A State-Triggered Line of Sight Constraint for 6-DoF Powered Descent Guidance Problems

Taylor P. Reynolds[*], Michael Szmuk[*], Danylo Malyuta[*],
Mehran Mesbahi[†], Behçet Açıkmeşe[†]
Dept. of Aeronautics & Astronautics, University of Washington, Seattle, WA 98195, USA
and
John M. Carson III[‡]
NASA Johnson Space Center, Houston, TX 77058, USA

This paper presents the formulation of a constrained 6-degree-of-freedom (6-DoF) powered descent guidance problem. The goal of this work is to design algorithms that obtain locally optimal solutions to such problems, and that are amenable to real-time implementation. Using unit dual quaternions to parameterize the equations of motion, we devise a free final time continuous optimal control problem that is subject to state and control constraints. A novel feature of this formulation is the use of *state-triggered constraints*, which are constraints enforced only when a certain state-dependent criterion is met. We use these constraints to model a line of sight pointing constraint that is enforced conditionally based on the distance from the landing site. A numerical example highlights how the inclusion of this constraint alters control commands and the resulting descent trajectory.

I. Introduction

Powered descent guidance refers the problem of transferring a vehicle from an estimated initial state to a desired final state by using rocket-powered engines and/or reaction control systems. The problem of *optimal* powered descent has been well studied since the Apollo program sought to achieve soft-landings on the moon.[1–3] The first tractable solutions adequate for flight operations, however, were limited to one-degree-of-freedom (1-DoF) systems for purely vertical descent trajectories.[1] The structure of optimal solutions for the more general 3-DoF translational guidance problem has been understood since the work of Lawden,[2,4] but numerical solutions were difficult to obtain at the time. While optimal guidance was not incorporated in the Apollo flight code, there is evidence that the system designers were aware of how their polynomial guidance scheme stacked up relative to the optimal solution.[5]

Near the turn of the century, there was renewed interest in optimal powered descent guidance problems due primarily to robotic Martian landing missions. These works expanded on previous theory in seek of analytical solutions to the 3-DoF problem. D'Souza studied the free final time, minimum energy solutions and obtained an analytical feedback control law as a function of the *time-to-go*.[6] Topcu, Casoliva and Mease presented further results on the minimum fuel problem and compared theoretical predictions against rapidly maturing numerical solvers.[7,8] Several other authors continued to study the necessary conditions of the minimum fuel 3-DoF guidance problem using optimal control theory.[9–11]

At the same time, Açıkmeşe and Ploen published work with an alternative viewpoint on the 3-DoF problem.[12–14] They took a convex programming approach, and showed that a non-convex (non-zero) lower bound on thrust magnitude can be relaxed by introducing a slack variable.[14] Using Pontryagin's maximum principle, they showed that in fact this relaxation is lossless, and yields the same optimal solution as the original problem. A subsequent change of variables and relaxation led to a fully convex problem formulation that can be solved efficiently. This lossless convexification result began a fruitful series of work that expanded the

[*] Doctoral Student, AIAA Student Member {tpr6,szmuk,danylo}@uw.edu
[†] Professor, AIAA Associate Fellow, {mesbahi,behcet}@uw.edu
[‡] SPLICE Project Manager, AIAA Associate Fellow, john.m.carson@nasa.gov

theory to non-convex thrust pointing constraints,[15–17] minimum-landing error problems[18] and more general optimal control problems.[19–21] Interior time state and control constraints can render the solution of the necessary conditions of optimal control theory a difficult proposition; but convex optimization does not suffer from such issues provided the constraints are convex. Fortunately, a majority of the constraints of interest in the powered descent guidance problem come in the form of second-order cones, a class of convex sets that we exploit in this work.

More recent work has explored extensions to the 6-DoF problem that considers both translational and rotational motion. These include the use of Lyapunov techniques,[22] model predictive control[23, 24] and feed-forward trajectory generation techniques.[25–28] The latter techniques devise iterative strategies that are used to obtain feasible solutions to nonlinear and nonconvex optimal control problems that approximate local optimality. A complete characterization of fuel optimal solution(s) for the 6-DoF problem is an active area of research. However, numerical techniques offer promising results that locally optimal solutions can be found by sequentially solving convex optimization problems with guaranteed convergence properties.[29–31]

The state variables selected for the 6-DoF problem formulation can be used to classify various methods. For example, one may use standard Cartesian variables in conjunction with unit quaternions[25–27] or dual quaternions.[22–24, 32, 33] This distinction should be viewed as a design choice during guidance system design. We find that when there are constraints that naturally couple rotation and translation (such as line of sight constraints), parameterizations using dual quaternions provide an efficient alternative. In this work, we elect to parameterize the equations of motion and state constraints using dual quaternions.

Dual quaternions are a generalization of Hamilton's quaternions that encode both relative orientation and position information in a single parameter.[34] An attractive feature of this parameterization is that the equations of motion can be expressed in a form similar to the standard quaternion kinematic and dynamic equations. Moreover, the formulation of several key constraints – including line of sight – are convex over a given set of dual quaternions.[24, 35]

A. Contributions of This Work

The Apollo guidance system designated three powered descent phases: the braking phase, the approach phase and the terminal-descent phase.[5] The braking phase slowed the vehicle from orbital speeds by thrusting primarily in the anti-velocity direction. Prior to the approach phase, a pitch-up maneuver to a desired attitude was executed to serve as an initial attitude for the approach phase. During the approach phase, the lunar module maintained continuous visibility of the landing site until roughly 5 seconds before the terminal-descent phase began. The attitude guidance system was designed to ensure a line of sight to the landing site *only when the geometry permitted*.[5] There was no a-priori guarantee that this would occur.

We consider a similar scenario whereby a landing vehicle must maintain a line of sight until a certain distance from the landing site, and use the newly introduced *state-triggered constraints* to model this constraint.[28] We note that time-based criteria can equivalently be used in our framework. This allows the vehicle to be free of the line of sight constraint once it is sufficiently close to the landing site, while maintaining a continuous optimization framework (i.e., we do not resort to binary or integer variables). As a result, guidance trajectories can be designed with the line of sight explicitly enforced when it is required, and not enforced when it is not required.

