Differentiable Convex Modeling in Robotic Planning and Control

Kevin Sledge Tracy

CMU-TR-XX-XXX

May 2024

The Robotics Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee

Zachary Manchester, Chair	*Carnegie Mellon University*
J. Zico Kolter	*Carnegie Mellon University*
Changliu Liu	*Carnegie Mellon University*
Tom Erez	*Google DeepMind Robotics*

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Keywords: motion planning, trajectory optimization, convex optimization, collision detection, guidance, navigation, control

To all of the people that light up my life. ♡

Abstract

Domain-specific modeling priors and specialized components are becoming increasingly important to the machine learning field. These components integrate specialized knowledge that we have as humans into model. We argue in this thesis that optimization methods provide an expressive set of operations that should be part of the machine learning practitioner's modeling toolbox.

We present two foundational approaches for optimization-based modeling: 1) the *OptNet* architecture that integrates optimization problems as individual layers in larger end-to-end trainable deep networks, and 2) the *input-convex neural network (ICNN)* architecture that helps make inference and learning in deep energy-based models and structured prediction more tractable.

We then show how to use the OptNet approach 1) as a way of combining model-free and model-based reinforcement learning and 2) for top-k learning problems. We conclude by showing how to differentiate cone programs and turn the `cvxpy` domain specific language into a differentiable optimization layer that enables rapid prototyping of the approaches in this thesis.

The source code for this thesis document is available in open source form at:
https://github.com/bamos/thesis

Acknowledgments

My ack

Contents

1 Introduction **1**

2 Preliminaries and Background **3**
2.1 Preliminaries . . . 3

3 Atmospheric Entry Guidance **5**
3.1 Introduction . . . 5
3.2 Entry Vehicle Dynamics . . . 7
3.2.1 The Vinh Model . . . 7
3.2.2 Cartesian Entry Dynamics . . . 8
3.2.3 State and Control Definitions . . . 10
3.3 Trajectory Optimization . . . 11
3.3.1 Safety Constraints . . . 11
3.3.2 Full Nonconvex Formulation . . . 11
3.4 Convex Predictor-corrector . . . 12
3.4.1 Prediction and Dynamics Linearization . . . 12
3.4.2 Cost Function . . . 13
3.4.3 Constraints . . . 14
3.4.4 Trust Region . . . 15
3.4.5 Convex Corrector Problem . . . 15
3.4.6 CPEG Algorithm . . . 16
3.5 Numerical Experiments . . . 16
3.5.1 Bank-Angle Control . . . 18
3.5.2 Bank-Angle and Angle-of-Attack Control . . . 18
3.6 Conclusion . . . 21

4 Robust Entry Guidance with Atmospheric Adaptation **23**

Bibliography **25**

List of Figures

3.1 The E frame is fixed to the entry vehicle, with $\hat{e}_1$ in the direction of the specific angular momentum vector, and $\hat{e}_2 = \hat{v} \times \hat{e}_1$. When defined in this frame, the lift vector can be expressed using only $\hat{e}_1$ and $\hat{e}_2$. 9

3.2 Altitude and downrange distance from the converged trajectories from CPEG on the three specified cases. The σ_{L1} case is with bank-angle control and an L1 penalty on bank-angle derivative, σ_{quad} is bank-angle only with a quadratic penalty on bank-angle derivative, and $\sigma + \alpha$ is control over both bank-angle and angle-of-attack. Due to the differences in control authority and cost function, all three converge on different trajectories that hit the target position at parachute deployment. 17

3.3 Crossrange and downrange trajectory data from the converged trajectories from CPEG on the three specified cases. The cases with only control over the bank-angle have to do a bank reversal to hit the target, whereas the case with control over bank-angle and angle-of-attack is able to leverage the full lift control to avoid the switching. 17

3.4 Predicted entry vehicle trajectories for the bank-angle only L1 penalty case, as seen by the altitude and downrange data. As the iterates continue, the entry vehicle converges on a trajectory that reaches the target at the 10km altitude mark. 18

3.5 Predicted entry vehicle trajectories for the bank-angle only L1 penalty case, as seen by the crossrange and downrange data. 19

3.6 Predicted entry vehicle trajectories for the bank-angle only quadratic penalty case, as seen by the altitude and downrange data. The 19

3.7 Predicted entry vehicle trajectories for the bank-angle only quadratic penalty case, as seen by the crossrange and downrange data. 20

3.8 Bank-angle only control plans for both the L1 and quadratic cost cases. The L1 cost motivated a bang-bang switching style bank-angle profile. The quadratic cost resulted in a smooth and continuous bank-angle profile. . . 20

3.9 Predicted entry vehicle trajectories for the bank-angle and angle-of-attack case, as seen by the altitude and downrange data. 21

3.10 Predicted entry vehicle trajectories for the bank-angle and angle-of-attack case, as seen by the crossrange and downrange data. 22

3.11 Bank-angle and angle-of-attack profiles for the case where both angles are being controlled. CPEG was able to converge on this control plan in just three iterations. 22

List of Tables

List of Algorithms

1 CPEG Algorithm 16

Chapter 1

Introduction

Chapter 2

Preliminaries and Background

This section provides a broad overview of foundational ideas and background material relevant to this thesis. In most chapters of this thesis, we include a deeper discussion of the related literature relevant to that material.

2.1 Preliminaries

Chapter 3

Atmospheric Entry Guidance

As scientific and crewed payloads have more demanding goals, precise atmospheric entry guidance is playing an increasing role in mission success. State-of-the-art entry guidance algorithms are structured in a predictor-corrector framework, where a simulation is used to predict a trajectory, and corrections are then made to the control inputs. These guidance methods are simple and effective, but current algorithms assume low lift-to-drag entry vehicles, are limited to only bank-angle control, and have a limited ability to guarantee the safety of the vehicle. We propose a new predictor-corrector entry guidance method that formulates the correction step as a convex optimization problem. This allows for more flexibility in specifying the vehicle's dynamics and control inputs, and the ability to explicitly handle safety constraints such as heating, pressure, and acceleration limits. We test the new algorithm in Mars entry scenarios similar to the Mars Science Laboratory with both bank-angle control and bank-angle plus angle-of-attack control, demonstrating both its performance and ability to generalize to future vehicle capabilities.

The contents of this chapter have been previously published at IEEE Aerospace Conference 2021 in Tracy and Manchester [26]

3.1 Introduction

In 1971 the Soviet Union's Mars-2 spacecraft made history by entering the Martian atmosphere before impacting the surface. Nine days later, an identical Mars-3 spacecraft performed the first soft-landing on the Martian surface, ushering in a new era in planetary exploration. NASA followed with successful Mars landings in 1976 with Viking 1 and 2 and has since then landed and operated multiple robotic systems on the Martian surface [15].

Entry vehicle architectures can be divided into three broad categories [15]: 1) Ballistic entry is an uncontrolled descent with drag as the only force, 2) unguided ballistic-lifting entry has an uncontrolled non-zero lift force, and 3) guided ballistic-lifting entry has some control over the vehicle's lift vector. Controlled entry guidance allows for the prioritization of landing locations with scientific merit instead of just those that minimize risk to the

vehicle.

The Mars Science Laboratory (MSL) carrying the Curiosity rover touched down in 2012 as the first Mars entry vehicle with guided ballistic-lifting entry. MSL had control over the vehicle bank angle during entry, enabling control of the direction of the lift vector within the lifting plane. While MSL dramatically reduced the size of the landing ellipse from over 100 km to 10 km, its guidance is still too coarse for pinpoint landings. By developing more performant entry guidance capabilities, entry vehicles could effectively place robotic or crewed landers in desirable science collection areas, including high altitude sites.

Much of the work on guidance for low lift-to-drag entry vehicles originated with the Apollo terminal guidance methods. These algorithms, as described in [12], rely on control of the bank angle with simple switching manuevers to control the cross-range and down-range errors. Slightly modified versions have been developed for use with more recent Mars entry vehicles, such as in [20]. Current research investigates the use of predictor-corrector algorithms [5] to improve the landing accuracy. A popular predictor-corrector formulation that exhibits bank-angle switching behavior is the Fully Numerical Predictor-corrector Entry Guidance (FNPEG) algorithm [17]. In the baseline FNPEG algorithm, Newton's method is used to solve for a static bank-angle that satisfies a terminal downrange distance constraint, and the sign of the bank-angle is modulated to control crossrange errors [16]. Here, the prediction phase is used to generate gradients for the terminal constraint, and corrections are applied to the open-loop commanded bank-angle in an effort to satisfy these terminal constraints. This framework is simple and effective but requires significant added complexity for incorporation of safety constraints or changes to the vehicle control inputs.

