Successive Convexification for 6-DoF Mars Rocket Powered Landing with Free-Final-Time

Michael Szmuk and Behçet Açıkmeşe

University of Washington, Seattle, WA 98195-2400

In this paper, we employ successive convexification to solve the minimum-time 6-DoF rocket powered landing problem. The contribution of this paper is the development and demonstration of a free-final-time problem formulation that can be solved iteratively using a successive convexification framework. This paper is an extension of our previous work on the 3-DoF *free-final-time* and the 6-DoF *fixed-final-time* minimum-fuel problems. Herein, the vehicle is modeled as a 6-DoF rigid-body controlled by a single gimbaled rocket engine. The trajectory is subject to a variety of convex and non-convex state- and control-constraints, and aerodynamic effects are assumed negligible. The objective of the problem is to determine the optimal *thrust commands* that will minimize the time-of-flight while satisfying the aforementioned constraints. Solving this problem quickly and reliably is challenging because (a) it is nonlinear and non-convex, (b) the validity of the solution is heavily dependent on the accuracy of the discretization scheme, and (c) it can be difficult to select a suitable reference trajectory to initialize an iterative solution process. To deal with these issues, our algorithm (a) uses successive convexification to eliminate non-convexities, (b) computes the discrete linear-time-variant system matrices to ensure that the converged solution perfectly satisfies the original nonlinear dynamics, and (c) can be initialized with a simple, dynamically inconsistent reference trajectory. Using the proposed convex formulation and successive convexification framework, we are able to convert the original non-convex problem into a sequence of convex second-order cone programming (SOCP) sub-problems. Through the use of Interior Point Method (IPM) solvers, this sequence can be solved quickly and reliably, thus enabling higher fidelity real-time guidance for rocket powered landings on Mars.

Nomenclature

DoF	Degrees-of-Freedom	m	Vehicle mass
IPM	Interior Point Method	$\mathbf{r}_{\mathcal{I}}$	Inertial position of the vehicle
SC	Successive Convexification	$\mathbf{v}_{\mathcal{I}}$	Inertial velocity of the vehicle
SOCP	Second-Order Cone Programming	$q_{\mathcal{B}/\mathcal{I}}$	Unit quaternion
DCM	Direction cosine matrix	$\boldsymbol{\omega}_{\mathcal{B}}$	Angular velocity vector
$\mathbf{e}_i$	Unit vector pointing along i^{th}-axis	$\mathbf{F}_{\mathcal{I}}$	Force acting on the vehicle
$\mathcal{I}$	Subscript used to denote the inertial frame	$\mathbf{T}_{\mathcal{B}}$	Commanded thrust vector
$\mathcal{B}$	Subscript used to denote the body frame	$C_{\mathcal{B}/\mathcal{I}}$	DCM corresponding to $q_{\mathcal{B}/\mathcal{I}}$
$\mathcal{F}_{\mathcal{I}}$	The inertial NED frame	θ	Vehicle tilt angle
$\mathcal{F}_{\mathcal{B}}$	The body frame	δ	Gimbal angle
t	Time	$J_{\mathcal{B}}$	Moment of inertia of the vehicle
τ	Normalized trajectory time	γ_{gs}	Glide-slope cone constraint angle
t_f	Time-of-flight	θ_{max}	Maximum allowable tilt angle
m_{wet}	Dry mass of the landing vehicle	ω_{max}	Maximum allowable angular rate
m_{dry}	Wet mass of the landing vehicle	δ_{max}	Maximum allowable gimbal angle
$\mathbf{r}_{T,\mathcal{B}}$	Gimbal-point position vector		
$\mathbf{g}_{\mathcal{I}}$	Gravity vector expressed in $\mathcal{F}_{\mathcal{I}}$		

American Institute of Aeronautics and Astronautics

I.   Introduction

In this paper, we employ successive convexification to solve the minimum-time 6-DoF rocket powered landing problem. The work presented herein builds on our previous work on the 3-DoF free-final-time[1] and the 6-DoF fixed-final-time[2] minimum-fuel problems. The contribution of this paper is the development and demonstration of a 6-DoF free-final-time formulation that can be used within a successive convexification framework to solve the non-convex minimum-time 6-DoF rocket landing problem.

Autonomous optimal guidance technology that enables precision rocket powered landing is important for one primary reason: it provides a way to recover from a wider range of dispersions and uncertainties encountered during the entry, descent, and landing phase. This increases the probability that the vehicle will land safely and at a desirable location, and thus increases the likelihood of mission success. Such capabilities may be used to improve the scientific return of planetary science missions through the selection of more challenging and scientifically interesting landing sites, or by requiring less propellant mass to achieve the same level of landing robustness. This technology also enables the robust recovery of vertical-takeoff-vertical-landing (VTVL) launch vehicles,[3] thus allowing for dramatic launch cost reductions through the reuse of boosters. The importance of the latter point is highlighted by the efforts and triumphs of commercial space companies to land and reuse launch vehicles in recent years.

Solving autonomous optimal landing guidance problems quickly and reliably is challenging for three reasons. First, they are nonlinear and non-convex. Second, for numerical implementations, the validity of the solution is heavily dependent on the accuracy of the discretization scheme. Third, it can be difficult to select a suitable reference trajectory to initialize an iterative solution process.

Several insightful methods have been proposed[4–6] to analytically obtain sub-optimal guidance trajectory solutions. Although some methods successfully incorporated control constraints, attitude, and attitude-rate into the problem,[7] these analytical solutions could only handle a limited set of typical mission constraints.

In contrast, convex optimization has been considered a prime candidate for on-board autonomous guidance applications due to its deterministic behavior, its global optimality guarantees,[8] its ability to provide certificates of convergence and infeasibility, and the availability of efficient interior point method (IPM) algorithms[9–12] to solve convex problems. However, as mentioned before, the primary roadblock to employing convex optimization in this context is the non-convex nature of the problem.

Lossless convexification was one method proposed to circumvent the non-convex minimum thrust constraint of the landing problem.[13–20] This method was successfully demonstrated numerous times during flight experiments.[21,22] In a different instance, a dual-quaternion-based approach was proposed in which a 6-DoF line-of-sight constraint was convexified.[23,24] Since this method was based on dual-quaternions, it was inherently equipped to handle 6-DoF motion, but relied on piecewise-affine approximations of the nonlinear dynamics. As such, the solution degraded in accuracy as the temporal resolution of the discretization was reduced. Both of these methods were limited to fixed-final-time problems.

Although these convexification techniques proved to greatly simplify the solution process for certain non-convex problems, they remain applicable only to a narrow class of problems. On the other hand, successive convexification techniques offer a framework for solving more general non-convex optimal control problems, at the expense of more computational complexity and weakened optimality and convergence guarantees. These methods work by transforming the non-convex problem into a sequence of convex optimization problems that are solved in succession to convergence.[25–32]

In this paper, our strategy is to employ successive convexification to solve the minimum-time 6-DoF landing problem. First, we initialize the process using a simple, dynamically inconsistent reference trajectory. Then, we linearize and discretize the problem to generate a convex Second-Order Cone Programming (SOCP) sub-problem. We solve these sub-problems in succession, with each iteration linearizing around the previously generated solution. Lastly, we structure this process in a way that ensures the converged solution will satisfy the original nonlinear dynamics, kinematics, and constraints. We emphasize that our method does *not* perform a line search on the time-of-flight. Furthermore, in contrast to existing heuristic strategies, our method is capable of generating trajectories that are dynamically feasible, that adhere to the prescribed constraints, and that are more optimal, thus enlarging the usable flight envelope. Thus, we believe that the proposed method is a promising option for future autonomous applications.

The paper is organized as follows. In Section II, we state the continuous-time non-convex problem description. In Section III, we describe the successive convexification framework, and propose a method for converting the original problem into a sequence of discrete-time convex optimization problems. In Section IV, we present simulation of the proposed algorithm. Lastly, in Section V we provide concluding remarks.

American Institute of Aeronautics and Astronautics

II. Problem Description

In this section, we outline the 6-DoF minimum-time rocket powered landing problem in its original continuous-time non-convex form. This problem description is similar to its fixed-final-time minimum-fuel counterpart provided in our previous work,[2] with the notable exception that the final time of the trajectory is now an optimization variable. For completeness, we include the entirety of the problem description. In what ensues, we use t_f to denote the final time of the trajectory, and refer to t_f as the time-of-flight. Furthermore, we use the following two reference frames:

1. $\mathcal{F}_\mathcal{I}$: An inertially-fixed Up-East-North reference frame with its origin located at the landing site.