State-triggered constraints (STCs) model constraints that are enforced only if a criterion conditioned on the state vector is met. While this work focuses in particular on a distance-triggered line of sight constraints, previous work has shown their applicability to speed-versus-angle-of-attack constraints for aerodynamic descent maneuvers.[28] Though not addressed here, state-triggered constraints can also be used to model state-based keep-out zone constraints for autonomous collision avoidance maneuvers, and minimum (or maximum) time-based constraints, among others. The continuous formulation of STCs offers a novel and elegant way to incorporate such constraints without resorting to engineering heuristics or mixed-integer programming. The resulting continuous optimal control problems are nonconvex; however we find that they can be readily solved using a successive convexification procedure. In this work, we use such a procedure to obtain feed-forward guidance solutions.

This paper is organized as follows. First, §II introduces our notation and description of 6-DoF motion using dual quaternions. Next, §III details the state and control constraints as a function of the dual quaternion and concludes with a statement of the problem that is solved. The solution method used in this work is briefly discussed in §IV. Lastly, §V provides a numerical example that highlights the distance-triggered line of sight constraint for a lunar landing scenario.

B. Notation

We denote the set of real numbers using $\mathbb{R}$, and use $\mathbb{R}_+$ and $\mathbb{R}_{++}$ to denote non-negative and positive real numbers, respectively. We use $0_{n\times m}$ to denote a matrix of size $n \times m$ whose entries are all zero, and I_n to denote the $n \times n$ identity matrix. We use $\textbf{spec}\{\cdot\}$ to denote the set of eigenvalues of a matrix and denote a symmetric positive semi-definite matrix M using $M \in \mathbb{S}^n_+$ or $M \succeq 0$. The skew symmetric cross product operator is defined as $\cdot^\times : \mathbb{R}^3 \to \mathbb{R}^{3\times 3}$ so that for any two vectors $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^3$ we have $\boldsymbol{a} \times \boldsymbol{b} = \boldsymbol{a}^\times \boldsymbol{b}$.
Unit quaternions are used to parameterize the set of isometric transformations (rotation matrices) of the Special Orthogonal group in 3-dimensions, $SO(3)$. We use $\mathbb{Q}$ to denote the quaternion manifold, and a general element $\boldsymbol{q} \in \mathbb{Q}$ is composed of a vector part, $\boldsymbol{q}_v$, and a scalar part q_0. We will write the set of unit quaternions as $\mathbb{Q}_u = \{\boldsymbol{q} \,|\, \boldsymbol{q} \cdot \boldsymbol{q} = 1\}$, where $\cdot$ denotes the Euclidean scalar product. The set of unit quaternions $\mathbb{Q}_u \subset \mathbb{Q}$ is said to form a three-dimensional hypersphere within the quaternion manifold. We refer to quaternions that have a zero scalar part as pure quaternions. Dual quaternions are denoted with a $\tilde{\cdot}$ to distinguish them from their quaternion counterparts.

II. Dual Quaternions and Rigid Body Motion

The Special Euclidean group, $SE(3)$, contains all possible configurations of a rigid body relative to a fixed inertial frame. Elements of $SE(3)$ can be described using 4×4 homogeneous transformation matrices,

$$SE(3) = \left\{ T \in \mathbb{R}^{4\times 4} \;\middle|\; T = \begin{bmatrix} C & \boldsymbol{r} \\ 0_{3\times 1} & 1 \end{bmatrix}, \, C \in SO(3), \, \boldsymbol{r} \in \mathbb{R}^3 \right\}. \tag{1}$$

Unit dual quaternions parameterize this set in a similar way that unit quaternions parameterize the Special Orthogonal group in three dimensions. Dual quaternions may be elegantly derived using the theory of Clifford algebras as in,[36] or by using geometric construction as in the original work of Clifford.[34] We adopt the notation that a dual quaternion is represented as

$$\tilde{\boldsymbol{q}} = \boldsymbol{q}_1 + \epsilon \boldsymbol{q}_2 \;\in \mathbb{Q}^2 \tag{2}$$

where $\boldsymbol{q}_1$, $\boldsymbol{q}_2 \in \mathbb{Q}$ are quaternions and $\epsilon \neq 0$ is termed the *dual unit* that satisfies the property $\epsilon^2 = 0$. We call $\boldsymbol{q}_1$ the *real part* and $\boldsymbol{q}_2$ the *dual part* of the dual quaternion $\tilde{\boldsymbol{q}}$. Dual quaternions are elements of the manifold $\mathbb{Q}^2$, an inherently different algebraic construct than the usual Euclidean vector space $\mathbb{R}^8$. Unit dual quaternions form a subset $\mathbb{Q}^2_u \subset \mathbb{Q}^2$ within the dual quaternion manifold. Under the usual scalar product, a dual quaternion of unit norm should satisfy

$$\tilde{\boldsymbol{q}} \cdot \tilde{\boldsymbol{q}} = (\boldsymbol{q}_1 + \epsilon \boldsymbol{q}_2) \cdot (\boldsymbol{q}_1 + \epsilon \boldsymbol{q}_2) = \boldsymbol{q}_1 \cdot \boldsymbol{q}_1 + \epsilon\, (\boldsymbol{q}_1 \cdot \boldsymbol{q}_2 + \boldsymbol{q}_2 \cdot \boldsymbol{q}_1) = 1 + \epsilon 0. \tag{3}$$

It can then be observed that the real and dual parts of $\tilde{\boldsymbol{q}}$ must satisfy

$$\boldsymbol{q}_1 \cdot \boldsymbol{q}_1 = 1 \quad \text{and} \quad \boldsymbol{q}_1 \cdot \boldsymbol{q}_2 = 0, \tag{4}$$

in order for $\tilde{\boldsymbol{q}}$ to be a unit dual quaternion. Consequently, we define the set of unit dual quaternions as

$$\mathbb{Q}^2_u \coloneqq \{ \tilde{\boldsymbol{q}} = \boldsymbol{q}_1 + \epsilon \boldsymbol{q}_2 \,|\, \boldsymbol{q}_1 \cdot \boldsymbol{q}_1 = 1, \; \boldsymbol{q}_1 \cdot \boldsymbol{q}_2 = 0 \}. \tag{5}$$

Note that the first constraint forces the real part of a unit dual quaternion to be a regular unit quaternion, i.e. an element of the three dimensional hypersphere $\mathbb{Q}_u$. The second constraint dictates that the dual part of a unit dual quaternion must be an element of the (three dimensional) tangent plane of the hypersphere at the point $\boldsymbol{q}_1$. As such, we may view the set of unit dual quaternions in (5) as the three dimensional hypersphere plus all of its tangent planes. Since attitude is encoded in the three dimensional hypersphere via unit quaternions, the additional three dimensional tangent planes shall provide a natural environment in which to encode position states.

