvements on reference trajectory design for this complex application, additional considerations should be noted going forward.

6.1.1 Open-loop Formulation

Constructing the open-loop NLP solver became increasingly difficult by increasing the scale of constraints to the problem. While the solutions formulated in CasAdi provided an adequate reference trajectory to the computational program, a program solver such as GPOPS may be better suited to perform offline open-loop optimization. Unlike CasADi, GPOPS is exclusively an optimal control problem that uses a very specific algorithm (pseudo spectral methods) to solve a problem. Because it is specialized it can easily handle all types of constraints for an optimal control problem such as different phases where the dynamics and constraints change, general path constraints, and specialized end constraints. This capability would be helpful given that a crewed entry has stricter constraints that any other entry problem and therefore requires a program that can handle strict constraints. However, the online capability of CasADi makes it especially beneficial to potential pre-entry replanning of the nominal trajectory.

6.1.2 Closed-loop Formulation

Overlaying the closed-loop optimizer demonstrated significant improvement on managing the conditional constraints (heat rate, g-load, dynamic pressure) necessary to the success of a crewed mission. The closed-loop solution also allowed for improvements on the trajectory design achieving a higher dip limit than what was attainable for the open-loop solution with the crewed mission design. This capability ultimately proves that the presented method is a viable approach for optimizing the reference trajectory for mission analysis and design.

6.2 iLQR Performance

While the iLQR controller does not present strict tracking of the open-loop constraints, tuning through the line-search parameter(±) in the backwards pass and cost function weights, Q and R, for the state and control error, respectively, were applied to constrict the terminal states and problem conditions (heat rate, dynamic pressure, g-loading) close if not within constrained bounds set by MSL mission. For a crewed mission, however, more work must be done to either reduce the maximum values or build up flight vehicle capability to address the concern. Additionally, work has been done by [47] on optimizing the weights of the cost function to more accurately represents the constraints of the problem. This method can be applied to future work for better tuning and ultimately better constraining the results. For each presented scenario, the iLQR controller shows improvement on constraining the problem conditions. While not all conditions are able to achieve values within their desired limitations, there is still evident improvement getting the maximum conditions and terminal states values closer to those constraints.

6.3 Reachable Set Analysis

The reachable sets derived for this approach had to be considerably constrained beyond the results produced from the MC simulations using tuning in the iLQR and rejection sampling. Given that certain deviations from the nominal entry conditions lead to crash landings, it is essential for future work on this approach to apply an online method for uncertainty quantification to ensure control is based around guiding the vehicle within a proper trajectory envelope. Work has already been done by [36, 28, 11, 14] to address this concern as it pertains to entry guidance.

Chapter 7

Conclusions

The computational program presented in this work builds a two-tired optimal control method with notable performance capability for a large-scale mission design and viable demonstration closed-loop optimal control for future entry guidance designs.

It is important to note that entry guidance of this scale will most likely require multiple stages to adequately address all the problem constraints. This approach does not take into account the pre-entry maneuvers that have been employed for previous Mars missions including Mars 2020 and MSL (TCM, cruise phase, etc.). Additionally, current investigation on potential phases to highly constrained entry scenarios on a crewed scale (cite) could address issues with adhering to strict terminal constraints in the closed-loop. As a result, this work can be extended to apply those considerations for improving on the presented results.

Overall, the extraordinary contributions to advance the Mars entry guidance problem emulates the critical need for developing a viable solution to this problem. Producing a robust and capable computational system promotes less reliance on the final stages of the EDL sequence to get the landing right, especially for missions of critical nature.

7.1 Future Work

Several elements to presented results create opportunity for future study to advance the capabilities initiated with this work. Reachable set analysis demonstrated a need for online uncertainty quantification incorporated as part of the closed-loop guidance. Because running Monte Carlo simulations is computationally expensive with long run times, especially for high-order problems, this method is only viable for offline planning. Other methods such as invariant funnels [28, 11, 36, 26] have been explored in robotics and applied to entry guidance to provide uncertainty bounds on trajectories in real time to further enhance guidance capabilities. Other methods such as tube MPC and polynomial chaos [14] have also been explored to better quantify and reduce the uncertainty for critical entry flight conditions.