2. $\mathcal{F}_\mathcal{B}$: A body-fixed frame centered at the vehicle center-of-mass, with its X-axis pointing along the vertical axis of the vehicle (i.e. parallel to the thrust vector when the engine gimbal angle is zero), its Y-axis pointing out the side of the vehicle, and its Z-axis completing the right-handed system.

The remainder of this section covers the dynamics and kinematics that govern the problem, the state and control constraints imposed on the trajectory, and the boundary conditions enforced at the beginning and end of the trajectory. The section concludes with a concise summary of the problem description.

II.A. Dynamics and Kinematics

Herein, we treat the vehicle as a rigid body subject to constant gravitational acceleration, $\mathbf{g}_\mathcal{I} \in \mathbb{R}^3$, and negligible aerodynamic forces. The vehicle is assumed to actuate a single gimbaled rocket engine to generate a thrust vector within a feasible range of magnitudes and gimbal angles. Note that with minor modifications, this formulation can be adapted for applications that use multiple non-gimbaled rocket motors. Furthermore, we assume the vehicle depletes its mass, $m(t) \in \mathbb{R}_{++}$, at a rate proportional to the magnitude of the commanded thrust vector,[17] $\mathbf{T}_\mathcal{B}(t) \in \mathbb{R}^3$, which is expressed in $\mathcal{F}_\mathcal{B}$ coordinates. For tractability, we assume that the inertia tensor and the position of the center-of-mass are constant despite the depletion of mass. The proportionality constant, $\alpha_{\dot{m}}$, is given in terms of the vacuum-specific-impulse, I_{sp}, and Earth's standard gravity constant, g_0, as follows:

$$\alpha_{\dot{m}} \triangleq \frac{1}{I_{sp} g_0}$$

Thus, the mass depletion dynamics are given by:

$$\dot{m}(t) = -\alpha_{\dot{m}} \|TB(t)\|_2 \tag{1}$$

We express the position, velocity, and force acting on the vehicle in $\mathcal{F}_\mathcal{I}$ coordinates, and denote them by $\mathbf{r}_\mathcal{I}(t) \in \mathbb{R}^3$, $\mathbf{v}_\mathcal{I}(t) \in \mathbb{R}^3$, and $\mathbf{F}_\mathcal{I}(t) \in \mathbb{R}^3$, respectively. Thus, the translational dynamics are given by:

$$\dot{\mathbf{r}}_\mathcal{I}(t) = \mathbf{v}_\mathcal{I}(t) \tag{2a}$$

$$\dot{\mathbf{v}}_\mathcal{I}(t) = \frac{1}{m(t)} \mathbf{F}_\mathcal{I}(t) + \mathbf{g}_\mathcal{I} \tag{2b}$$

We use unit quaternions to parametrize the attitude of $\mathcal{F}_\mathcal{B}$ relative to $\mathcal{F}_\mathcal{I}$, and denote them by $q_{\mathcal{B}/\mathcal{I}}(t) \in \mathcal{S}^3$. In this paper, we use the leading scalar element convention for quaternions:

$$q_{\mathcal{B}/\mathcal{I}} \triangleq \begin{bmatrix} \cos(\xi/2) \\ \sin(\xi/2)\hat{\mathbf{n}} \end{bmatrix} = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}^T$$

The direction cosine matrix (DCM) that encodes the attitude transformation from $\mathcal{F}_\mathcal{I}$ to $\mathcal{F}_\mathcal{B}$ is denoted by $C_{\mathcal{B}/\mathcal{I}}(t) \in SO(3)$, where $C_{\mathcal{B}/\mathcal{I}}$ is related to $q_{\mathcal{B}/\mathcal{I}}$ through the following relation:

$$C_{\mathcal{B}/\mathcal{I}} = \begin{bmatrix} 1 - 2(q_2^2 + q_3^2) & 2(q_1 q_2 + q_0 q_3) & 2(q_1 q_3 - q_0 q_2) \\ 2(q_1 q_2 - q_0 q_3) & 1 - 2(q_1^2 + q_3^2) & 2(q_2 q_3 + q_0 q_1) \\ 2(q_1 q_3 + q_0 q_2) & 2(q_2 q_3 - q_0 q_1) & 1 - 2(q_1^2 + q_2^2) \end{bmatrix}$$

American Institute of Aeronautics and Astronautics

The inverse transformation (i.e. from $\mathcal{F}_\mathcal{B}$ to $\mathcal{F}_\mathcal{I}$) is denoted by $C_{\mathcal{I}/\mathcal{B}}(t) \triangleq C_{\mathcal{B}/\mathcal{I}}^{-1}(t) = C_{\mathcal{B}/\mathcal{I}}^T(t)$. We use $\boldsymbol{\omega}_\mathcal{B}(t) \in \mathbb{R}^3$ to denote the angular velocity vector of $\mathcal{F}_\mathcal{B}$ relative to $\mathcal{F}_\mathcal{I}$, expressed in $\mathcal{F}_\mathcal{B}$ coordinates, and, for some $\boldsymbol{\xi} \in \mathbb{R}^3$, define the skew-symmetric matrices $[\boldsymbol{\xi}\times]$ and $\Omega(\boldsymbol{\xi})$ as:

$$[\boldsymbol{\xi}\times] \triangleq \begin{bmatrix} 0 & -\xi_z & \xi_y \\ \xi_z & 0 & -\xi_x \\ -\xi_y & \xi_x & 0 \end{bmatrix} \qquad \Omega(\boldsymbol{\xi}) \triangleq \begin{bmatrix} 0 & -\xi_x & -\xi_y & -\xi_z \\ \xi_x & 0 & \xi_z & -\xi_y \\ \xi_y & -\xi_z & 0 & \xi_x \\ \xi_z & \xi_y & -\xi_x & 0 \end{bmatrix}$$

We denote the torque acting on the vehicle as $\mathbf{M}_\mathcal{B}(t) \in \mathbb{R}^3$, and the inertia tensor of the vehicle in $\mathcal{F}_\mathcal{B}$ coordinates as $J_\mathcal{B} \in \mathbb{S}_{++}^3$. Thus, the quaternion kinematics and attitude dynamics are given by:

$$\dot{q}_{\mathcal{B}/\mathcal{I}}(t) = \frac{1}{2}\Omega\big(\boldsymbol{\omega}_\mathcal{B}(t)\big)q_{\mathcal{B}/\mathcal{I}}(t) \tag{3a}$$

$$J_\mathcal{B}\dot{\boldsymbol{\omega}}_\mathcal{B}(t) = \mathbf{M}_\mathcal{B}(t) - [\boldsymbol{\omega}_\mathcal{B}(t)\times]\,J_\mathcal{B}\boldsymbol{\omega}_\mathcal{B}(t) \tag{3b}$$

Lastly, we must relate the force and torque vectors, $\mathbf{F}_\mathcal{I}(t)$ and $\mathbf{M}_\mathcal{B}(t)$, to the thrust vector, $\mathbf{T}_\mathcal{B}(t)$. The force is related to the thrust vector through the following relation:

$$\mathbf{F}_\mathcal{I}(t) = C_{\mathcal{I}/\mathcal{B}}(t)\mathbf{T}_\mathcal{B}(t) \tag{4}$$

Based on our earlier assumption that the center-of-mass does not move, it follows that the moment arm from the center-of-mass to the gimbal point of the engine is constant. We denote this constant position vector in $\mathcal{F}_\mathcal{B}$ coordinates as $\mathbf{r}_{T,\mathcal{B}} \in \mathbb{R}^3$. Thus, the torque is given by:

$$\mathbf{M}_\mathcal{B}(t) = [\mathbf{r}_{T,\mathcal{B}}\times]\,\mathbf{T}_\mathcal{B}(t) \tag{5}$$

II.B.   State and Control Constraints

To begin, we restrict the mass of the vehicle to remain above the dry mass, m_{dry}, using the following convex constraint:

$$m_{dry} \leq m(t) \tag{6}$$

The path of the vehicle is restricted to lie within an upward-facing glide-slope cone that makes an angle $\gamma_{gs} \in [0°, 90°)$ with the horizontal, and that is centered at the origin of $\mathcal{F}_\mathcal{I}$. This amounts to the convex constraint given by:

$$\mathbf{e}_1 \cdot \mathbf{r}_\mathcal{I}(t) \geq \tan\gamma_{gs}\|H_{23}^T\mathbf{r}_\mathcal{I}(t)\|_2 \tag{7a}$$

$$H_{23} \triangleq \begin{bmatrix} \mathbf{e}_2 & \mathbf{e}_3 \end{bmatrix} \tag{7b}$$