Trajectory optimization for offline planning of entry vehicle trajectories has been explored in [31] and [33], where the nonconvex optimal control problem was solved by linearizing the nonlinear dynamics and constraints, solving a conic optimization problem with a trust region, and repeating until convergence. This successive-convexification method was used instead of standard NonLinear Programming (NLP) solvers, like SNOPT [11] or IPOPT [30], because it is able to directly handle second-order cone constraints instead of relying on local linear approximations. Optimal trajectories computed offline were then paired with an optimization-based tracking controller, as described in [31] and [33]. While these formulations are able to stabilize a trajectory, there are no guarantees that safety constraints can be satisfied online. Also, the computational complexity of the trajectory-optimization formulation makes these methods intractable for real-time control onboard an entry vehicle.

The Convex Predictor-corrector Entry Guidance (CPEG) algorithm proposed in this paper combines ideas from trajectory optimization with the predictor-corrector guidance framework by solving a constrained optimization problem during the correction step. First, the dynamics of the entry vehicle with the current control plan are simulated to a target altitude for a predicted trajectory. Next, the vehicle dynamics are linearized about the predicted trajectory and a convex trajectory optimization problem is solved that minimizes landing error. By solving for a correction using convex optimization, CPEG is able to reason about the full state and control history to inform the correction instead of just the final state. This also allows for the vehicle's safety constraints, such as heating, pressure, and acceleration, to be explicitly included in the correction computation. Our specific

contributions in this paper are:

1. A general quasi-linear formulation of entry vehicle dynamics that is well-suited to numerical optimization.

2. A predictor-corrector entry guidance algorithm with a highly generalizable correction step utilizing convex optimization.

3. Customized trust regions and objective functions for entry vehicles with multiple control modalities.

The paper proceeds as follows: In Section 3.2, the classic Vinh entry vehicle dynamics are compared with a more modern Cartesian approach. In Section 3.3, the details of the full nonconvex trajectory optimization problem are discussed. In Section 3.4, the CPEG algorithm is derived. In Section 3.5, CPEG is validated on entry vehicles with bank-angle control, as well as bank-angle and angle-of-attack control. Finally, Section 3.6 outlines our conclusions and potential future research directions.

3.2 Entry Vehicle Dynamics

Despite much of the recent powered-descent guidance literature using Cartesian state representations, entry vehicles are still most often represented in spherical coordinates. In this section, the traditional entry vehicle dynamics denoted below as the "Vinh" model will be discussed, as well as an alternative Cartesian formulation.

3.2.1 The Vinh Model

The classic Vinh model, presented in 1976 in [6] and again a few years Later in Vinh's textbook [28], has been the standard method for simulating entry vehicles for the past 45 years. Parameterizing the entry vehicle in spherical coordinates, the state in the Vinh model contains familiar terms like latitude, longitude, and flight-path angle. Despite being highly nonlinear and prone to scaling issues, it is the most common dynamics model in the literature [6, 28, 29, 34, 32, 16, 9].

The dynamics in the Vinh model are calculated with the angle-of-attack, α, bank-angle, σ, flight-path angle, γ, longitude, θ, latitude, ϕ, and heading angle, ψ. The resulting

equations of motion over a planet that's rotating with a constant angular velocity Ω are,

$$\dot{r} = V \sin\gamma, \tag{3.1}$$
$$\dot{\theta} = V \cos\gamma \sin\psi/(r\cos\phi), \tag{3.2}$$
$$\dot{\phi} = V \cos\gamma \cos\psi/r, \tag{3.3}$$
$$\dot{V} = -D - \sin\gamma/r^2 + \Omega^2 r \cos\phi \sin\gamma \cos\phi - \Omega^2 r \cos\phi \cos\gamma \sin\phi \cos\psi, \tag{3.4}$$
$$\dot{\gamma} = L\cos\sigma/V + \left(V^2 - 1/r\right)\cos\gamma/(Vr) + 2\Omega\cos\phi\sin\psi + \Omega^2 r\cos\phi\cos\gamma\cos\phi/V \tag{3.5}$$
$$+ \Omega^2 r \cos\phi \sin\gamma \sin\phi \cos\psi/V, \tag{3.6}$$
$$\dot{\psi} = L\sin\sigma/(V\cos\gamma) + V\cos\gamma\sin\psi\tan\phi/r - 2\Omega(\tan\gamma\cos\psi\cos\phi - \sin\phi) \tag{3.7}$$
$$+ \Omega^2 r \sin\phi\cos\phi\sin\psi/(V\cos\gamma), \tag{3.8}$$

where r is the normalized radial distance from the center of the planet, V is the normalized planet-relative velocity, and L and D are the magnitudes of the lift and drag accelerations.

This model is highly nonlinear in both the state and the control, even when the planetary motion is ignored. While the planet's angular velocity is assumed to be constant, its inclusion in the dynamics still contributes significant nonlinearities. Because of this, much of the literature ignores the planet's angular velocity [31]. There are also scaling issues present if these equations are naively implemented. Since r and V are not angles, they are usually of a much larger magnitude than the rest of the state. This can lead to poor accuracy in variable time-step integrators, as well as ill-conditioning in numerical trajectory optimization.

3.2.2 Cartesian Entry Dynamics

We have found that entry vehicle dynamics are both simpler to derive and numerically better-conditioned when represented in standard Cartesian coordinates instead of the spherical coordinates used in the Vinh formulation. This state representation is popular with the powered-descent guidance community, albeit without any aerodynamic forces in the dynamics [4, 2, 1].

We assume a planet-fixed frame P is aligned with an inertial frame N along the z axis. The planet spins with angular velocity $\omega \in \mathbb{R}^3$ in the positive z direction, making the velocity of the entry vehicle the following:

$$^P v = {}^N v - \omega \times r, \tag{3.9}$$

where $^N v \in \mathbb{R}^3$ is the inertial velocity, $^P v \in \mathbb{R}^3$ is the planet relative velocity, and $r \in \mathbb{R}^3$ is the position of the entry vehicle in the planet frame. This expression can be differentiated once more to provide the relationship between the inertial and planet-relative accelerations:

$$^P a = {}^N a - 2(\omega \times {}^P v) - \omega \times (\omega \times r). \tag{3.10}$$

The state of the entry vehicle can be parameterized with the planet-relative position vector r, and planet relative velocity $^P v$ denoted as just v, both expressed in the coordinates of

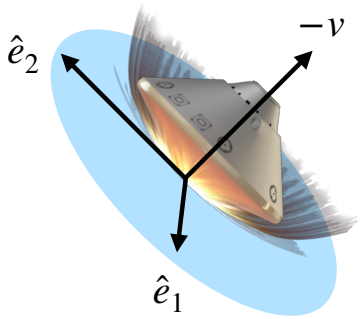

Figure 3.1: The E frame is fixed to the entry vehicle, with $\hat{e}_1$ in the direction of the specific angular momentum vector, and $\hat{e}_2 = \hat{v} \times \hat{e}_1$. When defined in this frame, the lift vector can be expressed using only $\hat{e}_1$ and $\hat{e}_2$.

the planet frame. The Cartesian dynamics can now be written in state space as,

$$\begin{bmatrix} v \\ a \end{bmatrix} = \begin{bmatrix} 0 & I \\ -[\omega\times]^2 & -2[\omega\times] \end{bmatrix} \begin{bmatrix} r \\ v \end{bmatrix} + \begin{bmatrix} 0 \\ a_g + a_D + a_L \end{bmatrix}, \tag{3.11}$$

where $[\omega\times]$ is the skew-symmetric cross product matrix,

$$[\omega\times] = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}. \tag{3.12}$$

One of the main benefits of the dynamics in equation (3.11) is the linear kinematics. This means that linear approximations of the relationship between position and velocity are exact, and the only nonlinearities present are in the accelerations. The gravitational acceleration in the direction of the planet's center is expressed assuming simple spherical gravity:

$$a_g = -\frac{\mu}{\|r\|^3} r, \tag{3.13}$$

where $\mu \in \mathbb{R}$ is the standard gravitational constant for the given planet. The acceleration caused by the drag force is in the direction opposing velocity, and is calculated as,

$$a_D = -\frac{1}{2m} \rho A C_d \|v\| v, \tag{3.14}$$

where $m \in \mathbb{R}$ is the mass of the entry vehicle, $\rho \in \mathbb{R}$ is the atmospheric density, $A \in \mathbb{R}$ is the aerodynamic reference area, and $C_d \in \mathbb{R}$ is the coefficient of drag. In this work, the atmospheric density $\rho \in \mathbb{R}$ will be represented by a piecewise exponential function [9].