Additionally, to better approximate potential disturbances due to environmental conditions, more work can be done to investigate Building uncertainty models within some of the aerodynamic (coefficient of lift and drag) and atmosphere (density model) constants as done with [10]. While the density model applied to this approach is a stronger approach than the basic density model given that it accounts for the temperature and pressure contributions at the provided altitude, there is still potential for error due to wind storms or other atmospheric conditions at the time of flight. Additionally, density levels are dynamic in that they change with the time of the year. Many approaches that look into navigating the Martian atmosphere use NASA's Mars-GRAM model to more accurately provide data on the available density based on the proposed time of entry [10, 29, 15].

While it is true that, as the first stage of the process, entry and the proper guidance thereof is critical to the success of subsequent missions and the EDL sequence at large, it is still important to consider advancement in the descent and landing phases. This will ultimately help to reduce the strain on the chosen entry guidance to complete the EDL objectives alone. Much work has been done to address the later stages of EDL to address this major concerns including capabilities in powered descent (retro-propulsion) and terrain relative navigation (autonomously select landing location and navigating around hazardous terrain). Additionally, range trigger methods have proven to reduce landing ellipses by 40 percent [32] to improve the landing accuracy required for a crewed mission and some proposed robotic missions. As a result, while a successful and robust entry method is necessary to supporting the later stages of EDL, it is not the only contribution to supporting the over successful landing of the spacecraft.

It is important for the reader to note that given the importance and critical nature of this problem with the nearing goals of NASA's Artemis program, many approaches have been developed, and continue to be developed, that may not be covered in this thesis. This work focuses on those methods most pertinent to the problem and chosen approach, but should not be regarded as a total abbreviation of the optimal control or entry guidance field. (not sure if this is good to add? should it go at the end of the conclusion instead?)

Appendix A

Supporting Figures

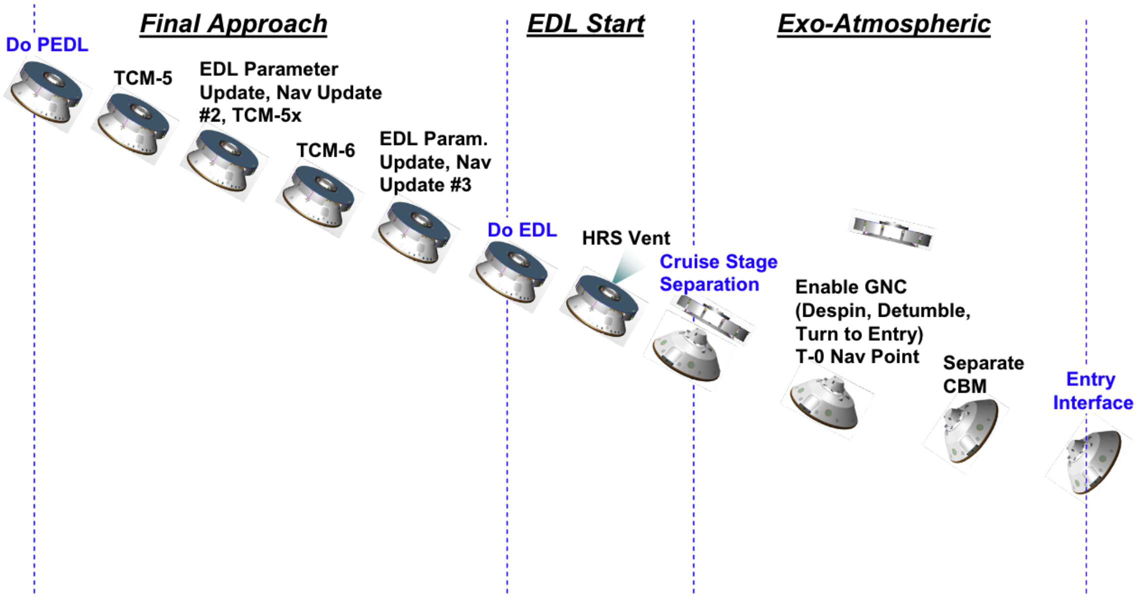
Final Approach
EDL Start
Exo-Atmospheric
Do PEDL
TCM-5
EDL Parameter Update, Nav Update #2, TCM-5x
TCM-6
EDL Param. Update, Nav Update #3
Do EDL
HRS Vent
Cruise Stage Separation
Enable GNC (Despin, Detumble, Turn to Entry) T-0 Nav Point
Separate CBM
Entry Interface