We define the tilt angle of the vehicle as the angle between the X-axes of $\mathcal{F}_\mathcal{B}$ and $\mathcal{F}_\mathcal{I}$, and denote it by $\theta(t)$. More formally, this is expressed as:

$$\cos\theta(t) = \mathbf{e}_1 \cdot C_{\mathcal{I}/\mathcal{B}}(t)\mathbf{e}_1 = 1 - 2\big(q_2^2(t) + q_3^2(t)\big) \tag{8}$$

Note that the quaternion elements in Eq. 8 are elements of $q_{\mathcal{B}/\mathcal{I}}(t)$. To avoid excessive tilt angles in the trajectory, we limit $\theta(t)$ to a maximum value of θ_{max}. If we immerse $q_{\mathcal{B}/\mathcal{I}}$ in $\mathbb{R}^4$, we can impose this limit through the following convex constraint:

$$\cos\theta_{max} \leq 1 - 2\big(q_2^2(t) + q_3^2(t)\big) \tag{9}$$

Furthermore, we limit the vehicle to a maximum angular rate of ω_{max} using the following convex constraint:

$$\|\boldsymbol{\omega}_\mathcal{B}(t)\|_2 \leq \omega_{max} \tag{10}$$

Lastly, the commanded thrust vector is constrained to lie within the magnitude interval $[T_{min}, T_{max}]$ and the gimbal angle interval $[0, \delta_{max}]$, where $\delta_{max} \in (0°, 90°)$:

$$0 < T_{min} \leq \|\mathbf{T}_\mathcal{B}(t)\|_2 \leq T_{max} \tag{11}$$

$$\cos\delta_{max}\|\mathbf{T}_\mathcal{B}(t)\|_2 \leq \mathbf{e}_1 \cdot \mathbf{T}_\mathcal{B}(t) \tag{12}$$

Note that the upper thrust magnitude bound is a convex constraint, whereas the lower thrust magnitude bound is non-convex. For the range of δ_{max} considered, the gimbal angle constraint is convex.

American Institute of Aeronautics and Astronautics

II.C. Boundary Conditions

Depending on the application, various combinations of boundary conditions may be enforced. In the scenario considered here, we enforce one common combination of boundary conditions, with the understanding that a different valid combination could be enforced with ease. First, we constrain the initial mass, position, velocity, and angular rate to m_{wet}, $\mathbf{r}_{\mathcal{I},i}$, $\mathbf{v}_{\mathcal{I},i}$, and $\boldsymbol{\omega}_{\mathcal{B},i}$, respectively, thus leaving the initial attitude free. We constrain the final position, velocity, attitude, and angular rate to $\mathbf{0}$, $\mathbf{v}_{\mathcal{I},f}$, $q_{\mathcal{B}/\mathcal{I},f}$, and $\mathbf{0}$, respectively, thus leaving the final mass free. Lastly, we leave the initial thrust vector unconstrained, while constraining the final thrust vector to point along the X-axis of $\mathcal{F}_{\mathcal{B}}$ in order to null out the torques at the final condition.

II.D. Problem Statement

We conclude by providing the complete problem description for the minimum-time 6-DoF rocket powered landing problem. We emphasize that in its original form, the problem is a non-convex continuous-time free-final-time problem. The objective is to minimize the time-of-flight subject to (a) the boundary conditions described in Section II.C, (b) the dynamics and kinematics in Eqs. 1-5, (c) the state constraints in Eqs. 6-7 and 9-10, and (d) the control constraints in Eqs. 11-12. The free variables are the commanded thrust vector, $\mathbf{T}_{\mathcal{B}}(t)$, and the time-of-flight, t_f. Problem 1 summarizes the complete non-convex problem description.

Problem 1. : *Non-Convex Continuous-Time Free-Final-Time Problem*

Cost Function:

$$\underset{t_f,\mathbf{T}_{\mathcal{B}}(t)}{\text{minimize}} \quad t_f$$

subject to:

Boundary Conditions:

$$m(0) = m_{wet}$$
$$\mathbf{r}_{\mathcal{I}}(0) = \mathbf{r}_{\mathcal{I},i} \qquad \mathbf{r}_{\mathcal{I}}(t_f) = \mathbf{0}$$
$$\mathbf{v}_{\mathcal{I}}(0) = \mathbf{v}_{\mathcal{I},i} \qquad \mathbf{v}_{\mathcal{I}}(t_f) = \mathbf{v}_{\mathcal{I},f}$$
$$q_{\mathcal{B}/\mathcal{I}}(t_f) = q_{\mathcal{B}/\mathcal{I},f}$$
$$\boldsymbol{\omega}_{\mathcal{B}}(0) = \boldsymbol{\omega}_{\mathcal{B},i} \qquad \boldsymbol{\omega}_{\mathcal{B}}(t_f) = \mathbf{0}$$
$$\mathbf{e}_2 \cdot \mathbf{T}_{\mathcal{B}}(t_f) = \mathbf{e}_3 \cdot \mathbf{T}_{\mathcal{B}}(t_f) = 0$$

Dynamics:

$$\dot{m}(t) = -\alpha_{\dot{m}}\|\mathbf{T}_{\mathcal{B}}(t)\|_2$$
$$\dot{\mathbf{r}}_{\mathcal{I}}(t) = \mathbf{v}_{\mathcal{I}}(t)$$
$$\dot{\mathbf{v}}_{\mathcal{I}}(t) = \frac{1}{m(t)}C_{\mathcal{I}/\mathcal{B}}(t)\mathbf{T}_{\mathcal{B}}(t) + \mathbf{g}_{\mathcal{I}}$$
$$\dot{q}_{\mathcal{B}/\mathcal{I}}(t) = \frac{1}{2}\Omega\big(\boldsymbol{\omega}_{\mathcal{B}}(t)\big)q_{\mathcal{B}/\mathcal{I}}(t)$$
$$J_{\mathcal{B}}\dot{\boldsymbol{\omega}}_{\mathcal{B}}(t) = [\mathbf{r}_{T,\mathcal{B}}\times]\,\mathbf{T}_{\mathcal{B}}(t) - [\boldsymbol{\omega}_{\mathcal{B}}(t)\times]\,J_{\mathcal{B}}\boldsymbol{\omega}_{\mathcal{B}}(t)$$

State Constraints:

$$m_{dry} \leq m(t)$$
$$\tan\gamma_{gs}\|H_{23}\mathbf{r}_{\mathcal{I}}(t)\|_2 \leq \mathbf{e}_1 \cdot \mathbf{r}_{\mathcal{I}}(t)$$
$$\cos\theta_{max} \leq 1 - 2\big(q_2^2(t) + q_3^2(t)\big)$$
$$\|\boldsymbol{\omega}_{\mathcal{B}}(t)\|_2 \leq \omega_{max}$$

Control Constraints:

$$0 < T_{min} \leq \|\mathbf{T}_{\mathcal{B}}(t)\|_2 \leq T_{max}$$
$$\cos\delta_{max}\|\mathbf{T}_{\mathcal{B}}(t)\|_2 \leq \mathbf{e}_1 \cdot \mathbf{T}_{\mathcal{B}}(t)$$

American Institute of Aeronautics and Astronautics

III.   Convex Formulation

In this section, we develop the convex formulation for Problem 1. Through a series of modifications, we morph the non-convex free-final-time problem into a convex fixed-final-time sub-problem, specifically an SOCP problem. This convex sub-problem will ultimately be solved repeatedly to convergence, a process we refer to as *successive convexification*. Assuming that the solution process is initialized properly and that the original problem admits a feasible solution, then the resulting converged solution will satisfy the dynamics, kinematics, and constraints imposed in Problem 1.

The remainder of this section is organized as follows. Section III.A outlines the process of linearizing Problem 1. Section III.B outlines the process of discretizing the linearized problem. Lastly, Section III.C summarizes the convex fixed-final-time SOCP sub-problem, and provides a detailed outline of the successive convexification algorithm.