For the description of the lift acceleration, a reference frame is defined that describes a plane about the entry vehicle that is orthogonal to the velocity vector. This two-dimensional frame, referred to as the E frame and depicted in Fig. 3.1, has two basis

vectors described by the following:

$$\hat{e}_1 = \frac{r \times v}{\|r \times v\|}, \tag{3.15}$$

$$\hat{e}_2 = \frac{v \times \hat{e}_1}{\|v \times \hat{e}_1\|}. \tag{3.16}$$

The magnitude of the lift vector is calculated as,

$$\|L\| = \frac{1}{2m} C_L \rho(r) A \|v\|^2, \tag{3.17}$$

where $C_L \in \mathbb{R}$ is the coefficient of lift. In the case where the entry vehicle only has control over the bank-angle, the resulting lift acceleration can be described by the magnitude of the lift and the bank-angle:

$$a_L = \|L\|(\sin(\sigma)\hat{e}_1 + \cos(\sigma)\hat{e}_2). \tag{3.18}$$

In the case where the entry vehicle can control both the angle-of-attack as well as the bank-angle, the lift vector can be written as,

$$a_L = \|L\|(\ell_1\hat{e}_1 + \ell_2\hat{e}_2), \tag{3.19}$$

subject to the constraint $||\ell_1^2 + \ell_2^2|| \leq 1$. Here the lift acceleration is a linear function of the control inputs, which is a key feature when this model is linearized in an optimization problem. Both the Vinh model and the Cartesian model are nonlinear, but the Cartesian model behaves significantly better under linearization, making it a far better candidate for trajectory optimization.

3.2.3 State and Control Definitions

In the case where only the bank-angle is controlled, the state is augmented with the bank-angle, and the sole control input is the derivative of this bank-angle with respect to time. This allows for cost functions that specify desired behavior for the derivative of the bank-angle, with the state and control as the following:

$$x = \begin{bmatrix} r^T & v^T & \sigma \end{bmatrix}, \tag{3.20}$$

$$u = \dot{\sigma}. \tag{3.21}$$

These dynamics are now in control-affine form with linear kinematics. For the case with actuation of both the bank angle and angle-of-attack, the state and control are the following:

$$x = \begin{bmatrix} r^T & v^T \end{bmatrix}^T, \tag{3.22}$$

$$u = \begin{bmatrix} \ell_1 & \ell_2 \end{bmatrix}^T, \tag{3.23}$$

where ℓ_1 and ℓ_2 were defined in (3.19).

3.3 Trajectory Optimization

Feedback control laws for entry vehicles suffer in performance due to the severe underactuation of the vehicle. This is, in part, due to the fact that an entry vehicle has very limited ability to speed up or slow down in the along-track direction. To deal with this, it makes more sense to solve the guidance problem with a holistic planning approach, one that can reason about this limited control authority and plan for it. Therefore, we pose this problem as a trajectory optimization problem, where a locally optimal state trajectory and control plan can be solved for numerically.

3.3.1 Safety Constraints

Three key vehicle safety constraints — heating, pressure, and acceleration — are most dependent on the atmospheric density. Unfortunately, this is also the part of the environment in which there is the largest amount of uncertainty. The atmospheric density is often only known to roughly within a factor of two, with even less known about the wind conditions [9].

The heating constraint has to do with the max allowable heat rate that the ablative heat shield can withstand [8]. This is measured in power per square centimeter, and it is expressed as the following:

$$\dot{Q} = k_q \sqrt{\rho} V^{3.15} \leq \dot{Q}_{max}. \tag{3.24}$$

This function is nonlinear but can be locally approximated with linear functions during the correction step. The next safety constraint is the maximum dynamic pressure on the entry vehicle, which is expressed as the following:

$$q = .5\rho V^2 \leq q_{max}. \tag{3.25}$$

The last safety constraint is the maximum allowable normal load, which is the total aerodynamic force on the entry vehicle. This is expressed as a norm of the lift and drag forces:

$$a = \sqrt{\|L\|^2 + \|D\|^2} \leq a_{max}. \tag{3.26}$$

3.3.2 Full Nonconvex Formulation

In order to formulate a convex correction problem, we first consider the full nonlinear nonconvex problem. First, the dynamics described in equation (3.11) are discretized with an explicit integrator like the classic fourth-order Runge-Kutta method [21], giving a discrete-time dynamics model of the form,

$$x_{k+1} = f(x_k, u_k, \Delta t_k). \tag{3.27}$$

No assumptions have been made about the control configuration in this dynamics model: it can account for either bank-angle-only or bank-angle plus angle-of-attack control. The

full nonlinear trajectory optimization problem has the form,

$$\begin{aligned} \underset{x,\,u,\,\Delta t}{\text{minimize}} \quad & \ell_N(x_N, u_N) + \sum_{k=1}^{N-1} \ell_k(x_k, u_k) \\ \text{subject to} \quad & x_{k+1} = f(x_k, u_k, \Delta t_k) \forall k, \\ & g_k(x_k, u_k) \leq 0 \qquad \forall k, \\ & \Delta t_{min} \leq \Delta t_k \leq \Delta t_{max} \; \forall k, \\ & x_N = x_{goal}, \end{aligned} \tag{3.28}$$

where safety constraints (3.24)—(3.26) are included in the inequality constraint function $g_k(x_k, u_k)$. Note that this is a free-final-time problem in which the Δt_k are decision variables in addition to the states and controls. This is necessary due to the inability of the entry vehicle to reach its goal state at an arbitrarily specified time. Problem (3.28) is nonconvex due to both the nonlinear dynamics, as well as the variable time between knot points. It is worth noting that, even with linear continuous-time dynamics, the discrete-time dynamics constraints (3.27) become nonlinear when the time step is made to be a decision variable.

Trajectory optimization problems like (3.28) can be solved with a variety of methods. One standard approach is to use an off-the-shelf NLP solver like IPOPT [30] or SNOPT [11]. Alternatively, more specialized trajectory optimizers like ALTRO can be used [13, 14]. While computationally tractable using one of the described methods, the nonconvexity of the problem means there are no available guarantees for the quality of the solution or convergence of the solver. As a result, running nonconvex trajectory optimization onboard safety-critical aerospace systems is unpopular, explaining the prevalence of simpler heritage methods for entry guidance.

3.4 Convex Predictor-corrector

CPEG combines ideas from numerical trajectory optimization with the classic predictor-corrector guidance framework: It uses a prediction step, in which the vehicle dynamics are simulated until a target altitude is reached, combined with a corrector step that is based on solving a local convex approximation of a nonlinear trajectory optimization problem to steer the vehicle to the desired target. These steps are then repeated until convergence is achieved. This section provides a detailed derivation of the CPEG algorithm.

3.4.1 Prediction and Dynamics Linearization

In the first stage of CPEG, the dynamics of the entry vehicle are simulated with a standard Runge-Kutta method using the current nominal control trajectory, $\bar{U}$, until a target altitude is reached. We denote this predicted trajectory by $\bar{X}$. After the prediction step, the discrete-time nonlinear dynamics are approximated using a first-order Taylor series,

$$\bar{x}_{k+1} + \delta x_{k+1} \approx f(\bar{x}_k, \bar{u}_k) + A_k \delta x_k + B_k \delta u_k, \tag{3.29}$$

where A_k and B_k are the following Jacobians,

$$A_k = \left.\frac{\partial f(x_k, u_k, \Delta t_k)}{\partial x_k}\right|_{\bar{x}_k, \bar{u}_k}, \tag{3.30}$$

$$B_k = \left.\frac{\partial f(x_k, u_k, \Delta t_k)}{\partial u_k}\right|_{\bar{x}_k, \bar{u}_k}. \tag{3.31}$$

Subtracting the dynamics of the reference trajectory from both sides, the local linear dynamics of trajectory corrections can be written as:

$$\delta x_{k+1} = A_k \delta x_k + B_k \delta u_k. \tag{3.32}$$

A crucial distinction between CPEG and sequential convexification methods [31, 18, 19], is that trajectory iterates are always dynamically feasible, thanks to the prediction step. This eliminates the possibility of inconsistent linearizations of the dynamics constraints [23], in which no feasible correction trajectory exists. Specifically, there is always a trivial solution to (3.32) of all zeros for δx and δu.