A. Dual Quaternion Operations

Let $\tilde{\boldsymbol{q}}, \tilde{\boldsymbol{p}} \in \mathbb{Q}^2_u$ be two unit dual quaternions, and let $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{Q}_u$ be two unit quaternions. Recall that quaternions are composed of a vector part and a scalar part, which we shall denote by $\boldsymbol{a} = (\boldsymbol{a}_v, \, a_0)$ and $\boldsymbol{b} = (\boldsymbol{b}_v, \, b_0)$. We first define quaternion multiplication as

$$\boldsymbol{a} \otimes \boldsymbol{b} = \left(a_0 \boldsymbol{b}_v + b_0 \boldsymbol{a}_v + \boldsymbol{a}_v^\times \boldsymbol{b}, \; a_0 b_0 - \boldsymbol{a}_v \cdot \boldsymbol{b}_v \right), \tag{6}$$

from which dual quaternion multiplication is defined as

$$\tilde{\boldsymbol{q}} \otimes \tilde{\boldsymbol{p}} = \boldsymbol{q}_1 \otimes \boldsymbol{p}_1 + \epsilon \left(\boldsymbol{q}_1 \otimes \boldsymbol{p}_2 + \boldsymbol{q}_2 \otimes \boldsymbol{p}_1 \right). \tag{7}$$

Next, the quaternion cross product is defined as

$$\boldsymbol{a} \oslash \boldsymbol{b} = \left(a_0 \boldsymbol{b}_v + b_0 \boldsymbol{a}_v + \boldsymbol{a}_v^{\times} \boldsymbol{b},\ 0 \right), \tag{8}$$

which is used in turn to define the dual quaternion cross product[23, 24]

$$\tilde{\boldsymbol{q}} \oslash \tilde{\boldsymbol{p}} = \boldsymbol{q}_1 \oslash \boldsymbol{p}_1 + \epsilon \left(\boldsymbol{q}_1 \oslash \boldsymbol{p}_2 + \boldsymbol{q}_2 \oslash \boldsymbol{p}_1 \right). \tag{9}$$

We define the quaternion conjugate as $\boldsymbol{a}^* = (-\boldsymbol{a}_v,\, a_0)$, which permits the definition of the dual quaternion conjugate as

$$\tilde{\boldsymbol{q}}^* = \boldsymbol{q}_1^* + \epsilon \boldsymbol{q}_2^*. \tag{10}$$

For our purposes, we shall embed unit dual quaternions in the eight-dimensional Euclidean space $\mathbb{R}^8$ so that we may use more familiar matrix-vector analysis to manipulate them. To be clear, the elements of $\mathbb{Q}_u^2$ defined in (5) form a submanifold of $\mathbb{Q}^2$; we merely view them as elements of $\mathbb{R}^8$ for convenience. Using the natural isomorphism

$$\tilde{\boldsymbol{q}} = \boldsymbol{q}_1 + \epsilon \boldsymbol{q}_2 \in \mathbb{Q}_u^2 \quad \mapsto \quad \tilde{\boldsymbol{q}} = \begin{bmatrix} \boldsymbol{q}_1 \\ \boldsymbol{q}_2 \end{bmatrix} \in \mathbb{R}_u^8 := \{ \tilde{\boldsymbol{q}} \in \mathbb{R}^8 \,|\, \boldsymbol{q}_1^T \boldsymbol{q}_1 = 1 \text{ and } \boldsymbol{q}_1^T \boldsymbol{q}_2 = 0 \}, \tag{11}$$

we henceforth view unit dual quaternions as the subset $\mathbb{R}_u^8 \subset \mathbb{R}^8$ defined by the constraints $\boldsymbol{q}_1^T \boldsymbol{q}_1 = 1$ and $\boldsymbol{q}_1^T \boldsymbol{q}_2 = 0$. By virtue of the first four elements of $\tilde{\boldsymbol{q}} \in \mathbb{R}_u^8$, it follows that we view unit quaternions as four dimensional unit vectors embedded in $\mathbb{R}^4$. As such, we may now define special matrices to represent the operations in (7) and (9). Specifically, when viewed as an element of $\mathbb{R}^4$ we may rewrite (6) as

$$\boldsymbol{q} \otimes \boldsymbol{p} = [\boldsymbol{q}]_{\otimes}\, \boldsymbol{p} = [\boldsymbol{p}]_{\otimes}^*\, \boldsymbol{q} \tag{12}$$

where,

$$[\boldsymbol{q}]_{\otimes} := \begin{bmatrix} q_0 I_3 + \boldsymbol{q}_v^{\times} & \boldsymbol{q}_v \\ -\boldsymbol{q}_v^T & q_0 \end{bmatrix} \quad \text{and} \quad [\boldsymbol{p}]_{\otimes}^* := \begin{bmatrix} p_0 I_3 - \boldsymbol{p}_v^{\times} & \boldsymbol{p}_v \\ -\boldsymbol{p}_v^T & p_0 \end{bmatrix}.$$

Using these definitions, we can then rewrite (7) as

$$\tilde{\boldsymbol{q}} \otimes \tilde{\boldsymbol{p}} = [\tilde{\boldsymbol{q}}]_{\otimes}\, \tilde{\boldsymbol{p}} = [\tilde{\boldsymbol{p}}]_{\otimes}^* \tilde{\boldsymbol{q}}, \tag{13}$$

where,

$$[\tilde{\boldsymbol{q}}]_{\otimes} := \begin{bmatrix} [\boldsymbol{q}_1]_{\otimes} & 0_{4\times4} \\ [\boldsymbol{q}_2]_{\otimes} & [\boldsymbol{q}_1]_{\otimes} \end{bmatrix} \quad \text{and} \quad [\tilde{\boldsymbol{p}}]_{\otimes}^* := \begin{bmatrix} [\boldsymbol{p}_1]_{\otimes}^* & 0_{4\times4} \\ [\boldsymbol{p}_2]_{\otimes}^* & [\boldsymbol{p}_1]_{\otimes}^* \end{bmatrix}.$$

The quaternion and dual quaternion cross products can be rewritten using matrices via the same methods, see[23, 24] for more details. It is important to note that due to (11) the dual unit is no longer present in these expressions. The matrices in (13) are structured so that the matrix-vector multiplication gives the same result as the definition in (7)[a].