III.A.   Linearization

III.A.1.   Dynamics and Kinematics

For convenience, we define the state vector $\mathbf{x}(t) \in \mathbb{R}^{14}$ and control vector $\mathbf{u}(t) \in \mathbb{R}^3$ as follows:

$$\mathbf{x}(t) \triangleq \begin{bmatrix} m(t) & \mathbf{r}_{\mathcal{I}}^T(t) & \mathbf{v}_{\mathcal{I}}^T(t) & q_{\mathcal{B}/\mathcal{I}}^T(t) & \boldsymbol{\omega}_{\mathcal{B}}^T(t) \end{bmatrix}^T \tag{13a}$$

$$\mathbf{u}(t) \triangleq \mathbf{T}_{\mathcal{B}}(t) \tag{13b}$$

We express the dynamics and kinematics from Eqs. 1-3 as a nonlinear vector-valued function $f : \mathbb{R}^{14} \times \mathbb{R}^3 \to \mathbb{R}^{14}$ of the state and control vectors defined in Eq. 13:

$$\frac{d}{dt}\mathbf{x}(t) = f\big(\mathbf{x}(t), \mathbf{u}(t)\big) \triangleq \begin{bmatrix} \dot{m}(t) & \dot{\mathbf{r}}_{\mathcal{I}}^T(t) & \dot{\mathbf{v}}_{\mathcal{I}}^T(t) & \dot{q}_{\mathcal{B}/\mathcal{I}}^T(t) & \dot{\boldsymbol{\omega}}_{\mathcal{B}}^T(t) \end{bmatrix}^T \tag{14}$$

Next, to address the free-final-time aspect of the problem, we would like to express Eq. 14 in terms of normalized trajectory time, $\tau \in [0, 1]$. To do so, we apply the chain rule to the left side of Eq. 14:

$$\frac{d}{dt}\mathbf{x}(t) = \frac{d\tau}{dt}\frac{d}{d\tau}\mathbf{x}(t)$$

We denote the time dilation between τ and t by the dilation coefficient σ:

$$\sigma \triangleq \left(\frac{d\tau}{dt}\right)^{-1} \tag{15}$$

Eq. 14 can now be rewritten in terms of τ as follows:

$$\mathbf{x}'(\tau) \triangleq \frac{d}{d\tau}\mathbf{x}(\tau) = \sigma f\big(\mathbf{x}(\tau), \mathbf{u}(\tau)\big) \tag{16}$$

In order to fit the nonlinear equations embedded in Eq. 16 into a convex optimization framework, we approximate the right hand side with a first-order Taylor series approximation. We define a reference trajectory comprised of state, $\hat{\mathbf{x}}(\tau)$, control, $\hat{\mathbf{u}}(\tau)$, and dilation coefficient, $\hat{\sigma}$. These quantities will be explicitly defined in Section III.C.1. The linearized system is expressed with respect to normalized time as shown below:

$$\mathbf{x}'(\tau) = A(\tau)\mathbf{x}(\tau) + B(\tau)\mathbf{u}(\tau) + \Sigma(\tau)\sigma + \mathbf{z}(\tau) \tag{17a}$$

$$A(\tau) \triangleq \hat{\sigma} \cdot \frac{\partial}{\partial \mathbf{x}} f(\mathbf{x}, \mathbf{u})\Big|_{\hat{\mathbf{x}}(\tau), \hat{\mathbf{u}}(\tau)} \tag{17b}$$

$$B(\tau) \triangleq \hat{\sigma} \cdot \frac{\partial}{\partial \mathbf{u}} f(\mathbf{x}, \mathbf{u})\Big|_{\hat{\mathbf{x}}(\tau), \hat{\mathbf{u}}(\tau)} \tag{17c}$$

$$\Sigma(\tau) \triangleq f\big(\hat{\mathbf{x}}(\tau), \hat{\mathbf{u}}(\tau)\big) \tag{17d}$$

$$\mathbf{z}(\tau) \triangleq -A(\tau)\hat{\mathbf{x}}(\tau) - B(\tau)\hat{\mathbf{u}}(\tau) \tag{17e}$$

American Institute of Aeronautics and Astronautics

III.A.2.   State Constraints

Per the discussion in Section II.B, the thrust magnitude lower bound constraint is the only remaining source of non-convexity in Problem 1. As such, we proceed to linearize this constraint. First, we define the function $g : \mathbb{R}^3 \to \mathbb{R}$ as follows:

$$g\big(\mathbf{u}(\tau)\big) \triangleq T_{min} - \|\mathbf{u}(\tau)\|_2 \leq 0 \tag{18}$$

Then, replacing the left hand side of Eq. 18 with a first-order Taylor series approximation and simplifying, we obtain the following:

$$T_{min} \leq B_g(\tau)\mathbf{u}(\tau) \tag{19a}$$

$$B_g(\tau) \triangleq \frac{\hat{\mathbf{u}}^T(\tau)}{\|\hat{\mathbf{u}}(\tau)\|_2} \tag{19b}$$

III.B.   Discretization

To cast the continuous-time optimal control problem into a finite-dimensional parameter optimization problem, we discretize the trajectory into K evenly distributed discretization points. We perform the discretization with respect to normalized trajectory time, τ. For convenience, we define the following two sets:

$$\mathcal{K} \triangleq \{0, 1, \ldots, K - 2, K - 1\}$$
$$\bar{\mathcal{K}} \triangleq \{0, 1, \ldots, K - 3, K - 2\}$$

Since normalized trajectory time is defined on the interval $\tau \in [0, 1]$, we define the time at index k as:

$$\tau_k \triangleq \left(\frac{k}{K - 1}\right), \quad \forall k \in \mathcal{K}$$

To preserve more feasibility, we assume a first-order-hold on the control over each time step. Thus, over the interval $\tau \in [\tau_k, \tau_{k+1}]$, we can express $\mathbf{u}(\tau)$ in terms of $\mathbf{u}_k \triangleq \mathbf{u}(\tau_k)$ and $\mathbf{u}_{k+1} \triangleq \mathbf{u}(\tau_{k+1})$ as follows:

$$\mathbf{u}(\tau) \triangleq \alpha_k(\tau)\mathbf{u}_k + \beta_k(\tau)\mathbf{u}_{k+1}, \quad \tau \in [\tau_k, \tau_{k+1}], \, \forall k \in \bar{\mathcal{K}} \tag{20a}$$

$$\alpha_k(\tau) \triangleq \left(\frac{\tau_{k+1} - \tau}{\tau_{k+1} - \tau_k}\right) \tag{20b}$$

$$\beta_k(\tau) \triangleq \left(\frac{\tau - \tau_k}{\tau_{k+1} - \tau_k}\right) \tag{20c}$$

We use $\Phi_A\big(\tau_{k+1}, \tau_k\big)$ to denote the state transition matrix that describes the zero-input evolution from $\mathbf{x}_k \triangleq \mathbf{x}(\tau_k)$ to $\mathbf{x}_{k+1} \triangleq \mathbf{x}(\tau_{k+1})$. The state transition matrix is governed by the following differential equation:

$$\frac{d}{d\tau}\Phi_A\big(\tau, \tau_k\big) = A(\tau)\Phi_A\big(\tau, \tau_k\big), \quad \Phi_A\big(\tau_k, \tau_k\big) = I, \quad \forall k \in \bar{\mathcal{K}} \tag{21}$$

Using Eqs. 20 and 21, we express the discrete-time dynamics that relate $\mathbf{x}_k$ to $\mathbf{x}_{k+1}$ as follows:

$$\mathbf{x}_{k+1} = \bar{A}_k\mathbf{x}_k + \bar{B}_k\mathbf{u}_k + \bar{C}_k\mathbf{u}_{k+1} + \bar{\Sigma}_k\sigma + \bar{\mathbf{z}}_k, \quad \forall k \in \bar{\mathcal{K}} \tag{22a}$$

$$\bar{A}_k \triangleq \Phi_A\big(\tau_{k+1}, \tau_k\big) \tag{22b}$$

$$\bar{B}_k \triangleq \int_{\tau_k}^{\tau_{k+1}} \Phi_A\big(\tau_{k+1}, \xi\big)B(\xi)\alpha_k(\xi)d\xi \tag{22c}$$

$$\bar{C}_k \triangleq \int_{\tau_k}^{\tau_{k+1}} \Phi_A\big(\tau_{k+1}, \xi\big)B(\xi)\beta_k(\xi)d\xi \tag{22d}$$

$$\bar{\Sigma}_k \triangleq \int_{\tau_k}^{\tau_{k+1}} \Phi_A\big(\tau_{k+1}, \xi\big)\Sigma(\xi)d\xi \tag{22e}$$

$$\bar{\mathbf{z}}_k \triangleq \int_{\tau_k}^{\tau_{k+1}} \Phi_A\big(\tau_{k+1}, \xi\big)\mathbf{z}(\xi)d\xi \tag{22f}$$

Lastly, the remaining state and control constraints in Problem 1 are enforced for each τ_k, $\forall k \in \mathcal{K}$.