3.4.2 Cost Function

The cost function used in CPEG is comprised of a term that penalizes the miss distance from the target and a term that penalizes specified control behaviors. For the penalty on miss distance, putting a naive quadratic cost on the error between the final position and the desired position is inappropriate since it also penalizes altitude errors. Instead, only the position error projected onto the landing plane is penalized, effectively ignoring altitude error. Since the altitude target is implicitly satisfied during the prediction step, this allows for the correction to only apply changes to the control plan that minimize the projected miss distance. The cost function for this projected miss distance is the following:

$$\ell_{miss}(\delta X, \delta U) = \|W(r_N + \delta r_N - r_{goal})\|_2^2, \tag{3.33}$$

where $r_N \in \mathbb{R}^3$ is the final position in the reference trajectory, $\delta r_N \in \mathbb{R}^3$ is the correction computed for this position, and $r_{goal} \in \mathbb{R}^3$ is the desired final position for parachute deployment. To project this error onto the landing plane, we define following projection matrix,

$$W = I - pp^T, \tag{3.34}$$

where p is the unit vector normal to the planetary surface at the target position:

$$p = \frac{r_{goal}}{\|r_{goal}\|}. \tag{3.35}$$

The second part of the cost function seeks to shape the control behavior. In the case of bank-angle control, we consider two different control cost functions that produce qualitatively different behavior:

$$\ell_{\sigma,L1}(\delta U) = \lambda \|\dot{\sigma_k}\|_1, \tag{3.36}$$

and

$$\ell_{\sigma,quad}(\delta U) = \lambda \dot{\sigma_k}^2, \tag{3.37}$$

where λ is a scalar tuning parameter. The first cost function (3.36) penalizes the L1 norm of the derivative of the bank-angle, resulting in bank-angle trajectories with a minimum number of discrete switches. The second cost function (3.37) penalizes the square of the bank-angle derivative, resulting in smooth bank-angle trajectories. For the bank-angle plus angle-of-attack case, as described in (3.19), we apply a simple quadratic cost to the norm of the controlled lift vector, effectively penalizing high angles of attack:

$$\ell_{\sigma\alpha}(\delta U) = \lambda \|u_k\|_2^2. \tag{3.38}$$

3.4.3 Constraints

Of the three nonlinear safety constraints, two can be linearized, and the third can be converted to a conservative convex relaxation. For the heating and dynamic pressure constraints (3.24)–(3.25), a Taylor expansion of each is formed, approximating the constraint to first-order. From here, a linearized inequality constraint can be directly included in the convex correction problem. For these constraints, the linearized versions are:

$$[\nabla \dot{Q}(\bar{x}_k)]^T \delta x_k \leq \dot{Q}_{max} - \dot{Q}(\bar{x}_k), \tag{3.39}$$

$$[\nabla q(\bar{x}_k)]^T \delta x_k \leq q_{max} - q(\bar{x}_k). \tag{3.40}$$

The acceleration loading constraint (3.26) is nonlinear, but a conservative convex relaxation can be derived in the form of a second-order cone constraint. First, the kinematics for the velocity can be conservatively approximated as the following:

$$v_{k+1} = v_k + a_k \Delta t, \tag{3.41}$$

$$a_k = \frac{v_{k+1} - v_k}{\Delta t}, \tag{3.42}$$

$$a_k = \frac{\bar{v}_{k+1} + \delta v_{k+1} - \bar{v}_k - \delta v_k}{\Delta t}. \tag{3.43}$$

The maximum loading constraint can then be re-written as,

$$\|\bar{v}_{k+1} + \delta v_{k+1} - \bar{v}_k - \delta v_k\| \leq \Delta t \cdot a_{max}, \tag{3.44}$$

which is in the form of a convex second-order cone, and can be directly incorporated into the correction problem.

The three safety constraints from equations (3.39), (3.40), and (3.44), are stacked into a generic safety constraint function,

$$g_{safety}(\delta x_k, \delta u_k) \leq 0. \tag{3.45}$$

3.4.4 Trust Region

To ensure that corrections are sufficiently small that the dynamics linearizations and constraint approximations remain accurate, a trust-region constraint is added to the convex correction problem. While standard trust-region methods apply norm constraints to δX and δU [23], insight into the entry guidance problem enables a more tailored approach.

The quality of the linearization presented in (3.32) is highly accurate for approximating the vehicle kinematics, gravity, and atmospheric drag, but is much less accurate when applied to the bank-angle in the bank-angle-only control case. Therefore, we design a trust region that restricts corrections to the bank-angle, $\delta\sigma_k$, given the known accuracy of small-angle approximations but allows large corrections to the other states. This approach also allows us to avoid the need to adapt trust regions inside the solver, enabling faster and more reliable convergence. We apply the following trust-region constraints to each corrector problem:

$$\|\delta u_k\|_2 \leq \delta u_{max} \tag{3.46}$$

$$|\delta\sigma_k| \leq \delta\sigma_{max} \tag{3.47}$$

3.4.5 Convex Corrector Problem

For the case where the entry vehicle has control of only the bank-angle as described in (3.18), the convex correction problem can be formulated as,

$$\begin{aligned} \underset{\delta X,\, \delta U}{\text{minimize}} \quad & \ell_{miss}(\delta X, \delta U) + \ell_{\sigma}(\delta U) \\ \text{subject to} \quad & A_k \delta x_k + B_k \delta u_k = \delta x_{k+1}, \\ & g_{safety}(\delta x_k, \delta u_k) \leq 0, \\ & \|\delta u_k\|_2 \leq \delta u_{max}, \\ & |\delta\sigma_k| \leq \delta\sigma_{max}, \end{aligned} \tag{3.48}$$

where the miss cost function is described in (3.33), and the bank-angle cost function can be either (3.36) or (3.37).

For the case where the entry vehicle has control over both bank-angle and angle-of-attack as described in (3.19), the convex correction problem can be posed as:

$$\begin{aligned} \underset{\delta X,\, \delta U}{\text{minimize}} \quad & \ell_{miss}(\delta X, \delta U) + \ell_{\sigma\alpha}(\delta U) \\ \text{subject to} \quad & A_k \delta x_k + B_k \delta u_k = \delta x_{k+1}, \\ & g_{safety}(\delta x_k, \delta u_k) \leq 0, \\ & \|u_k + \delta u_k\|_2 \leq 1. \end{aligned} \tag{3.49}$$

These problems can be solved quickly and reliably by standard conic solvers such as Mosek [22], COSMO [10], and ECOS [7].

3.4.6 CPEG Algorithm

The full CPEG algorithm is detailed in algorithm 1. The inputs to CPEG are the current position and the current control plan. From here, the dynamics of the entry vehicle are simulated until parachute deployment with the current control plan. This predicted trajectory is then discretized and linearized, resulting in dynamics Jacobians A_k and B_k (equations (3.30)-(3.31)). From here, the convex correction problem is posed given the control configuration and cost strategy. This convex optimization problem is solved, and the correction δU is used to correct the control plan. The prediction-correction steps are repeated until the norm of the correction being made to the control plan is below a specified tolerance.

Algorithm 1 CPEG Algorithm

input x_0, U ▷ nominal control plan
while $\|\delta U\| >$ tolerance **do**
    $\bar{X}, \bar{U} = \text{simulate}(x_0, U)$ ▷ predict trajectory
    $A, B = \text{linearize}(\bar{X}, \bar{U})$ ▷ linearize about prediction
    $\delta X, \delta U = \text{cvx}(\bar{X}, \bar{U}, A, B)$ ▷ solve for correction
    $U \mathrel{+}= \delta U$ ▷ correct control plan
end while
return U ▷ return updated control plan

3.5 Numerical Experiments

Parameters roughly matching those of the Mars Science Laboratory (MSL) [20] were used to test the CPEG algorithm. All scenarios begin at an altitude of 125 km above the Martian surface with a Mars-relative velocity of 5.845 km/second. CPEG was implemented in the Julia programming language [3], using the Convex.jl optimization modeling library [27], and the Mosek [22] and OSQP [24] solvers. CPEG was validated on the following three cases:

- Bank-angle control with L1 cost penalty, denoted σ_{L1}.
- Bank-angle control with quadratic penalty, denoted σ_2.
- Bank-angle plus angle-of-attack control, denoted $\sigma + \alpha$.

For the bank-angle cases, CPEG was arbitrarily initialized with a constant bank-angle of zero, with noise added to the bank-angle derivative. For the bank-angle plus angle-of-attack case, a similar approach was used, but noise was added to the normalized lift vector. The final converged trajectories for the three cases are shown in figures 3.2 and 3.3. In all of the cases, CPEG was able to successfully guide the entry vehicle to the target point at the desired altitude.