In deriving constraints as a function of a dual quaternion, we will make heavy use of the following two results. For $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c} \in \mathbb{R}^4$ we have

$$\boldsymbol{a}^T (\boldsymbol{b} \otimes \boldsymbol{c}) = \boldsymbol{b}^T (\boldsymbol{a} \otimes \boldsymbol{c}^*) = \boldsymbol{c}^T (\boldsymbol{b}^* \otimes \boldsymbol{a}), \tag{14}$$

and if $\boldsymbol{q}$ is a unit quaternion, then

$$\boldsymbol{a}^T \boldsymbol{b} = (\boldsymbol{a} \otimes \boldsymbol{q})^T (\boldsymbol{b} \otimes \boldsymbol{q}) = (\boldsymbol{q} \otimes \boldsymbol{a})^T (\boldsymbol{q} \otimes \boldsymbol{b}), \tag{15}$$

which is referred to as the *quaternion triple identity*.[23]

[a] The columns of these matrices can also be interpreted as the projection of the dual quaternion onto the basis vectors of the Clifford sub-algebra used to derive them.[36]

B. Rigid Body Motion

Let $\boldsymbol{r}$ denote the origin of a body-fixed coordinate frame $\mathcal{F}_{\mathcal{B}}$ with respect to an inertial coordinate frame $\mathcal{F}_{\mathcal{I}}$, and let $\boldsymbol{q} := \boldsymbol{q}_{\mathcal{B}\leftarrow\mathcal{I}} \in \mathbb{R}^4$ be the unit quaternion representing the orientation of $\mathcal{F}_{\mathcal{B}}$ with respect to $\mathcal{F}_{\mathcal{I}}$. The composition of a rotation and translation is represented using dual quaternion multiplication. As such, the unit dual quaternion that represents a translation by $\boldsymbol{r}$ followed by rotation $\boldsymbol{q}$ is given by (taking (11) into account)

$$\tilde{\boldsymbol{q}} = \begin{bmatrix} \boldsymbol{q} \\ \frac{1}{2}\boldsymbol{q} \otimes \boldsymbol{r}_{\mathcal{I}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{q} \\ \frac{1}{2}\boldsymbol{r}_{\mathcal{B}} \otimes \boldsymbol{q} \end{bmatrix} \in \mathbb{R}^8_u, \tag{16}$$

where $\boldsymbol{r}_{\mathcal{I}}$ and $\boldsymbol{r}_{\mathcal{B}}$ denote the coordinates of the vector $\boldsymbol{r}$ in the inertial and body frame respectively. The first expression describes a translation by $\boldsymbol{r}_{\mathcal{I}}$ followed by a rotation $\boldsymbol{q}$, whereas the second expression describes a rotation by $\boldsymbol{q}$ followed by a translation $\boldsymbol{r}_{\mathcal{B}}$. Figure 1 depicts each of these cases, and shows how they result in the same geometric definition of relative position and orientation.

The equivalence of the two expressions in (16) leads to the observation that

$$\boldsymbol{r}_{\mathcal{I}} = \boldsymbol{q} \otimes \boldsymbol{r}_{\mathcal{B}} \otimes \boldsymbol{q}^* \quad \text{and} \quad \boldsymbol{r}_{\mathcal{B}} = \boldsymbol{q}^* \otimes \boldsymbol{r}_{\mathcal{I}} \otimes \boldsymbol{q}. \tag{17}$$

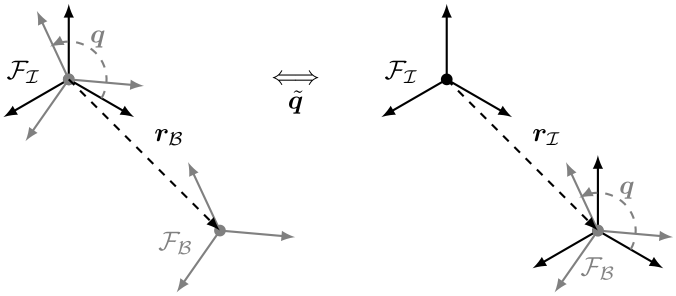

$\boldsymbol{q}$
$\mathcal{F}_{\mathcal{I}}$
$\boldsymbol{r}_{\mathcal{B}}$
$\mathcal{F}_{\mathcal{B}}$
$\Longleftrightarrow$
$\tilde{\boldsymbol{q}}$
$\mathcal{F}_{\mathcal{I}}$
$\boldsymbol{r}_{\mathcal{I}}$
$\boldsymbol{q}$
$\mathcal{F}_{\mathcal{B}}$

Figure 1: Rotation followed by translation is geometrically equivalent to a translation followed by a rotation. This leads to the two equivalent definitions of the dual quaternion in (16).

Similarly, we can represent the velocity states using dual quaternions as follows. Let $\boldsymbol{\omega}_{\mathcal{B}}, \boldsymbol{v}_{\mathcal{B}} \in \mathbb{R}^3$ denote the angular and linear velocity of a rigid body, respectively, expressed in body frame coordinates. By appending a zero to each vector, we represent these as pure quaternions and define the *dual velocity* to be

$$\tilde{\boldsymbol{\omega}} = \begin{bmatrix} \boldsymbol{\omega}_{\mathcal{B}} \\ \boldsymbol{v}_{\mathcal{B}} \end{bmatrix} \in \mathbb{R}^8. \tag{18}$$

Note that there is no requirement that this is unit dual quaternion.