American Institute of Aeronautics and Astronautics

III.C.   Successive Convexification

In this section we outline the successive convexification process, in which we indirectly solve a non-convex optimization problem by instead solving a sequence of related convex sub-problems. We denote the ith iterate in this sequence by the superscript i. Before we present the finalized SOCP sub-problem and successive convexification algorithm, we introduce two more important modifications: (1) trust regions, and (2) dynamic relaxation.

III.C.1.   Trust Regions

In order for successive convexification to work, we must ensure that the problem remains bounded and feasible throughout the convergence process. The former issue arises when the linearization of an iterate results in constraints that admit an unbounded cost, whereas the latter issue will be discussed in Section III.C.2. We mitigate the possibility of unbounded solutions in each sub-problem by augmenting the cost function with terms that serve as soft trust regions defined around the previous iterate. To do so, we first define the relative quantities shown below:

$$\delta\mathbf{x}_k^i \triangleq \mathbf{x}_k^i - \mathbf{x}_k^{i-1}, \quad \forall\, k \in \mathcal{K} \tag{23a}$$

$$\delta\mathbf{u}_k^i \triangleq \mathbf{u}_k^i - \mathbf{u}_k^{i-1}, \quad \forall\, k \in \mathcal{K} \tag{23b}$$

$$\delta\sigma^i \triangleq \sigma^i - \sigma^{i-1} \tag{23c}$$

Then, defining $\bar{\boldsymbol{\Delta}}^i \in \mathbb{R}_+^K$ and $\Delta_\sigma^i \in \mathbb{R}_+$, we impose the following constraints:

$$\delta\mathbf{x}_k^i \cdot \delta\mathbf{x}_k^i + \delta\mathbf{u}_k^i \cdot \delta\mathbf{u}_k^i \leq \mathbf{e}_k \cdot \bar{\boldsymbol{\Delta}}^i, \quad \forall\, k \in \mathcal{K} \tag{24a}$$

$$\delta\sigma^i \cdot \delta\sigma^i \leq \Delta_\sigma^i \tag{24b}$$

Lastly, defining the weight terms $w_\Delta^i \in \mathbb{R}_{++}$ and $w_{\Delta_\sigma} \in \mathbb{R}_{++}$, we augment the cost function from Problem 1 with the $c_{TR}^i \in \mathbb{R}_{++}$, defined as:

$$c_{TR}^i \triangleq w_\Delta^i \|\bar{\boldsymbol{\Delta}}^i\|_2 + w_{\Delta_\sigma} \|\Delta_\sigma^i\|_1 \tag{25}$$

Note that the trust regions are centered at $\mathbf{x}_k^{i-1}$, $\mathbf{u}_k^{i-1}$, and σ^{i-1}. Thus, when we perform the numerical integration of Eqs. 21, 22b-22f over the interval $\tau \in (\tau_k, \tau_{k+1}]$, we elect to evaluate the linearization in Eq. 17 about the nonlinear trajectory beginning at $\mathbf{x}_k^{i-1}$, and generated using the control $\mathbf{u}(\tau)$ defined in Eq. 20. Doing so for each $k \in \bar{\mathcal{K}}$ explicitly defines the linearization path $\hat{\mathbf{x}}(\tau)$, $\hat{\mathbf{u}}(\tau)$, and $\hat{\sigma}$ introduced in Section III.A.1.

III.C.2.   Dynamic Relaxation

The second issue alluded to in the previous section is referred to as *artificial infeasibility*. Artificial infeasibility can be encountered during the convergence process when the linearization is not favorable for feasibility. For example, if the problem is linearized about an unrealistically short time-of-flight, the linearized equations will likely not admit a feasible solution. Artificial infeasibility is encountered very frequently during the first few iterations of a typical successive convexification sequence, and is largely due to initiating the process with a poor initial guess. To mitigate this commonly encountered condition, we introduce a dynamic relaxation term, $\boldsymbol{\nu}_k^i \in \mathbb{R}^{14}$, and add it to Eq. 22a. Consequently, the dynamics we encode in the problem are given by:

$$\mathbf{x}_{k+1}^i = \bar{A}_k^i \mathbf{x}_k^i + \bar{B}_k^i \mathbf{u}_k + \bar{C}_k^i \mathbf{u}_{k+1}^i + \bar{\Sigma}_k^i \sigma^i + \bar{\mathbf{z}}_k^i + \boldsymbol{\nu}_k^i, \quad \forall\, k \in \bar{\mathcal{K}} \tag{26}$$

For notational convenience, we concatenate the $\boldsymbol{\nu}_k^i$ vectors into a larger vector $\bar{\boldsymbol{\nu}}^i \in \mathbb{R}^{14(K-1)}$:

$$\bar{\boldsymbol{\nu}}^i \triangleq \begin{bmatrix} \boldsymbol{\nu}_0^{i\,T} & \cdots & \boldsymbol{\nu}_{K-2}^{i\,T} \end{bmatrix}^T$$

Lastly, defining the weight term $w_\nu \in \mathbb{R}_{++}$, we augment the already modified cost function from Problem 1 with $c_{DR}^i \in \mathbb{R}_+$, defined as:

$$c_{DR}^i \triangleq w_\nu \|\bar{\boldsymbol{\nu}}^i\|_1 \tag{27}$$

The intuition behind this modification is that by selecting a large value for w_ν, $\boldsymbol{\nu}_k^i$ serves as a heavily penalized auxiliary control variable that acts when necessary to prevent infeasibility. A converged solution with a negligible c_{DR}^i component of the cost indicates that the converged solution is dynamically feasible.

American Institute of Aeronautics and Astronautics

III.C.3.   Convex Sub-Problem

We are now ready to summarize the convex sub-problem that will be solved repeatedly by the successive convexification algorithm. Note that by virtue of the time normalization introduced in Section III.A.1, this problem can be viewed as a fixed-final-time optimization problem due to the fact that the final normalized time is always equal to unity. As a consequence, t_f in Problem 1 is substituted with σ_i. Lastly, for convenience, we rewrite Eq. 9 in more compact form:

$$\cos\theta_{max} \leq 1 - 2\|H_q q_{\mathcal{B}/\mathcal{I},k}^i\|_2^2 \tag{28a}$$

$$H_q \triangleq \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{28b}$$

The summary of the convex sub-problem is provided in Problem 2.

Problem 2. : *Convex Discrete-Time Fixed-Final-Time Problem*

<u>Cost Function:</u>

$$\underset{\sigma^i, \mathbf{u}_k^i}{\text{minimize}} \quad \sigma^i + w_\nu\|\bar{\boldsymbol{\nu}}^i\|_1 + w_\Delta^i\|\bar{\boldsymbol{\Delta}}^i\|_2 + w_{\Delta_\sigma}\|\Delta_\sigma\|_1 \quad \text{(See Eqs. 25,27)}$$

subject to:

<u>Boundary Conditions:</u>

$$m_0^i = m_{wet}$$
$$\mathbf{r}_{\mathcal{I},0}^i = \mathbf{r}_{\mathcal{I},i} \qquad \mathbf{r}_{\mathcal{I},K}^i = \mathbf{0}$$
$$\mathbf{v}_{\mathcal{I},0}^i = \mathbf{v}_{\mathcal{I},i} \qquad \mathbf{v}_{\mathcal{I},K}^i = \mathbf{v}_{\mathcal{I},f}$$
$$q_{\mathcal{B}/\mathcal{I},K}^i = q_{\mathcal{B}/\mathcal{I},f}$$
$$\boldsymbol{\omega}_{\mathcal{B},0}^i = \boldsymbol{\omega}_{\mathcal{B},i} \qquad \boldsymbol{\omega}_{\mathcal{B},K}^i = \mathbf{0}$$
$$\mathbf{e}_2 \cdot \mathbf{u}_K^i = \mathbf{e}_3 \cdot \mathbf{u}_K^i = 0$$

<u>Dynamics:</u>

$$\mathbf{x}_{k+1}^i = \bar{A}_k^i \mathbf{x}_k^i + \bar{B}_k^i \mathbf{u}_k^i + \bar{C}_k^i \mathbf{u}_{k+1}^i + \bar{\Sigma}_k^i \sigma^i + \bar{\mathbf{z}}_k^i + \boldsymbol{\nu}_k^i \qquad \text{(See Eqs. 17,}$$
$$\text{20,21,22,26)}$$