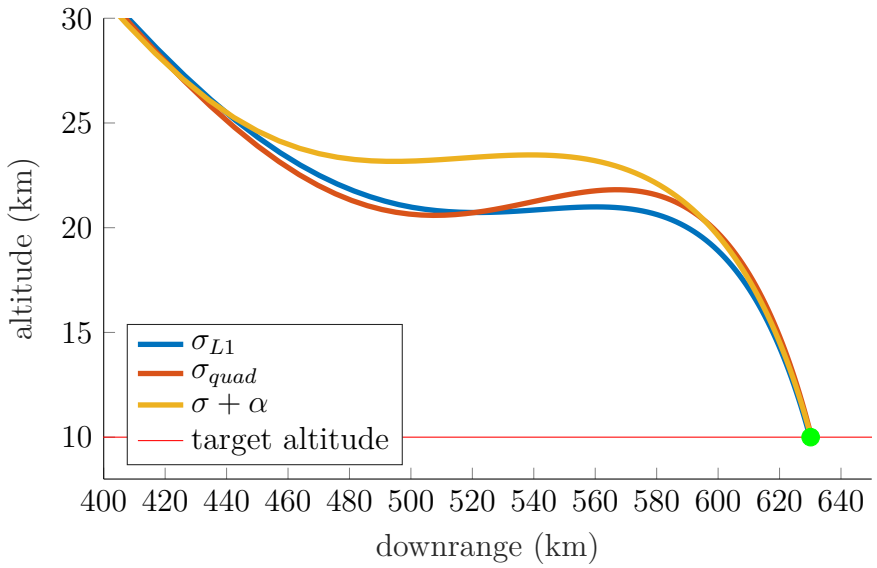

Figure 3.2: Altitude and downrange distance from the converged trajectories from CPEG on the three specified cases. The σ_{L1} case is with bank-angle control and an L1 penalty on bank-angle derivative, σ_{quad} is bank-angle only with a quadratic penalty on bank-angle derivative, and $\sigma + \alpha$ is control over both bank-angle and angle-of-attack. Due to the differences in control authority and cost function, all three converge on different trajectories that hit the target position at parachute deployment.

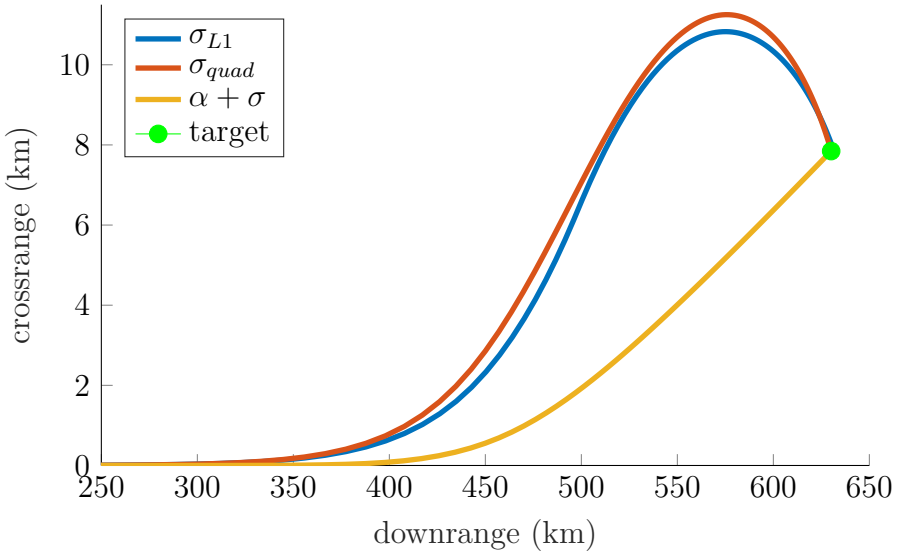

Figure 3.3: Crossrange and downrange trajectory data from the converged trajectories from CPEG on the three specified cases. The cases with only control over the bank-angle have to do a bank reversal to hit the target, whereas the case with control over bank-angle and angle-of-attack is able to leverage the full lift control to avoid the switching.

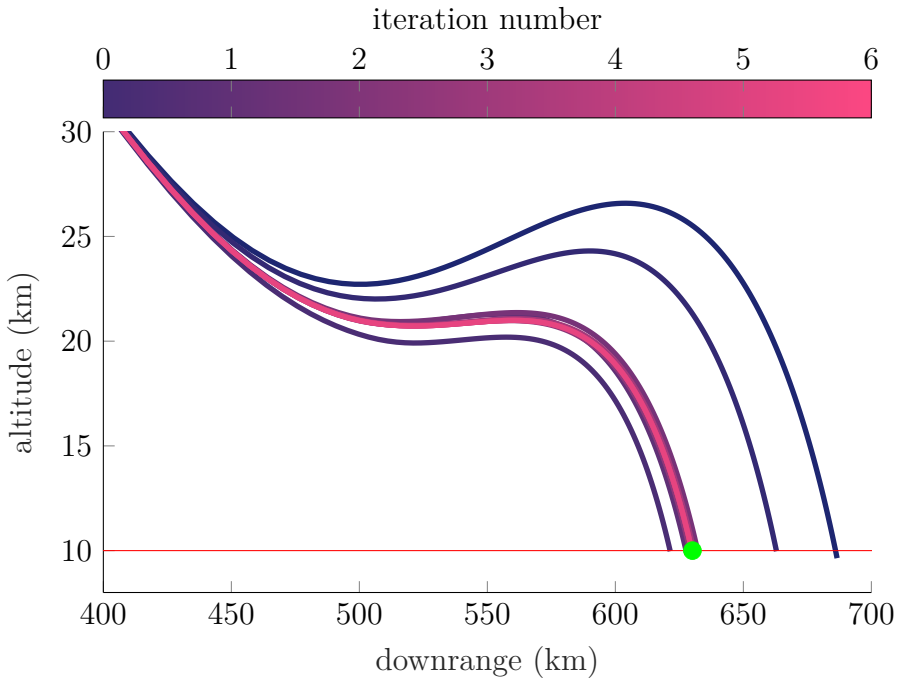

Figure 3.4: Predicted entry vehicle trajectories for the bank-angle only L1 penalty case, as seen by the altitude and downrange data. As the iterates continue, the entry vehicle converges on a trajectory that reaches the target at the 10km altitude mark.

3.5.1 Bank-Angle Control

For the case where the entry vehicle has only bank-angle control, the convergence of CPEG can be observed in figures 3.4 and 3.5 for the case with an L1 cost on the bank-angle derivative, and figures 3.6 and 3.7 with a quadratic cost. These plots show the output of the prediction step of CPEG, where the color of the predictions is blue for the first iteration of the algorithm and turns purple, then pink for later iterations. After convergence, the two bank-angle profiles that CPEG produced for the L1 and quadratic cost functions are shown in figure 3.8. The L1 cost on the derivative of the bank-angle encouraged sparsity in this derivative, resulting in a bank-angle profile that switches between constant bank-angles. For the case with a quadratic cost on the bank-angle derivative, the resulting bank-angle profile is smooth with no discrete switching behavior.

3.5.2 Bank-Angle and Angle-of-Attack Control

As described by the dynamics in equation (3.19), the control input for this case is the lift vector itself in the directions orthogonal to the velocity vector. This allows for manipulation of both the bank-angle and angle-of-attack and is guaranteed to be within the maximum allowable lift by the unit norm constraint in equation (3.49). In this control case, the control input Jacobian is constant and independent of the nominal control plan, making the linearization significantly more accurate than the bank-angle-only case. As a result, the

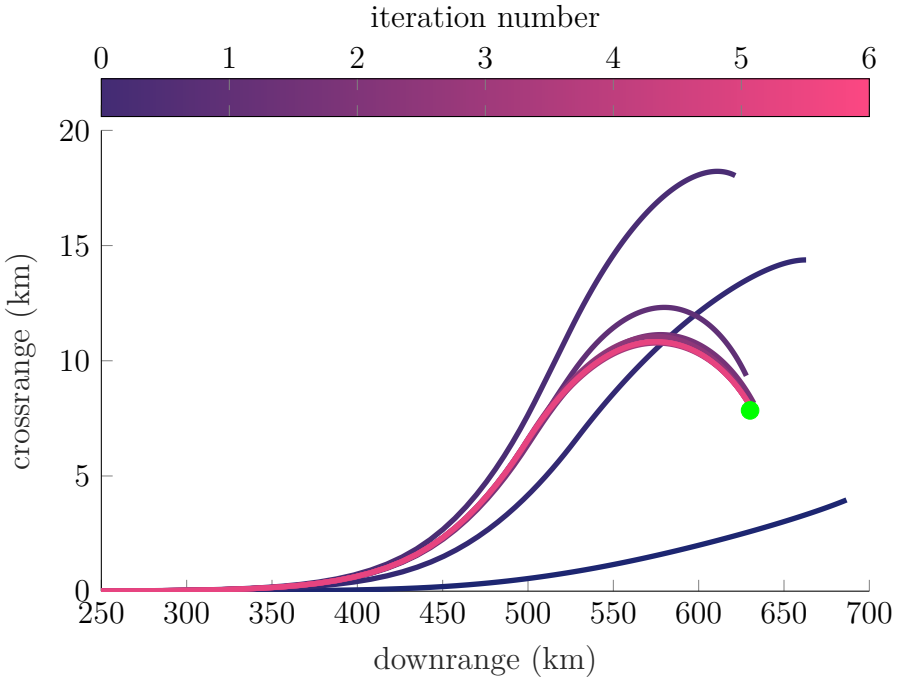

Figure 3.5: Predicted entry vehicle trajectories for the bank-angle only L1 penalty case, as seen by the crossrange and downrange data.