1. *Kinematics & Dynamics*

We can take a time derivative of (16) to arrive at the dual quaternion kinematic equation.

$$\begin{aligned} \frac{d\tilde{\boldsymbol{q}}}{dt} = \frac{d}{dt}\begin{bmatrix} \boldsymbol{q} \\ \frac{1}{2}\boldsymbol{r}_{\mathcal{I}} \otimes \boldsymbol{q} \end{bmatrix} = \begin{bmatrix} \dot{\boldsymbol{q}} \\ \frac{1}{2}(\dot{\boldsymbol{r}}_{\mathcal{I}} \otimes \boldsymbol{q} + \boldsymbol{r}_{\mathcal{I}} \otimes \dot{\boldsymbol{q}}) \end{bmatrix} &= \begin{bmatrix} \frac{1}{2}\boldsymbol{q} \otimes \boldsymbol{\omega}_{\mathcal{B}} \\ \frac{1}{2}(\boldsymbol{v}_{\mathcal{I}} \otimes \boldsymbol{q} + \frac{1}{2}\boldsymbol{r}_{\mathcal{I}} \otimes \boldsymbol{q} \otimes \boldsymbol{\omega}_{\mathcal{B}}) \end{bmatrix} \\ &= \frac{1}{2}\begin{bmatrix} \boldsymbol{q} \\ \frac{1}{2}\boldsymbol{r}_{\mathcal{I}} \otimes \boldsymbol{q} \end{bmatrix} \otimes \begin{bmatrix} \boldsymbol{\omega}_{\mathcal{B}} \\ \boldsymbol{v}_{\mathcal{B}} \end{bmatrix} = \frac{1}{2}\tilde{\boldsymbol{q}} \otimes \tilde{\boldsymbol{\omega}} \end{aligned} \tag{19}$$

where we've used (17) and (13) to write $\boldsymbol{v}_{\mathcal{I}} \otimes \boldsymbol{q} = \boldsymbol{q} \otimes \boldsymbol{v}_{\mathcal{B}}$ and obtain the second to last equality.

The dynamics are obtained using the Newton-Euler equations in a rotating frame. We assume that neither the mass nor the inertia are constant. Rather, the mass, $m \in \mathbb{R}_+$, is assumed to vary as a linear function of the thrust magnitude according to

$$\dot{m} = -\alpha\|\boldsymbol{u}_{\mathcal{B}}\|_2, \qquad \alpha := \frac{1}{I_{\text{sp}}g_e} \tag{20}$$

where $\boldsymbol{u}_{\mathcal{B}} \in \mathbb{R}^3$ is the thrust vector in body frame coordinates, I_{sp} is the specific impulse of the rocket engine in vacuum and $g_e = 9.806\,\text{m/s}^2$ is the acceleration due to gravity at sea level on Earth. The inertia is assumed to be a function of the mass,

$$J \coloneqq J(m) \in \mathbb{S}^3_{++}, \tag{21}$$

where the specific form of $J(m)$ may vary with applications. We shall define the form assumed for our lunar landing application in §V. The dynamical equations of motion are then obtained by taking a derivative of the linear and angular momenta in the rotating body frame according to

$$\frac{d}{dt}(m\boldsymbol{v}_{\mathcal{B}}) = m\dot{\boldsymbol{v}}_{\mathcal{B}} + \boldsymbol{\omega}^{\times}_{\mathcal{B}}(m\boldsymbol{v}_{\mathcal{B}}) = \sum \boldsymbol{F}_{\mathcal{B}} + \boldsymbol{u}_{\mathcal{B}}, \tag{22a}$$

$$\frac{d}{dt}(J\boldsymbol{\omega}_{\mathcal{B}}) = \dot{J}\boldsymbol{\omega}_{\mathcal{B}} + J\dot{\boldsymbol{\omega}}_{\mathcal{B}} + \boldsymbol{\omega}^{\times}_{\mathcal{B}}(J\boldsymbol{\omega}_{\mathcal{B}}) = \sum \boldsymbol{T}_{\mathcal{B}} + \boldsymbol{r}^{\times}_{u}\boldsymbol{u}_{\mathcal{B}}, \tag{22b}$$

where $\boldsymbol{F}_{\mathcal{B}} \in \mathbb{R}^3$ and $\boldsymbol{T}_{\mathcal{B}} \in \mathbb{R}^3$ represent the externally applied forces and torques, respectively, and $\boldsymbol{r}_u$ denotes the constant vector from the vehicle's center of mass to the point where the thrust is applied. We assume in this work that $\sum \boldsymbol{F}_{\mathcal{B}} = \boldsymbol{g}_{\mathcal{B}}$ and $\sum \boldsymbol{T}_{\mathcal{B}} = 0$, where $\boldsymbol{g}_{\mathcal{B}}$ is the force due to gravity in the body frame. Note that our definition of the mass depletion dynamics in (20) allows us to capture momentum changes due to mass variability in (22a) using the term $\boldsymbol{u}_{\mathcal{B}}$; see[37] for details.

Combining the expressions in (22) with the definition of the dual velocity in (18) leads us to express the equations of motion in terms of the dual velocity as

$$\boldsymbol{J}\dot{\tilde{\boldsymbol{\omega}}} + \left(\tilde{\boldsymbol{\omega}} \oslash \boldsymbol{J} + \dot{\boldsymbol{J}}E_r\right)\tilde{\boldsymbol{\omega}} = \Phi\boldsymbol{u}_{\mathcal{B}} + \tilde{\boldsymbol{g}}_{\mathcal{B}}, \tag{23}$$

where $E_r = \mathbf{diag}\,\{I_4, 0_{4\times 4}\}$ and,

$$\boldsymbol{J} = \left[\begin{array}{cc|cc} \multicolumn{2}{c|}{0_{4\times 4}} & mI_3 & 0 \\ & & 0 & 1 \\ \hline J & 0 & \multicolumn{2}{c}{0_{4\times 4}} \\ 0 & 1 & & \end{array}\right]_{8\times 8} \qquad \Phi = \left[\begin{array}{cc} I_3 & 0 \\ 0 & 1 \\ \hline \boldsymbol{r}^{\times}_{u} & 0_{3\times 1} \\ 0_{1\times 3} & 0 \end{array}\right]_{8\times 4} \qquad \tilde{\boldsymbol{g}}_{\mathcal{B}} = \begin{bmatrix} \boldsymbol{g}_{\mathcal{B}} \\ 0 \end{bmatrix}_{8\times 1}.$$

We encourage the reader to refer to[23, 24, 32, 33] for more details on rigid body kinematics and dynamics using dual quaternions.