<u>State Constraints:</u>

$$m_{dry} \leq m_k^i \qquad \text{(See Eq. 6)}$$
$$\tan\gamma_{gs}\|H_{23}\mathbf{r}_{\mathcal{I},k}^i\|_2 \leq \mathbf{e}_1 \cdot \mathbf{r}_{\mathcal{I},k}^i \qquad \text{(See Eq. 7)}$$
$$\cos\theta_{max} \leq 1 - 2\|H_q q_{\mathcal{B}/\mathcal{I},k}^i\|_2^2 \qquad \text{(See Eq. 28)}$$
$$\|\boldsymbol{\omega}_{\mathcal{B},k}^i\|_2 \leq \omega_{max} \qquad \text{(See Eq. 10)}$$

<u>Control Constraints:</u>

$$T_{min} \leq B_g(\tau_k)\mathbf{u}_k^i \qquad \text{(See Eq. 19)}$$
$$\|\mathbf{u}_k^i\|_2 \leq T_{max} \qquad \text{(See Eq. 11)}$$
$$\cos\delta_{max}\|\mathbf{u}_k^i\|_2 \leq \mathbf{e}_1 \cdot \mathbf{u}_k \qquad \text{(See Eq. 12)}$$

<u>Trust Regions:</u>

$$\delta\mathbf{x}_k^i \cdot \delta\mathbf{x}_k^i + \delta\mathbf{u}_k^i \cdot \delta\mathbf{u}_k^i \leq \Delta_k^i \qquad \text{(See Eqs. 23-24)}$$
$$\|\delta\sigma^i\|_1 \leq \Delta_\sigma^i$$

III.C.4.   Algorithm

In this section we summarize the successive convexification algorithm. We define two tolerances, $\Delta_{tol} \in \mathbb{R}_{++}$ and $\nu_{tol} \in \mathbb{R}_{++}$. The algorithm is shown in Algorithm 1.

Algorithm 1. : *Successive Convexification*

Initialization:

for $\forall\, k \in \mathcal{K}$:

define: $\quad \alpha_1 \triangleq \dfrac{K-k}{K} \qquad \alpha_2 \triangleq \dfrac{k}{K}$

$$m_k^0 = \alpha_1 m_{wet} + \alpha_2 m_{dry}$$
$$\mathbf{r}_{\mathcal{I},k}^0 = \alpha_1 \mathbf{r}_{\mathcal{I},i}$$
$$\mathbf{v}_{\mathcal{I},k}^0 = \alpha_1 \mathbf{v}_{\mathcal{I},i} + \alpha_2 \mathbf{v}_{\mathcal{I},f}$$
$$q_{\mathcal{B}/\mathcal{I},k}^0 = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^T$$
$$\boldsymbol{\omega}_{\mathcal{B},k}^0 = \mathbf{0}$$

set: $\mathbf{x}_k^0 = \begin{bmatrix} m_k & \mathbf{r}_{\mathcal{I},k}^T & \mathbf{v}_{\mathcal{I},k}^T & q_{\mathcal{B}/\mathcal{I},k}^T & \boldsymbol{\omega}_{\mathcal{B},k}^T \end{bmatrix}^T$, $\mathbf{u}_k^0 = -m_k^0 \mathbf{g}_{\mathcal{I}}$, and $\sigma^0 = t_{f,guess}$

end

for $\forall\, k \in \mathcal{K}$:

compute: $\bar{A}_k^0$, $\bar{B}_k^0$, $\bar{C}_k^0$, $\bar{\Sigma}_k^0$, and $\bar{\mathbf{z}}_k^0$ using Eqs. 17, 20-22, 26

end

Successive Convexification Loop:

for $i \in \{1, N_{iter,max}\}$

(1) solve Problem 2 using $\mathbf{x}_k^{(i-1)}$, $\mathbf{u}_k^{(i-1)}$, $\sigma^{(i-1)}$, $\bar{A}_k^{(i-1)}$, $\bar{B}_k^{(i-1)}$, $\bar{C}_k^{(i-1)}$, $\bar{\Sigma}_k^{(i-1)}$, $\bar{\mathbf{z}}_k^{(i-1)}$

(2) store newly computed $\mathbf{x}_k^i$, $\mathbf{u}_k^i$, and σ^i

(3) if $(\|\bar{\boldsymbol{\Delta}}^i\|_2 \leq \Delta_{tol})$ and $(\|\bar{\boldsymbol{\nu}}^i\|_1 \leq \nu_{tol})$

exit loop

else

(3.1) compute new $\bar{A}_k^i$, $\bar{B}_k^i$, $\bar{C}_k^i$, $\bar{\Sigma}_k^i$, and $\bar{\mathbf{z}}_k^i$ using Eqs. 17, 20-22, 26

(3.2) increment i

(3.3) return to step 1

end

end

American Institute of Aeronautics and Astronautics

IV.   Simulation Results

In this section we present simulation results of our algorithm. We implemented our algorithm in MATLAB using CVX[11] and SDPT3.[33] We investigated two scenarios, a two-dimensional in-plane maneuver, and a three-dimensional maneuver. Note that for the nature of the two-dimensional case was dictated by the initial conditions provided, and was not a simplified formulation of our algorithm. For illustration purposes, notional non-dimensional quantities were used, and thus do not necessarily represent the behavior of any active rocket system. Table 1 contains the simulation parameters common to both cases. Table 2 contains the algorithm parameters and boundary conditions common to both cases.

Table 1. Simulation Parameters

Parameter	Value	Units
$\mathbf{g}_{\mathcal{I}}$	$-\mathbf{e}_1$	U_L/U_T^2
m_{wet}	2.00	U_M
m_{dry}	1.00	U_M
T_{min}	0.30	$U_M U_L/U_T^2$
T_{max}	5.00	$U_M U_L/U_T^2$
δ_{max}	20	°
θ_{max}	90	°
γ_{gs}	20	°
ω_{max}	60	$°/U_T$
$J_{\mathcal{B}}$	$1e^{-2} \cdot I$	$U_M U_L^2$
$\mathbf{r}_{T,\mathcal{B}}$	$-1e^{-2} \cdot \mathbf{e}_1$	U_L

Table 2. B.C.'s and Algorithm Parameters

Parameter	Value	Units
w_ν	$1e^5$	-
w_Δ^i	$1e^{-3}$	-
w_{Δ_σ}	$1e^{-1}$	-
ν_{tol}	$1e^{-10}$	-
Δ_{tol}	$1e^{-3}$	-
$N_{iter,max}$	15	-
K	50	-
$\mathbf{r}_{\mathcal{I},i}$	$\begin{bmatrix} 4 & 4 & 0 \end{bmatrix}^T$	U_L
$\mathbf{v}_{\mathcal{I},f}$	$-1e^{-1} \cdot \mathbf{e}_1$	U_L/U_T
$\boldsymbol{\omega}_{\mathcal{B},i}$	$\mathbf{0}$	$°/U_T$
$q_{\mathcal{B}/\mathcal{I},f}$	$\begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^T$	-

IV.A.   2-D Example

In this example, we specify the initial conditions such that the vehicle remains in the Up-East plane. The vehicle is initialized with a horizontal velocity of $4\,U_L/U_T$ to the west. Figure 1 shows the trajectory with the thrust and pointing vectors marked at every fourth discretization point, the tilt angle, the thrust magnitude, and the angular velocity profile. The trajectory is seen to ride the maximum tilt angle limit for about half of the trajectory, and then activates the angular velocity limit during the second half. Throughout, the thrust magnitude activates both the minimum and maximum thrust magnitude bounds, as one would expect for a minimum-time trajectory.

Ten different time-of-flight guesses were attempted, ranging from 1.0 to 10.0 time-units, in 1.0 time-unit increments. All generated attempts generated the same trajectory, yielding time-of-flights within $1e^{-2}$ time-units of each other. Thus, this suggests that despite the non-convexity of the problem, and despite the coarse initialization routine specified in Algorithm 1, the algorithm converged to the same solution. The algorithm converged in approximately 12 iterations.