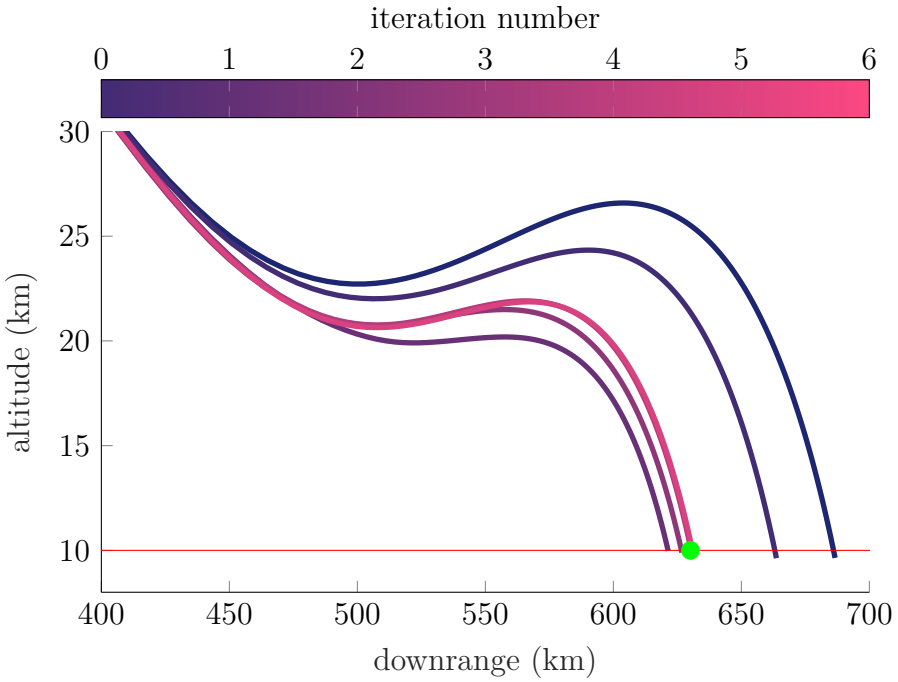

Figure 3.6: Predicted entry vehicle trajectories for the bank-angle only quadratic penalty case, as seen by the altitude and downrange data. The

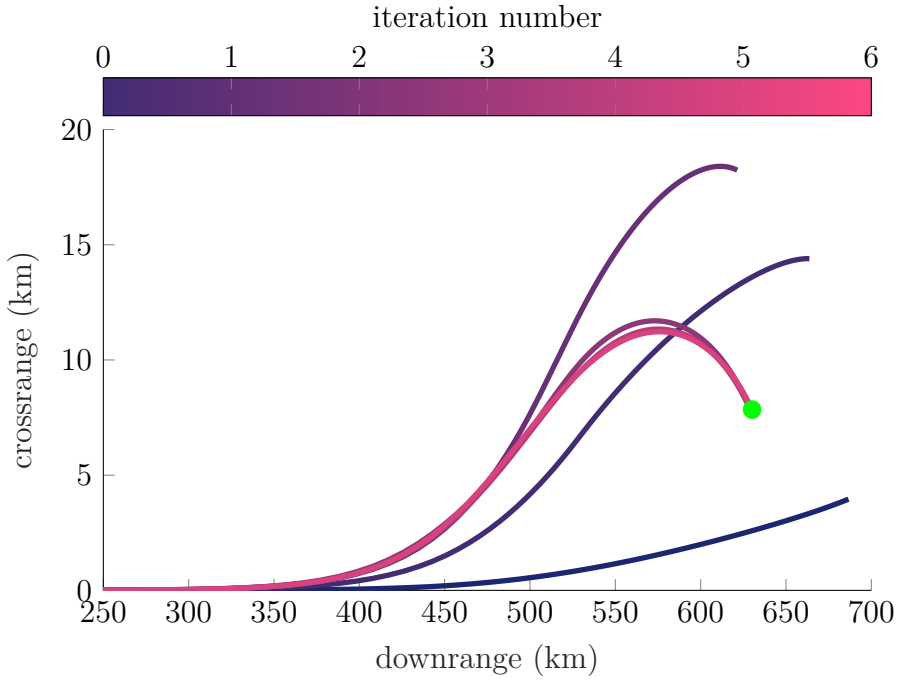

Figure 3.7: Predicted entry vehicle trajectories for the bank-angle only quadratic penalty case, as seen by the crossrange and downrange data.

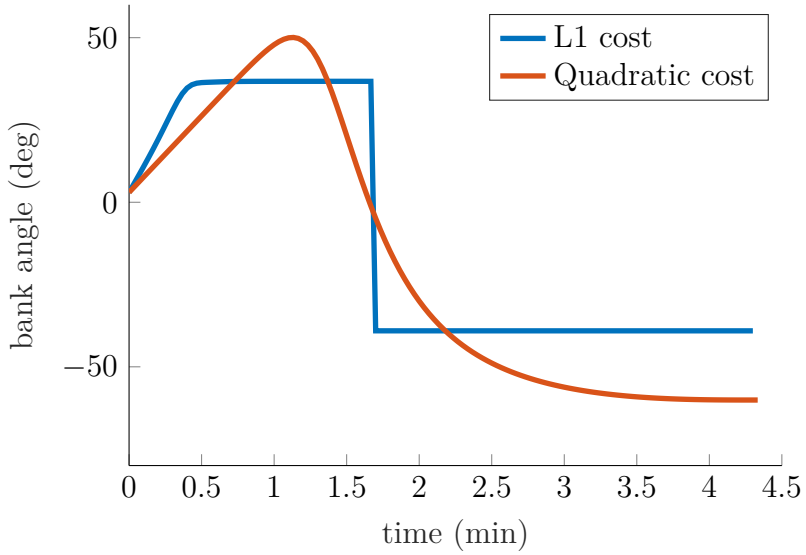

Figure 3.8: Bank-angle only control plans for both the L1 and quadratic cost cases. The L1 cost motivated a bang-bang switching style bank-angle profile. The quadratic cost resulted in a smooth and continuous bank-angle profile.

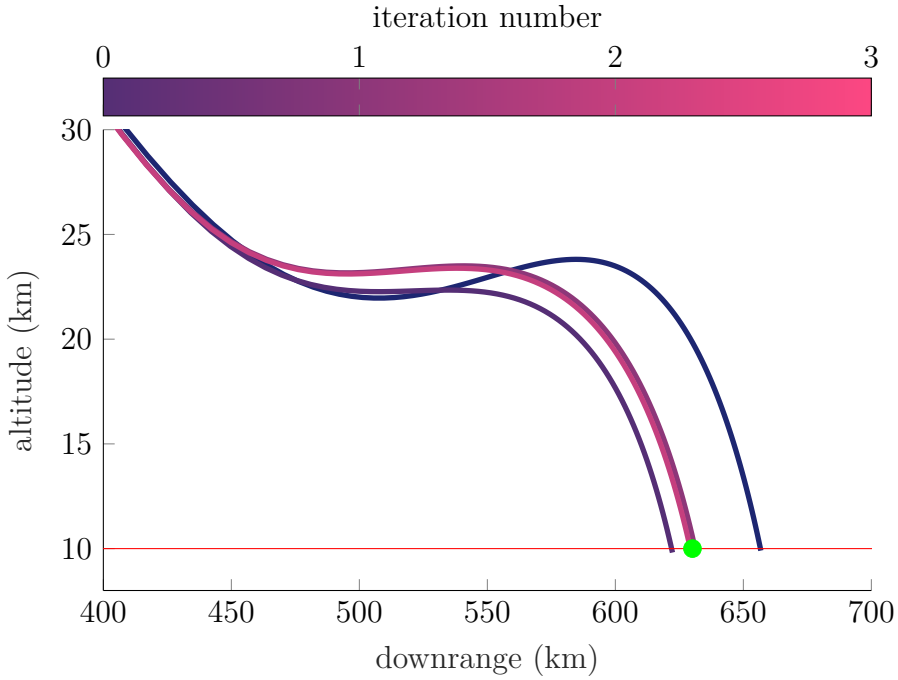

Figure 3.9: Predicted entry vehicle trajectories for the bank-angle and angle-of-attack case, as seen by the altitude and downrange data.

convergence of CPEG with bank-angle plus angle-of-attack control is significantly faster than with the bank-angle alone. The evolution of the predicted trajectories is shown in figures 3.9 and 3.10, with the same coloring scheme as the bank-angle only section. After convergence, the control inputs were converted back into bank-angle and angle-of-attack and shown together in figure 3.11.