III. Problem Statement

This section details the ingredients necessary to state the continuous-time powered descent guidance problem that we consider in this paper. Having already stated the equations of motion, this section focuses specifically on the state and control constraints imposed during powered descent maneuvers. We now assume that the inertial frame can be described by a set of three orthonormal vectors $\{\boldsymbol{x}_{\mathcal{I}},\, \boldsymbol{y}_{\mathcal{I}},\, \boldsymbol{z}_{\mathcal{I}}\}$ such that $\boldsymbol{z}_{\mathcal{I}}$ points locally up, $\boldsymbol{x}_{\mathcal{I}}$ points south and $\boldsymbol{y}_{\mathcal{I}}$ points east. The body frame can similarly be described by the orthonormal vectors $\{\boldsymbol{x}_{\mathcal{B}},\, \boldsymbol{y}_{\mathcal{B}},\, \boldsymbol{z}_{\mathcal{B}}\}$ that are assumed to describe the vehicle's principal axes of inertia. We assume that $\boldsymbol{z}_{\mathcal{B}}$ is the vector closest to the vehicle's vertical axis.

This section is organized as follows. First, §A introduces the state and control constraints that we consider to form a baseline problem. Next, §B introduces the concept of state-triggered constraints, and §C details their application to distance-triggered line of sight constraints. We conclude in §D with a complete statement of the optimal control problem that is to be solved.

A. Baseline Problem

Powered descent guidance problems are subject to several state and control constraints to ensure that trajectories adhere to both safety considerations and vehicle limitations. We begin our formulation of the baseline problem by discussing the control constraints.

1. *Control Constraints*

The control authority of a rocket-powered vehicle is limited by the rocket engine it is equipped with. These complex machines necessitate, at times, operation in restricted regimes of thrust and maneuverability. For example, the main engines of Apollo-era landers could operate either at 93% thrust, or in the permitted thrust interval of 11% to 65% of the rated thrust value.[5] The intervening thrust regions were forbidden to avoid oxidizer and fuel cavitation, which could lead to the deterioration of propulsion system components. We have assumed (in (23)) that main engine thrust is the only source of actuation, and thus do not consider the additional constraints imposed by reaction control systems.

To model permitted thrust regions, we place restrictions on the norm of the thrust vector as

$$0 < u_{\min} \leq \|\boldsymbol{u}_\mathcal{B}\| \leq u_{\max}, \tag{24}$$

where $[u_{\min}, u_{\max}] \subset \mathbb{R}_{++}$ denotes the permitted thrust interval. This constraint implies that once ignited, the engines are not turned off until touchdown.

Next, limited maneuverability of the engine is modeled with a gimbal angle constraint. The gimbal angle, $\delta \in [0, 90°)$, is defined as the total angular deflection of the thrust vector from its nominal position. We express a gimbal angle constraint as the following second order cone constraint,

$$-\boldsymbol{z}_\mathcal{B}^T\boldsymbol{u}_\mathcal{B} + \|\boldsymbol{u}_\mathcal{B}\|_2 \cos\delta_{\max} \leq 0 \iff \|\boldsymbol{u}_\mathcal{B}\|_2 \leq \bar{\boldsymbol{z}}_\mathcal{B}^T\boldsymbol{u}_\mathcal{B} \tag{25}$$

where $\delta_{\max} \in [0, 90°)$ is the maximum allowable gimbal angle and $\bar{\boldsymbol{z}}_\mathcal{B} = (1/\cos\delta_{\max})\,\boldsymbol{z}_\mathcal{B}$.

The final constraint imposed on the control is a rate constraint that ensures commanded thrust vectors do not change too rapidly for the engine to follow. This is formulated as

$$\frac{\Delta\boldsymbol{u}_\mathcal{B}}{\Delta t} \leq \Delta\boldsymbol{u}_{\max} \tag{26}$$

where $\Delta\boldsymbol{u}_{\max} \in \mathbb{R}^3_{++}$ denotes the vector of maximum allowable thrust deviations over a specified time interval $\Delta t \in \mathbb{R}_{++}$.

2. *State Constraints*

We now proceed to describe the state constraints enforced in the baseline problem formulation. Powered descent maneuvers must not use more fuel than is stored on board, a constraint enforced on the mass of the spacecraft using

$$m \geq m_{\text{dry}} \tag{27}$$

where $m_{\text{dry}} \in \mathbb{R}_{++}$ is the dry mass of the vehicle. Next, we use a glide slope cone to ensure the vehicle's altitude lies above the surface of the planet, while also ensuring sufficient elevation at large distances from the landing site. If $\boldsymbol{r}_\mathcal{I}$ denotes the inertial position of the vehicle, then we define the glide slope angle to be the angle formed between $\boldsymbol{r}_\mathcal{I}$ and $\boldsymbol{z}_\mathcal{I}$ and denote it by $\gamma \in [0, 90°]$. A glide slope constraint enforces that the glide slope angle must be less than some prescribed maximum value, $\gamma_{\max}$. Formally, the constraint is expressed as

$$-\boldsymbol{r}_\mathcal{I}^T\boldsymbol{z}_\mathcal{I} + \|\boldsymbol{r}_\mathcal{I}\|_2 \cos\gamma_{\max} \leq 0. \tag{28}$$

We can express this with dual quaternions by using (15) to write

$$\|\boldsymbol{r}_\mathcal{I}\| = \left(\boldsymbol{r}_\mathcal{I}^T\boldsymbol{r}_\mathcal{I}\right)^{1/2} = \left((\boldsymbol{r}_\mathcal{I}\otimes\boldsymbol{q})^T(\boldsymbol{r}_\mathcal{I}\otimes\boldsymbol{q})\right)^{1/2} = \left\|\begin{bmatrix}0_{4\times1}\\ \boldsymbol{r}_\mathcal{I}\otimes\boldsymbol{q}\end{bmatrix}\right\|_2 = \left\|2\begin{bmatrix}0_{4\times1}\\ \frac{1}{2}\boldsymbol{r}_\mathcal{I}\otimes\boldsymbol{q}\end{bmatrix}\right\|_2 = \|2E_d\tilde{\boldsymbol{q}}\| \tag{29}$$

where $E_d = \textbf{diag}\,\{0_{4\times4}, I_4\}$. Another application of (15) yields