Figure A-1: MSL Entry, Landing, and Descent Event Sequence (1 of 3)

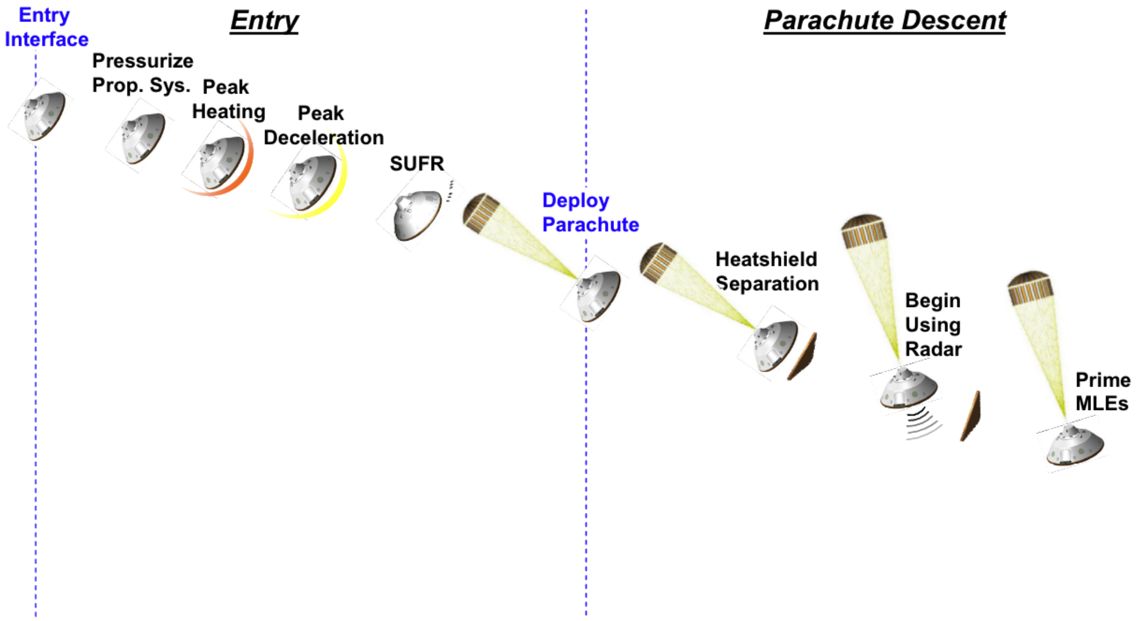

Figure A-2: MSL Entry, Landing, and Descent Event Sequence (2 of 3)