IV.B.   3-D Example

Figure 2 and 3 shows results from the 3-D example. Despite the more complex nature of the trajectory, the algorithm performed nearly identically when compared to the 2-D case. Here, bang-off-bang features similar to those in the 2-D case are apparent in the thrust magnitude profile, while the angular rate limit is also saturated in the middle of the trajectory. This case had the same success in converging to a common time-of-flight given different initial guesses.
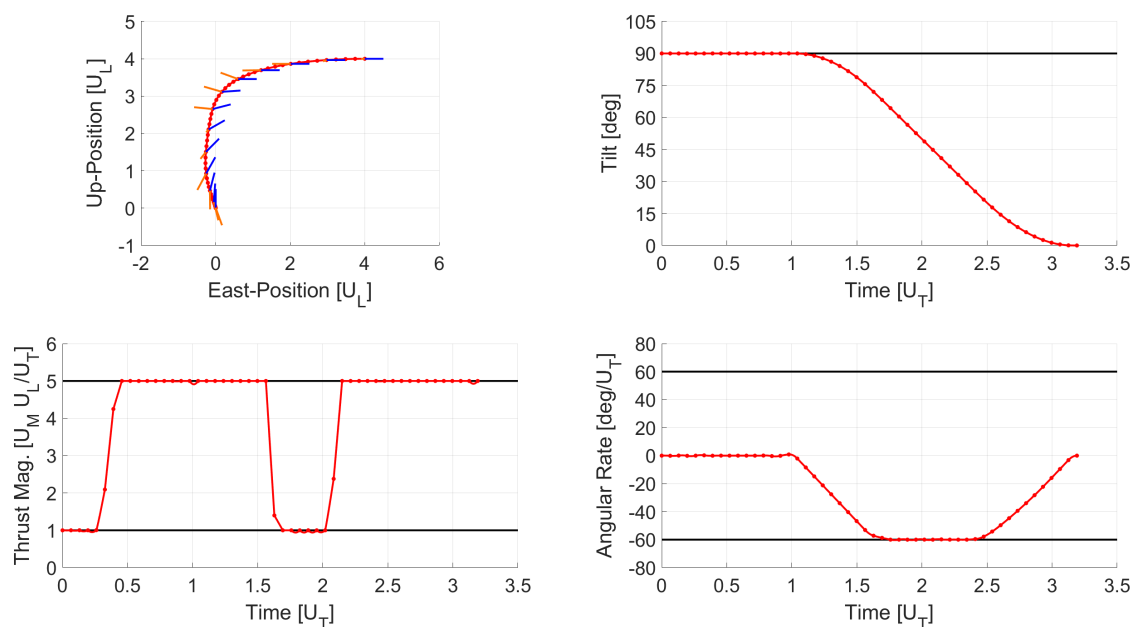
American Institute of Aeronautics and Astronautics

Figure 1. Top Left - The trajectory in the Up-East plane. The blue vectors represent the X-axis of the vehicle, whereas the orange lines represent the thrust-plume direction. The red line represents the trajectory of the vehicle, while the makers along the trajectory represent the discretization points. Top Right - The tilt profile versus time. The black line represents that maximum allowable tilt angle. Bottom Left - The thrust magnitude profile versus time. The top and bottom black lines represent the maximum and minimum allowable thrust magnitude, respectively. Bottom Right - The angular velocity profile versus time. The black lines represent the maximum allowable angular rate.
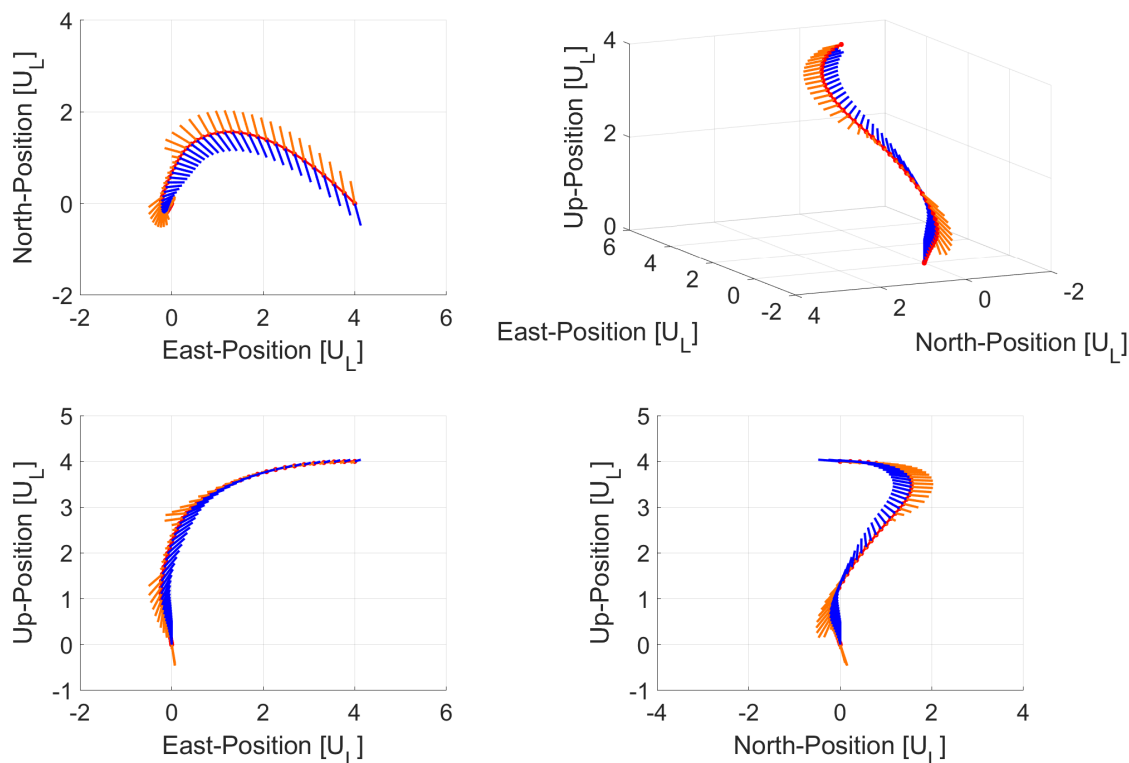

Figure 2. Top Left - North-East view of the trajectory. Top Right - Perspective view of the trajectory. Bottom Left - Up-East view of the trajectory. Bottom Right - Up-North view of the trajectory.

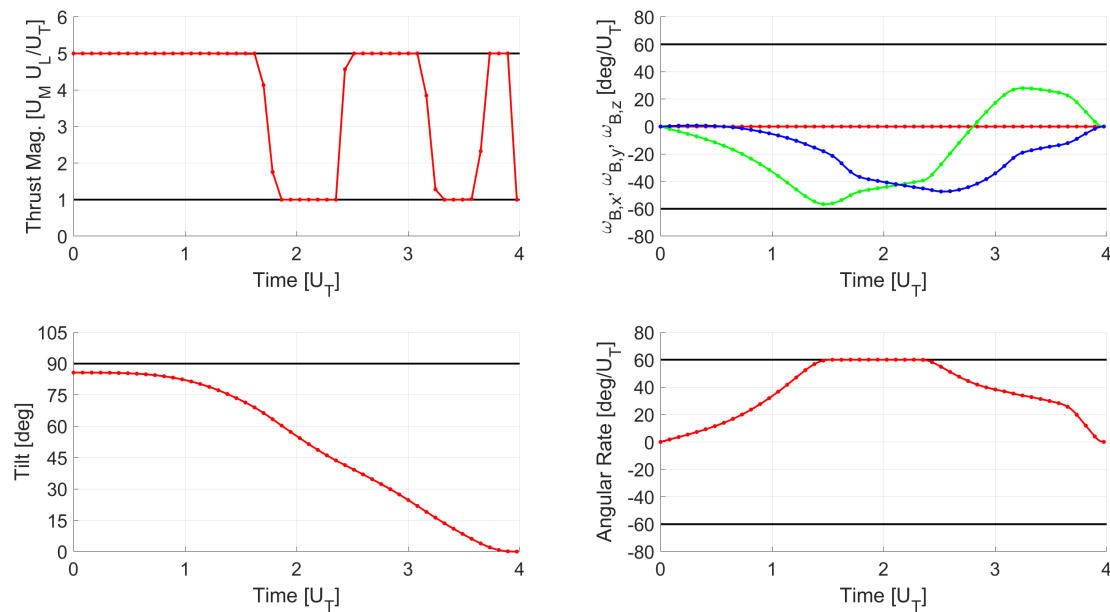American Institute of Aeronautics and Astronautics

Figure 3. Top Left - Thrust magnitude profile versus time. Top Right - Angular rate components versus time. Bottom Left - tilt angle profile versus time. Bottom Right - Angular rate magnitude versus time.