$$\begin{aligned}
\boldsymbol{r}_\mathcal{I}^T\boldsymbol{z}_\mathcal{I} = (\boldsymbol{r}_\mathcal{I}\otimes\boldsymbol{q})^T(\boldsymbol{z}_\mathcal{I}\otimes\boldsymbol{q}) = (\boldsymbol{r}_\mathcal{I}\otimes\boldsymbol{q})^T[\boldsymbol{z}_\mathcal{I}]_\otimes\boldsymbol{q} &= \frac{1}{2}(\boldsymbol{r}_\mathcal{I}\otimes\boldsymbol{q})^T[\boldsymbol{z}_\mathcal{I}]_\otimes\boldsymbol{q} + \frac{1}{2}(\boldsymbol{r}_\mathcal{I}\otimes\boldsymbol{q})^T[\boldsymbol{z}_\mathcal{I}]_\otimes\boldsymbol{q}, \\
&= \begin{bmatrix}\boldsymbol{q}\\ \frac{1}{2}\boldsymbol{r}_\mathcal{I}\otimes\boldsymbol{q}\end{bmatrix}^T \begin{bmatrix}0_{4\times4} & [\boldsymbol{z}_\mathcal{I}]_\otimes^T\\ [\boldsymbol{z}_\mathcal{I}]_\otimes & 0_{4\times4}\end{bmatrix} \begin{bmatrix}\boldsymbol{q}\\ \frac{1}{2}\boldsymbol{r}_\mathcal{I}\otimes\boldsymbol{q}\end{bmatrix} \\
&= \tilde{\boldsymbol{q}}^T M_g\tilde{\boldsymbol{q}}.
\end{aligned} \tag{30}$$

Using (29) and (30), we arrive at the following proposition.

Proposition 1. *The constraint* (28) *is expressed in terms of the dual quaternion as*

$$c_g(\tilde{\boldsymbol{q}}) \coloneqq -\tilde{\boldsymbol{q}}^T M_g \tilde{\boldsymbol{q}} + \|2E_d\tilde{\boldsymbol{q}}\|_2 \cos\gamma_{\max} \leq 0. \tag{31}$$

Moreover, the function $c_g : \mathbb{R}^8_u \to \mathbb{R}$ *is convex over the domain* $\mathbf{dom}\, c_g = \mathbb{R}^8_u \bigcap \left\{\tilde{\boldsymbol{q}} \,|\, \tilde{\boldsymbol{q}}^T\tilde{\boldsymbol{q}} \leq 1 + \frac{1}{4}\Delta^2\right\}$*, where* $\|\boldsymbol{r}_{\mathcal{I}}\|_2 \leq \Delta$ *is an upper bound on the distance from the landing site.*

Proof. See[23, 24] for the proof. □

The next constraint we consider is a tilt angle constraint. The vehicle's tilt angle is the angle formed between $\boldsymbol{z}_{\mathcal{I}}$ and $\boldsymbol{z}_{\mathcal{B}}$, denoted by $\theta \in [0, 90°)$, and constrained to be less than some prescribed value. By expressing $\boldsymbol{z}_{\mathcal{B}}$ in inertial coordinates using (17), a tilt constraint is written in terms of the attitude quaternion as

$$-\boldsymbol{z}_{\mathcal{I}}^T\left(\boldsymbol{q} \otimes \boldsymbol{z}_{\mathcal{B}} \otimes \boldsymbol{q}^*\right) + \cos\theta_{\max} \leq 0, \tag{32}$$

where $\theta_{\max} \in [0, 90°)$ is the maximum allowable tilt angle. Note that $\boldsymbol{z}_{\mathcal{I}}$ and $\boldsymbol{z}_{\mathcal{B}}$ are being treated as pure quaternions in (32). To write this constraint as a function of the dual quaternion, we first use (15) to write

$$\boldsymbol{z}_{\mathcal{I}}^T(\boldsymbol{q} \otimes \boldsymbol{z}_{\mathcal{B}} \otimes \boldsymbol{q}^*) = (\boldsymbol{z}_{\mathcal{I}} \otimes \boldsymbol{q})^T(\boldsymbol{q} \otimes \boldsymbol{z}_{\mathcal{B}} \otimes \boldsymbol{q}^* \otimes \boldsymbol{q}) = (\boldsymbol{z}_{\mathcal{I}} \otimes \boldsymbol{q})^T(\boldsymbol{q} \otimes \boldsymbol{z}_{\mathcal{B}}) = \boldsymbol{q}^T(\boldsymbol{z}_{\mathcal{I}} \otimes \boldsymbol{q} \otimes \boldsymbol{z}_{\mathcal{B}}^*) = \boldsymbol{q}^T[\boldsymbol{z}_{\mathcal{I}}]_{\otimes}[\boldsymbol{z}_{\mathcal{B}}^*]_{\otimes}^*\boldsymbol{q},$$

with which we can rewrite (32) to be

$$\tilde{\boldsymbol{q}}^T M_t \tilde{\boldsymbol{q}} + \cos\theta_{\max} \leq 0, \quad M_t = \begin{bmatrix} [\boldsymbol{z}_{\mathcal{I}}]_{\otimes}[\boldsymbol{z}_{\mathcal{B}}]_{\otimes}^* & 0_{4\times4} \\ 0_{4\times4} & 0_{4\times4} \end{bmatrix} \tag{33}$$

by noting that $-[\boldsymbol{z}_{\mathcal{B}}^*]_{\otimes}^* = [\boldsymbol{z}_{\mathcal{B}}]_{\otimes}^*$. Since both $\boldsymbol{z}_{\mathcal{I}}$ and $\boldsymbol{z}_{\mathcal{B}}$ are unit vectors, we find that the eigenvalues of M_t lie in the set $\{-1,\, 0,\, 1\}$, and thus it is a symmetric but indefinite matrix.

Proposition 2. *The tilt constraint* (32) *is equivalently expressed in terms of the dual quaternion as*

$$c_t(\tilde{\boldsymbol{q}}) \coloneqq \tilde{\boldsymbol{q}}^T \tilde{M}_t \tilde{\boldsymbol{q}} + \cos\theta_{\max} - \zeta \leq 0, \quad \tilde{M}_t = M_t + \zeta E_r \tag{34}$$

where $E_r = \mathbf{diag}\left\{I_4, 0_{4\times4}\right\}$ *and* ζ *is a constant. Moreover, the function* c_t *is convex for all* $\tilde{\boldsymbol{q}} \in \mathbb{R}^8_u$ *when* $\theta \in (0, 90°]$ *and* $\zeta \geq 1$.