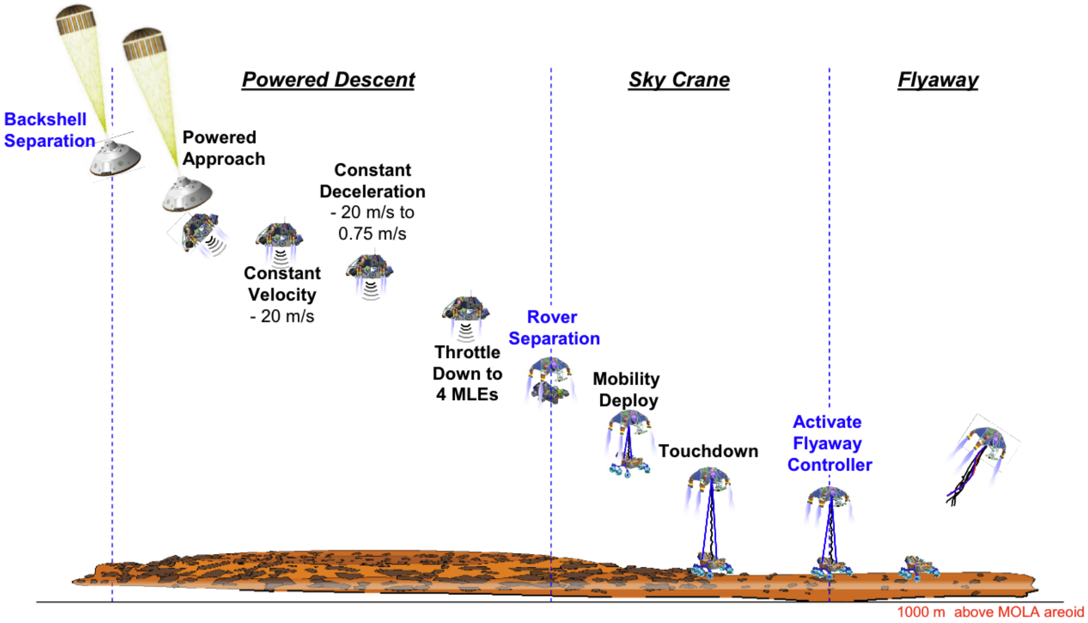

Figure A-3: MSL Entry, Landing, and Descent Event Sequence (3 of 3)

Bibliography

[1] John D. Anderson. *Hypersonic and High-Temperature Gas Dynamics*. American Institute of Aeronautics and Astronautics, Inc., 2019.

[2] Joel Andersson, Joris Gillis, and Moritz Diehl. User Documentation for CasADi v3.4.4. Technical report, 2018.

[3] Jonathan D. Aziz, Jeffrey S. Parker, Daniel J. Scheeres, and Jacob A. Englander. Low-Thrust Many-Revolution Trajectory Optimization via Differential Dynamic Programming and a Sundman Transformation. *Journal of the Astronautical Sciences*, 65(2):205–228, 6 2018.

[4] Sarah Bechtle, Yixin Lin, Akshara Rai, Ludovic Righetti, and Franziska Meier. Curious iLQR: Resolving Uncertainty in Model-based RL. Technical report.

[5] John T. Betts. Survey of numerical methods for trajectory optimization, 1998.

[6] Robert D. Braun and Robert M. Manning. Mars exploration entry, descent and landing challenges. In *IEEE Aerospace Conference Proceedings*, volume 2006, 2006.

[7] Gilbert L. Carman, Dallas G. Ives, and David K. Geller. Apollo-derived mars precision lander guidance. In *23rd Atmospheric Flight Mechanics Conference*, pages 1–18. American Institute of Aeronautics and Astronautics Inc, AIAA, 1998.

[8] Zilong Cheng, Jun Ma, Xiaoxue Zhang, Frank L. Lewis, and Tong Heng Lee. Neural Network iLQR: A Reinforcement Learning Architecture for Trajectory Optimization. 11 2020.

[9] Alicia Dwyer Cianciolo and Richard W Powell. Entry, Descent, and Landing Guidance and Control Approaches to Satisfy Mars Human Mission Landing Criteria. In *AAS/AIAA Space Flight Mechanics Meeting*, 2017.

[10] Connor David Noyes. Robust Optimal Entry Guidance for Future Mars Landers. Technical report, 2021.

[11] Remy Derollez, Zac Manchester, and Zachary Manchester. Sample-Based Robust Uncertainty Propagation For Entry Vehicles Motion Planning View project Robust Mars Entry Guidance View project (Preprint) AAS 20-071 SAMPLE-BASED ROBUST UNCERTAINTY PROPAGATION FOR ENTRY VEHICLES. Technical report.