V. Conclusion and Future Work

In this paper, we presented a successive convexification formulation to solve the minimum-time 6-DoF Mars rocket powered landing problem. To solve this nonlinear, non-convex optimal control problem, we employed the following procedure. First, we initialized the process with an easily generated, dynamically inconsistent trajectory. Second, for each iterate, (a) the problem was linearized and discretized by computing the state and control matrices for the corresponding linear-time-varying system, (b) non-convex constraints were linearized, and (c) the resulting convex SOCP sub-problem was solved to full optimality. Third, the process was repeated until the changes between iterates was sufficiently small, and the relaxation term subsided.

Simulations were conducted in order to exercise the proposed algorithm. The results presented showed that the algorithm can indeed be initialized with a simple trajectory and a poor time-of-flight guess. Despite the complexity of the problem, this method was shown to converge in approximately 15 successive convexification iterations or less.

In future work, we plan to explore the convergence properties of the successive convexification technique outlined in this paper, present real-time timing statistics, and incorporate simple aerodynamic effects into the 6-DoF problem.

Acknowledgments

References

[1]Szmuk, M., Açıkmeşe, B. A., and Jr., A. W. B., "Successive Convexification for Fuel-Optimal Powered Landing with Aerodynamic Drag and Non-Convex Constraints," *AIAA Guidance, Navigation, and Control Conference*, San Diego, CA, 2016, p. 0378.

[2]Szmuk, M., Eren, U., and Açıkmeşe, B. A., "Successive Convexification for Mars 6-DoF Powered Descent Landing Guidance," *AIAA Guidance, Navigation, and Control Conference*, Grapevine, TX, 2017, p. 1500.

[3]Blackmore, L., "Autonomous Precision Landing of Space Rockets," *NAE Winter Bridge on Frontiers of Engineering*, Vol. 4, No. 46, 2016.

[4]Klumpp, A. R., "Apollo Lunar Descent Guidance," *Automatica*, Vol. 10, 1974, pp. 133–146.

[5]Najson, F. and Mease, K. D., "Computationally Inexpensive Guidance Algorithm for Fuel-Efficient Terminal Descent," *Journal of Guidance, Control, and Dynamics*, Vol. 29, No. 4, 2006, pp. 955–964.

[6]Topcu, U., Casoliva, J., and Mease, K. D., "Minimum-Fuel Powered Descent for Mars Pinpoint Landing," *Journal of Spacecraft and Rockets*, Vol. 44, No. 2, 2007, pp. 324–331.

[7]Sostaric, R. R. and Rea, J. R., "Powered descent guidance methods for the moon and mars," *AIAA Guidance, Navigation, and Control Conference, San Francisco, CA*, 2005.

[8]Boyd, S. and Vandenberghe, L., *Convex Optimization*, Cambridge University Press, 2004.

[9]Dueri, D., Zhang, J., and Açıkmeşe, B., "Automated Custom Code Generation for Embedded, Real-time Second Order Cone Programming," *IFAC World Congress*, 2014.

[10]Dueri, D., Açıkmeşe, B., Scharf, D. P., and Harris, M. W., "Customized Real-Time Interior-Point Methods for Onboard Powered-Descent Guidance," *Journal of Guidance, Control, and Dynamics*, 2016, pp. 1–16.

[11]Grant, M. and Boyd, S., "CVX: Matlab Software for Disciplined Convex Programming, version 2.1," *http://cvxr.com/cvx*, March 2014.

[12]Domahidi, A., Zgraggen, A. U., Zeilinger, M. N., Morari, M., and Jones, C. N., "Efficient interior point methods for multistage problems arising in receding horizon control," *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, IEEE, 2012, pp. 668–674.

[13]Açıkmeşe, B. and Blackmore, L., "Lossless Convexification of a Class of Optimal Control Problems with Non-Convex Control Constraints," *Automatica*, Vol. 47, No. 2, 2011, pp. 341–347.

[14]Harris, M. and Açıkmeşe, B., "Lossless Convexification of Non-convex Optimal Control Problems for State Constrained Linear Systems," *Automatica*, Vol. 50, No. 9, 2014, pp. 2304–2311.

[15]Blackmore, L., Açıkmeşe, B., and Carson, J. M., "Lossless convexfication of control constraints for a class of nonlinear optimal control problems," *System and Control Letters*, Vol. 61, No. 4, 2012, pp. 863–871.

[16]Açıkmeşe, B. and Ploen, S. R., "A powered descent guidance algorithm for Mars pinpoint landing," *AIAA GNC Conference and Exhibit, San Francisco*, 2005.

[17]Açıkmeşe, B. and Ploen, S. R., "Convex Programming Approach to Powered Descent Guidance for Mars Landing," *AIAA Journal of Guidance, Control and Dynamics*, Vol. 30, No. 5, 2007, pp. 1353–1366.

[18]Blackmore, L., Açıkmeşe, B., and Scharf, D. P., "Minimum Landing Error Powered Descent Guidance for Mars Landing using Convex Optimization," *AIAA Journal of Guidance, Control and Dynamics*, Vol. 33, No. 4, 2010.

[19]Açıkmeşe, B., Carson, J., and Blackmore, L., "Lossless Convexification of Non-convex Control Bound and Pointing Constraints of the Soft Landing Optimal Control Problem," *IEEE Transactions on Control Systems Technology*, Vol. 21, No. 6, 2013, pp. 2104–2113.

[20]Harris, M. W. and Açıkmeşe, B., "Maximum Divert for Planetary Landing Using Convex Optimization," *Journal of Optimization Theory and Applications*, 2013, pp. 1–21.

[21]Açıkmeşe, B., Aung, M., Casoliva, J., Mohan, S., Johnson, A., Scharf, D., Masten, D., Scotkin, J., Wolf, A., and Regehr, M. W., "Flight Testing of Trajectories Computed by G-FOLD: Fuel Optimal Large Divert Guidance Algorithm for Planetary Landing," *AAS/AIAA Spaceflight Mechanics Meeting*, 2013.

[22]Scharf, D. P., M. W. Regehr, D. D., Açıkmeşe, B., Vaughan, G. M., Benito, J., Ansari, H., M. Aung, A. J., Masten, D., Nietfeld, S., Casoliva, J., and Mohan, S., "ADAPT Demonstrations of Onboard Large-Divert Guidance with a Reusable Launch Vehicle," *IEEE Aerospace Conference*, 2014.

[23]Lee, U. and Mesbahi, M., "Optimal Power Descent Guidance with 6-DoF Line of Sight Constraints via Unit Dual Quaternions," *AIAA SciTech, Guidance, Navigation, and Control*, Kissimmee, FL, 01/2015 2015.

[24]Lee, U. and Mesbahi, M., "Constrained Autonomous Precision Landing via Dual Quaternions and Model Predictive Control," *Journal of Guidance, Control, and Dynamics*, 2016.

[25]Casoliva, J., *Spacecraft Trajectory Generation by Successive Approximation for Powered Descent and Cyclers*, Ph.D. thesis, University of California, Irvine, 2013.

[26]Liu, X. and Lu, P., "Solving Nonconvex Optimal Control Problems by Convex Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 3, 2014, pp. 750–765.

[27]Liu, X., Shen, Z., and Lu, P., "Entry trajectory optimization by second-order cone programming," *Journal of Guidance, Control, and Dynamics*, Vol. 39, No. 2, 2015, pp. 227–241.

[28]Mao, Y., Szmuk, M., and Acikmese, B., "Successive Convexification of Non-Convex Optimal Control Problems and Its Convergence Properties," *ArXiv e-prints*, Aug. 2016, arXiv:1608.05133.

[29]Mao, Y., Dueri, D., Szmuk, M., and Açıkmeşe, B., "Successive Convexification of Non-Convex Optimal Control Problems with State Constraints," *ArXiv e-prints*, Jan. 2017, arXiv:1701.00558.

[30]Mao, Y., Szmuk, M., and Açıkmeşe, B., "Successive convexification of non-convex optimal control problems and its convergence properties," *2016 IEEE 55th Conference on Decision and Control (CDC)*, Dec 2016, pp. 3636–3641.

[31]Wang, Z. and Grant, M. J., "Constrained Trajectory Optimization for Planetary Entry via Sequential Convex Programming," *AIAA Atmospheric Flight Mechanics Conference*, 2016, p. 3241.

[32]Liu, X. and Lu, P., "Solving nonconvex optimal control problems by convex optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 3, 2014, pp. 750–765.

[33]Toh, K., Todd, M., and Tutuncu, R., "SDPT3 — a Matlab Software Package for Semidefinite Programming," *Optimization Methods and Software*, Vol. 11, No. 1, 1999, pp. 545–581.