3.6 Conclusion

This paper proposes an improved version of the classic predictor-corrector entry guidance scheme in which the correction step is formulated as a convex optimization problem. Two control strategies were tested with CPEG: bank-angle control, and bank-angle plus angle-of-attack modulation. For the bank-angle-only case, cost functions that penalized the derivative with an L1 cost and a quadratic cost were both demonstrated, resulting in dramatically different optimal bank-angle profiles. For the case with both bank-angle and angle-of-attack control, the quality of the dynamics linearization was accurate enough that CPEG was able to converge on an optimal trajectory in just a few iterations. An implementation of CPEG running all of the examples in this paper is available at `https://github.com/RoboticExplorationLab/EntryGuidance.jl`.

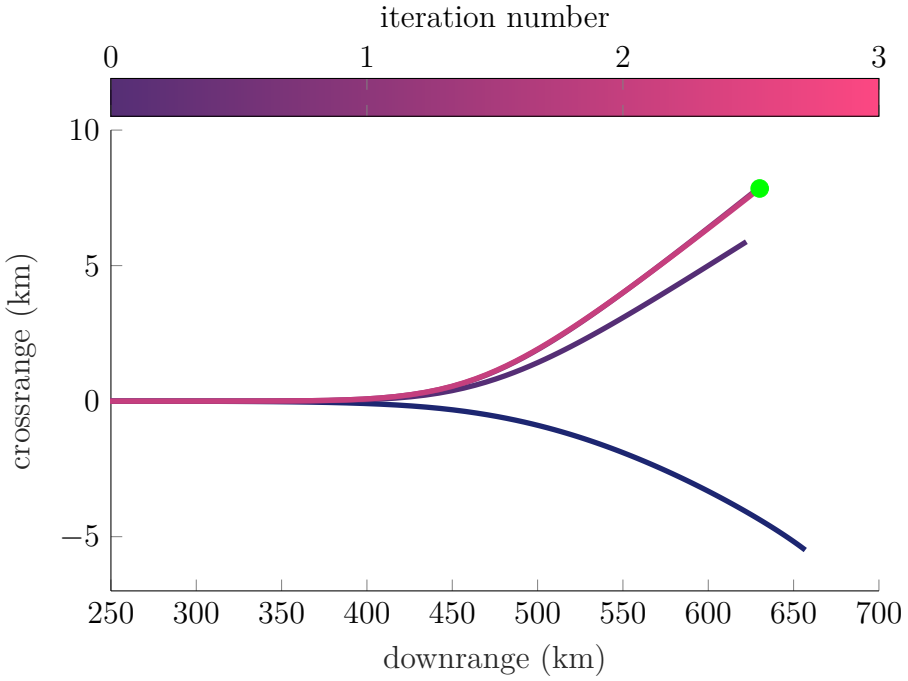

Figure 3.10: Predicted entry vehicle trajectories for the bank-angle and angle-of-attack case, as seen by the crossrange and downrange data.

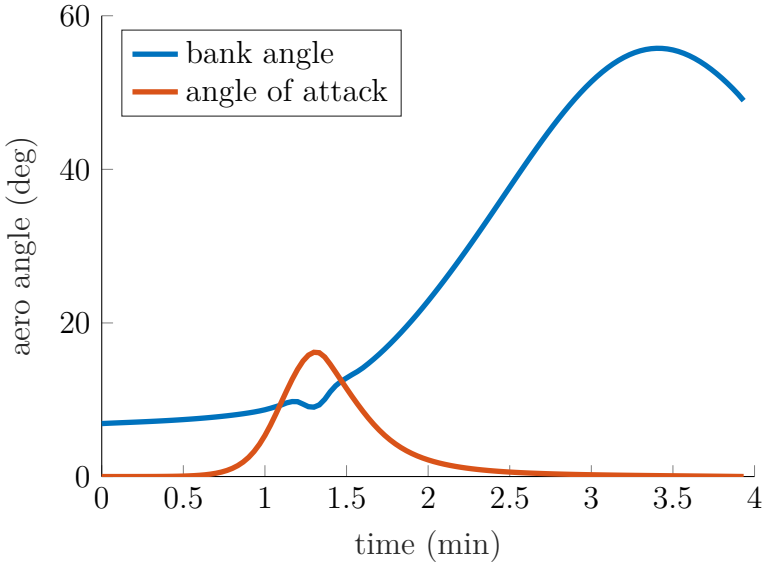

Figure 3.11: Bank-angle and angle-of-attack profiles for the case where both angles are being controlled. CPEG was able to converge on this control plan in just three iterations.

Chapter 4

Robust Entry Guidance with Atmospheric Adaptation

Robust atmospheric entry guidance for blunt-body entry vehicles with bank angle modulation is achieved by combining online atmospheric density estimation with an updated version of the Convex Predictor-corrector Entry Guidance (CPEG) algorithm. During atmospheric entry, a square-root Extended Kalman Filter is used to estimate a ratio between the density of the experienced atmosphere with that of an approximate model, which is spline-fit based on MarsGRAM perturbed data. The information from this filter is used to modify the approximate model used by the guidance algorithm. The proposed update to CPEG includes time as a decision variable, dramatically improving the robustness of the algorithm. CPEG predicts the trajectory at each control call with a nonlinear simulation followed by a single convex trajectory optimization problem that updates the commanded bank angle derivative. The robustness and performance of this estimator and controller guidance architecture are demonstrated on a wide range of realistic Martian atmospheres and is able to achieve state-of-the-art accuracy with respect to altitude-triggered parachute deployment.

The contents of this chapter have been previously published at IEEE Aerospace Conference 2021 in Tracy, Falcone, and Manchester [25]

Bibliography

This bibliography contains 34 references.

[1] Behcet Acikmese, John M. Carson, and Lars Blackmore. "Lossless Convexification of Nonconvex Control Bound and Pointing Constraints of the Soft Landing Optimal Control Problem". In: *IEEE Transactions on Control Systems Technology* 21.6 (2013), pp. 2104–2113. DOI: 10.1109/TCST.2012.2237346.

[2] Behcet Acikmese and Scott R. Ploen. "Convex Programming Approach to Powered Descent Guidance for Mars Landing". In: *Journal of Guidance, Control, and Dynamics* 30.5 (Sept. 2007), pp. 1353–1366. DOI: 10.2514/1.27553. (Visited on 10/27/2020).

[3] Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B. Shah. "Julia: A Fresh Approach to Numerical Computing". In: *SIAM Review* 59.1 (Jan. 2017), pp. 65–98. ISSN: 0036-1445, 1095-7200. DOI: 10.1137/141000671. (Visited on 01/30/2020).

[4] L. Blackmore, B. Açikmeşe, and J. M. Carson. "Lossless Convexification of Control Constraints for a Class of Nonlinear Optimal Control Problems". In: *2012 American Control Conference (ACC)*. June 2012, pp. 5519–5525. DOI: 10.1109/ACC.2012.6314722.

[5] Christopher W. Brunner and Ping Lu. "Skip Entry Trajectory Planning and Guidance". In: *Journal of Guidance, Control, and Dynamics* (May 2012). DOI: 10.2514/1.35055. (Visited on 05/12/2021).

[6] Adolf Busemann, Nguyen Vinh, and Robert Culp. *Hypersonic Flight Mechanics*. Tech. rep. NASA-CR-149170. Sept. 1976, p. 440.

[7] Alexander Domahidi, Manfred Morari, Manfred Morari, Stephen P Boyd, and Stephen P Boyd. "Methods and Tools for Embedded Optimization and Control". PhD thesis. Zürich: ETH, 2013. ISBN: 9783906031385.