[12] Monica Ekal, Keenan Albee, Brian Coltin, Rodrigo Ventura, Richard Linares, and David W Miller. Online Information-Aware Motion Planning with Inertial Parameter Learning for Robotic Free-Flyers. Technical report.

[13] Chen Ji, Minxiu Kong, and Ruifeng Li. Weight lifting trajectory optimization for variable stiffness actuated robot. *Complexity*, 2019, 2019.

[14] Xiuqiang Jiang and Shuang Li. Uncertainty quantification for mars atmospheric entry using polynomial chaos and spectral decomposition. In *AIAA Guidance, Navigation, and Control Conference, 2018*, number 210039. American Institute of Aeronautics and Astronautics Inc, AIAA, 1 2018.

[15] Minji Jo. *Autonomous and Integrated Guidance, Navigation, and Control System for Fuel-Optimal Atmospheric Entry, Descent, and Landing Maneuver.* PhD thesis, University of Hawai'i at Mãnoa, 5 2021.

[16] Jonathan Hui. RL - LQR & iLQR Linear Quadratic Regulator, 9 2018.

[17] Matthew Kelly. An introduction to trajectory optimization: How to do your own direct collocation. *SIAM Review*, 59(4):883–888, 2017.

[18] C. A. Kluever. Entry guidance performance for mars precision landing. In *Journal of Guidance, Control, and Dynamics*, volume 31, pages 1537–1544. American Institute of Aeronautics and Astronautics Inc., 2008.

[19] Vince Kurtz and Hai Lin. Contact-Implicit Trajectory Optimization with Hydroelastic Contact and iLQR. 2 2022.

[20] Steven M. LaValle. *Planning Algorithms.* Cambridge University Press, University of Illinois, 2006.

[21] Jean-Francois. Levesque. *Advanced navigation and guidance for high-precision planetary landing on Mars = Navigation et guidage avances pour un atterrissage planetaire a haute precision sur Mars.* [editeur non identifie], 2006.

[22] Weiwei Li and Emanuel Todorov. Iterative Linear Quadratic Regulator Design for Nonlinear Biological Movement Systems. Technical report.

[23] Ping Lu. Entry guidance: A unified Method. *Journal of Guidance, Control, and Dynamics*, 37(3):713–728, 2014.

[24] Wei-Min Lu and David S Bayard. GUIDANCE AND CONTROL FOR MARS ATMOSPHERIC ENTRY: ADAPTIVITY AND ROBUSTNESS. Technical report.

[25] David G. Luenberger and Yinyu Ye. *Linear and Nonlinear Programming*, volume 228. Springer Nature Switzerland AG, Cham, Switzerland, 5 edition, 2016.

[26] Anirudha Majumdar and Russ Tedrake. Funnel libraries for real-time robust feedback motion planning. *International Journal of Robotics Research*, 36(8):947–982, 7 2017.

[27] Kshitij Mall and Michael J. Grant. High mass mars exploration using slender entry vehicles. In *AIAA Atmospheric Flight Mechanics Conference*. American Institute of Aeronautics and Astronautics Inc, AIAA, 2016.

[28] Zachary Manchester and Scott Kuindersma. Robust direct trajectory optimization using approximate invariant funnels. *Autonomous Robots*, 43(2):375–387, 2 2019.

[29] Joel Benito Manrique. *Advances in Spacecraft Atmospheric Entry Guidance*. PhD thesis, University of California, Irvine, 2010.

[30] Gavin F. Mendeck and Lynn Craig McGrew. Entry Guidance Design and Post-flight Performance for 2011 Mars Science Laboratory Mission. *Journal of Spacecraft and Rockets*, 51(4):1094–1105, 2014.

[31] D. M. Murray and S. J. Yakowitz. Differential Dynamic Programming and Newton's Method for Discrete Optimal Control Problems I. Technical Report 3, 1984.

[32] Adam Nelessen, Chloe Sackier, Paul Brugarolas, Gregorio Villar, Allen Chen, Aaron Stehura, Richard Otero, Erisa Stilley, David Way, Karl Edquist, Swati Mohan, Cj Giovingo, and Mallory Lefland. Mars 2020 Entry, Descent, and Landing Overview. In *2019 IEEE Aerospace Conference*, Pasadena, 2019.