[8] Karl T. Edquist, Brian R. Hollis, Artem A. Dyakonov, Bernard Laub, Michael J. Wright, Tomasso P. Rivellini, Eric M. Slimko, and William H. Willcockson. "Mars Science Laboratory Entry Capsule Aerothermodynamics and Thermal Protection System". In: *2007 IEEE Aerospace Conference*. Big Sky, MT, USA: IEEE, 2007, pp. 1–13. ISBN: 978-1-4244-0524-4. DOI: 10.1109/AERO.2007.352823. (Visited on 04/02/2021).

[9] Patrick Gallais. *Atmospheric Re-Entry Vehicle Mechanics*. Berlin ; New York: Springer, 2007. ISBN: 978-3-540-73646-2.

[10] Michael Garstka, Mark Cannon, and Paul Goulart. "COSMO: A Conic Operator Splitting Method for Convex Conic Problems". In: *arXiv:1901.10887 [math]* (Sept. 2020). arXiv: 1901.10887 [math]. (Visited on 10/19/2020).

[11] Philip E. Gill, Walter Murray, and Michael A. Saunders. "SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization". In: *SIAM Review* 47.1 (Jan. 2005), pp. 99–131. ISSN: 0036-1445, 1095-7200. DOI: 10.1137/S0036144504446096. (Visited on 06/04/2020).

[12] C. Graves and A. Harpold. *Apollo Experience Report: Mission Planning for Apollo Entry*. Technical Report NASA-TN-D-6725. Houston, TX: NASA Johnson Space Center, Mar. 1972. (Visited on 03/18/2018).

[13] Taylor A. Howell, Brian E. Jackson, and Zachary Manchester. "ALTRO: A Fast Solver for Constrained Trajectory Optimization". In: *IEEE International Conference on Intelligent Robots and Systems* (Nov. 2019), pp. 7674–7679. ISSN: 9781728140049. DOI: 10.1109/IROS40897.2019.8967788.

[14] Brian E. Jackson, Kevin Tracy, and Zachary Manchester. "Planning With Attitude". In: *IEEE Robotics and Automation Letters* (2021), pp. 1–1. ISSN: 2377-3766, 2377-3774. DOI: 10.1109/LRA.2021.3052431. (Visited on 08/25/2021).

[15] Shuang Li and Xiuqiang Jiang. "Review and Prospect of Guidance and Control for Mars Atmospheric Entry". In: *Progress in Aerospace Sciences* 69 (Aug. 2014), pp. 40–57. ISSN: 03760421. DOI: 10.1016/j.paerosci.2014.04.001. (Visited on 11/02/2019).

[16] Dongning Lu and Yiwu Liu. "Singular Formalism and Admissible Control of Spacecraft with Rotating Flexible Solar Array". In: *Chinese Journal of Aeronautics* 27.1 (Feb. 2014), pp. 136–144. ISSN: 10009361. DOI: 10.1016/j.cja.2013.12.010. (Visited on 05/29/2020).

[17] Ping Lu. "Predictor-Corrector Entry Guidance for Low-Lifting Vehicles". In: *Journal of Guidance, Control, and Dynamics* 31.4 (July 2008), pp. 1067–1075. ISSN: 0731-5090, 1533-3884. DOI: 10.2514/1.32055. (Visited on 11/02/2019).

[18] Danylo Malyuta, Taylor P. Reynolds, Michael Szmuk, Thomas Lew, Riccardo Bonalli, Marco Pavone, and Behcet Acikmese. "Convex Optimization for Trajectory Generation". In: *arXiv:2106.09125 [cs, eess, math]* (June 2021). DOI: 10.48550/arXiv.2106.09125. arXiv: 2106.09125 [cs, eess, math]. (Visited on 07/09/2021).

[19] Yuanqi Mao, Michael Szmuk, Xiangru Xu, and Behcet Acikmese. "Successive Convexification: A Superlinearly Convergent Algorithm for Non-convex Optimal Control Problems". In: *arXiv:1804.06539 [math]* (Feb. 2019). DOI: 10.48550/arXiv.1804.06539. arXiv: 1804.06539 [math]. (Visited on 02/09/2021).

[20] Gavin F. Mendeck and Lynn Craig McGrew. "Entry Guidance Design and Postflight Performance for 2011 Mars Science Laboratory Mission". In: *Journal of Spacecraft and Rockets* 51.4 (July 2014), pp. 1094–1105. ISSN: 0022-4650, 1533-6794. DOI: 10.2514/1.A32737. (Visited on 02/21/2018).

[21] O Montenbruck, E Gill, and Fh Lutze. "Satellite Orbits: Models, Methods, and Applications". In: *Applied Mechanics Reviews* 55.2 (2002), B27. ISSN: 00036900. DOI: 10.1115/1.1451162. (Visited on 12/15/2020).

[22] Mosek ApS. *The MOSEK Optimization Software*. Tech. rep. 2014.

[23] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. 2nd. Springer, 2006.

[24] Bartolomeo Stellato, Goran Banjac, Paul Goulart, Alberto Bemporad, and Stephen Boyd. "OSQP: An Operator Splitting Solver for Quadratic Programs". In: (), p. 40.

[25] Kevin Tracy, Giusy Falcone, and Zachary Manchester. "Robust Entry Guidance with Atmospheric Adaptation". In: *AIAA SciTech Forum and Exposition*. National Harbor, Maryland: AIAA, Jan. 2023.

[26] Kevin Tracy and Zachary Manchester. "CPEG: A Convex Predictor-corrector Entry Guidance Algorithm". In: *2022 IEEE Aerospace Conference (AERO)*. Big Sky, MT, USA: IEEE, Mar. 2022, pp. 1–10. ISBN: 978-1-66543-760-8. DOI: 10.1109/AERO53065.2022.9843641. (Visited on 09/13/2022).

[27] Madeleine Udell, Karanveer Mohan, David Zeng, Jenny Hong, Steven Diamond, and Stephen Boyd. "Convex Optimization in Julia". In: *2014 First Workshop for High Performance Technical Computing in Dynamic Languages*. LA, USA: IEEE, Nov. 2014, pp. 18–28. ISBN: 978-1-4799-7020-9. DOI: 10.1109/HPTCDL.2014.5. (Visited on 10/15/2021).

[28] N. X. Vinh, A. Busemann, and R. D. Culp. "Hypersonic and Planetary Entry Flight Mechanics". In: *NASA STI/Recon Technical Report A* 81 (Jan. 1980), p. 16245. (Visited on 10/15/2021).

[29] Nguyen Vinh, Wyatt Johnson, and James Longuski. "Mars Aerocapture Using Bank Modulation". In: *Astrodynamics Specialist Conference*. Denver,CO,U.S.A.: American Institute of Aeronautics and Astronautics, Aug. 2000. DOI: 10.2514/6.2000-4424. (Visited on 09/07/2021).

[30] Andreas Wächter and Lorenz T. Biegler. "On the Implementation of an Interior-Point Filter Line-Search Algorithm for Large-Scale Nonlinear Programming". In: *Mathematical Programming* 106.1 (Mar. 2006), pp. 25–57. ISSN: 0025-5610, 1436-4646. DOI: 10.1007/s10107-004-0559-y. (Visited on 02/22/2019).

[31] Zhenbo Wang and Michael J. Grant. "Constrained Trajectory Optimization for Planetary Entry via Sequential Convex Programming". In: *AIAA Atmospheric Flight Mechanics Conference*. Washington, D.C.: American Institute of Aeronautics and Astronautics, June 2016. ISBN: 978-1-62410-430-5. DOI: 10.2514/6.2016-3241. (Visited on 04/02/2021).

[32] Zhenbo Wang and Michael J. Grant. "Improved Sequential Convex Programming Algorithms for Entry Trajectory Optimization". In: *AIAA Scitech 2019 Forum*. San Diego, California: American Institute of Aeronautics and Astronautics, Jan. 2019. ISBN: 978-1-62410-578-4. DOI: 10.2514/6.2019-0667. (Visited on 04/21/2021).

[33] Zhenbo Wang and Michael J. Grant. "Near-Optimal Entry Guidance for Reference Trajectory Tracking via Convex Optimization". In: *2018 AIAA Atmospheric Flight Mechanics Conference*. Kissimmee, Florida: American Institute of Aeronautics and Astronautics, Jan. 2018. ISBN: 978-1-62410-525-8. DOI: 10.2514/6.2018-0013. (Visited on 04/02/2021).

[34] Zhenbo Wang and Michael J. Grant. "Near-Optimal Entry Guidance for Reference Trajectory Tracking via Convex Optimization". In: American Institute of Aeronautics

and Astronautics, Jan. 2018. ISBN: 978-1-62410-525-8. DOI: 10.2514/6.2018-0013. (Visited on 02/21/2018).