[33] Emily Palmer and Anil V. Rao. Mars Entry Optimal Trajectory Generation, Guidance, and Control. In *AIAA Science and Technology Forum and Exposition, AIAA SciTech Forum 2022*. American Institute of Aeronautics and Astronautics Inc, AIAA, 2022.

[34] Yanjun Pan, Qin Lin, Het Shah, and John M. Dolan. Safe planning for self-driving via adaptive constrained ILQR. In *IEEE International Conference on Intelligent Robots and Systems*, pages 2377–2383. Institute of Electrical and Electronics Engineers Inc., 10 2020.

[35] Ravi Prakash, P Dan Burkhart, Allen Chen, Keith A Comeaux, Carl S Guernsey, Devin M Kipp, Leila V Lorenzoni, Gavin F Mendeck, Richard W Powell, Tommaso P Rivellini, A Miguel, San Martin, Steven W Sell, Adam D Steltzner, and David W Way. Mars Science Laboratory Entry, Descent, and Landing System Overview. Technical report, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, 2008.

[36] Taylor P. Reynolds, Danylo Malyuta, Mehran Mesbahi, Behçet Açıkmeşe, and John M. Carson. Funnel synthesis for the 6-dof powered descent guidance problem. In *AIAA Scitech 2021 Forum*, pages 1–21. American Institute of Aeronautics and Astronautics Inc, AIAA, 2021.

[37] Jack Ridderhof, Panagiotis Tsiotras, and Breanna J. Johnson. Stochastic Entry Guidance. *Journal of Guidance, Control, and Dynamics*, 45(2):320–334, 2022.

[38] Wajdi Sadik, Smyye Rh, and Joao P. Hespanha. Undergraduate Lecture Notes on LQG/LQR controller design Lect ure not es on LQR/LQG cont roller design Cont ent s. Technical report, 2007.

[39] Yuval Tassa, Tom Erez, and Emmanuel Todorov. Synthesis and Stabilization of Complex Behaviors through Online Trajectory Optimization. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012.

[40] Yuval Tassa, Nicolas Mansard, and Emo Todorov. Control-Limited Differential Dynamic Programming. Technical report.

[41] Russ Tedrake. Trajectory Optimization. In *Underactuated Robotics: Algorithms for Walking, Running, Swimming, Flying, and Manipulation*, chapter 10. Course Notes from MIT 6.832, Cambridge, 4 2022.

[42] Emanuel Todorov. General duality between optimal control and estimation. In *Proceedings of the IEEE Conference on Decision and Control*, pages 4286–4292, 2008.

[43] Emanuel Todorov and Weiwei Li. A generalized iterative LQG method for locally-optimal feedback control of constrained nonlinear stochastic systems. In *43rd IEEE Conference on Decision and Control*, 2004.

[44] M B Tumbiwro. KIRTIAND AFB. N MFY GUIDANCE, FLIGHT MECHANICS AND TRAJECTORY OPTIMIZATION Volume XIV-Entry Guidance Equations. Technical report.

[45] Yi Wang, Xiaoqian Chen, Dechao Ran, Yong Zhao, Yang Chen, and Yuzhu Bai. Spacecraft formation reconfiguration with multi-obstacle avoidance under navigation and control uncertainties using adaptive artificial potential function method. *Astrodynamics*, 4(1):41–56, 3 2020.

[46] David W Way Richard W Powell Allen Chen, Adam D Steltzner, A Miguel San Martin, P Daniel Burkhart, and Gavin F Mendeck. Mars Science Laboratory: Entry, Descent, and Landing System Performance. Technical report.

[47] Wang Yi, Chen Xiaoqian, Bai Yuzhu, Cao Lu, and Zhu Xiaozhou. Motion planning of spacecraft with obstacle avoidance under low uncertainty using the improved equal-collision-probability-curve and improved linear quadratic regulator strategy. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 233(14):5456–5467, 11 2019.