Proof. The proof follows by noting that $\tilde{\boldsymbol{q}}^T E_r \tilde{\boldsymbol{q}} - 1 = 0$, due to the fact that the real part of the dual quaternion is a unit quaternion. Hence for any $\zeta > 0$ we have $\zeta\big(\tilde{\boldsymbol{q}}^T E_r \tilde{\boldsymbol{q}} - 1\big) = 0$. Adding this term to (33) yields

$$\begin{aligned}\tilde{\boldsymbol{q}}^T M_t \tilde{\boldsymbol{q}} + \cos\theta_{\max} + \zeta\tilde{\boldsymbol{q}}^T E_r \tilde{\boldsymbol{q}} - \zeta &\leq 0 \\ \tilde{\boldsymbol{q}}^T (M_t + \zeta E_r)\tilde{\boldsymbol{q}} + \cos\theta_{\max} - \zeta &\leq 0 \\ \tilde{\boldsymbol{q}}^T \tilde{M}_t \tilde{\boldsymbol{q}} + \cos\theta_{\max} - \zeta &\leq 0.\end{aligned}$$

Now the eigenvalues of $\tilde{M}_t$ can be obtained as a function of the parameter ζ. Specifically, we find that

$$\mathbf{spec}\left\{\tilde{M}_t\right\} = \{-1 + \zeta,\, 0,\, 1 + \zeta\}.$$

It is clear then that for $\zeta \geq 1$, we have $\tilde{M}_t \succeq 0$. When this is the case, the Hessian of c_t is positive semidefinite for all $\tilde{\boldsymbol{q}} \in \mathbb{R}^8_u$ and $\theta \in [0, 90°)$. Hence c_t is convex over this domain. □

The final state constraint considered part of our baseline problem is a bound on the angular rate of the vehicle. We note that due to our assumption that the main engine is the only actuator producing torque, angular accelerations can be limited by suitable choice of $\Delta\boldsymbol{u}_{\max}$ in (26). We impose an addition angular rate constraint of the form

$$\|\boldsymbol{\omega}_{\mathcal{B}}\|_{\infty} \leq \omega_{\max} \tag{35}$$

where $\omega_{\max} \in \mathbb{R}_{++}$ is the maximum allowable angular velocity about any axis.

B. State-Triggered Constraints

We now detail the *state-triggered constraints* and their continuous formulation. State-triggered constraints (STCs) were introduced in[28] and represent constraints that are enforced only when a state-dependent criterion is met. When formulated using the continuous variables of an optimization problem, these constraints model an if-statement conditioned on the solution variables that they are constraining. In a trajectory generation problem, optimal solutions are thus obtained with a simultaneous understanding of how the constraint affects the solution, and of how the solution enables or disables the constraint.

An STC is composed of two functions, the *trigger function* and the *constraint function.* Using $\boldsymbol{z} \in \mathbb{R}^{n_z}$ to denote an arbitrary solution vector, we denote these by $g : \mathbb{R}^{n_z} \to \mathbb{R}$ and $c : \mathbb{R}^{n_z} \to \mathbb{R}$, respectively. Vector-valued trigger and constraint functions are addressed in.[38] The constraint function is to be conditionally enforced based on the value of the trigger function, and thus we formally define the STC to be

$$g(\boldsymbol{z}) < 0 \quad \Rightarrow \quad c(\boldsymbol{z}) \leq 0. \tag{36}$$

We refer to $g(\boldsymbol{z}) < 0$ as the *trigger condition* and $c(\boldsymbol{z}) \leq 0$ as the *constraint condition.* If the trigger function is non-negative, then the optimization variable $\boldsymbol{z}$ is not subject to the constraint condition. If however, the trigger function becomes strictly negative (i.e., becomes active), then the state is subject to the constraint condition. The feasible region in the $g(\boldsymbol{z})$-$c(\boldsymbol{z})$ space is depicted in the bottom two axes of Figure 2.

1. *Continuous Formulation*

In order to incorporate constraints of the form (36) into a continuous optimization problem, they must be represented using continuous variables. To do so, we introduce the auxiliary variable $\sigma \in \mathbb{R}_{++}$ and the following system of equations

$$\begin{aligned} \sigma &\geq 0, && (37a)\\ g(\boldsymbol{z}) + \sigma &\geq 0, && (37b)\\ \sigma c(\boldsymbol{z}) &\leq 0. && (37c) \end{aligned}$$

The idea is to represent the (binary) logical implication in (36) as the outcome of this system of equations in continuous variables. The equations in (37) are such that when the trigger condition is satisfied, (37b) and (37a) imply that $\sigma > 0$. It follows then that (37c) holds if and only if constraint condition is met. Therefore (37) is logically equivalent to (36). We refer to this continuous formulation as *continuous state-triggered constraints* (cSTCs).

The formulation in (37) does not, however, admit a unique solution for σ. As illustrated in Figure 2, when $g(\boldsymbol{z}) < 0$, the auxiliary variable satisfies $\sigma \in [-g(\boldsymbol{z}), \infty)$. Moreover, when the trigger condition is not met, σ is free to be any non-negative number, including a non-zero value that enforces the constraint condition. We have found that this ambiguity can both inadvertently constrain the solution variables and cause numerical issues during solutions due to the unboundedness of the slack variable.

To alleviate these problems, we introduce an altered set of constraints motivated by the linear complementarity problem (LCP).[39] The new set of constraints form a complementarity condition between the left-hand sides of (37b) and (37a), and results in

$$\begin{aligned} 0 \leq \sigma \perp \big(g(\boldsymbol{z}) + \sigma\big) &\geq 0, && (38a)\\ \sigma c(\boldsymbol{z}) &\leq 0, && (38b) \end{aligned}$$

where the notation in (38a) represents the trio of constraints $\sigma \geq 0$, $g(\boldsymbol{z}) + \sigma \geq 0$ and $s \cdot \big(g(\boldsymbol{z}) + \sigma\big) = 0$. For a given $\boldsymbol{z}$, (38a) defines an LCP in σ, and we refer to this formulation as the *projected continuous state-triggered constraints* (Projected cSTCs). This problem has a unique solution that varies continuously as a function of $\boldsymbol{z}$,[39] and can be solved for analytically as

$$\sigma^* := -\min(g(\boldsymbol{z}),\, 0). \tag{39}$$

Substituting σ^* into (38) guarantees the satisfaction of (38a), and thus we may remove it from the formulation. The result is that (38b) becomes a single, logically equivalent, constraint to (36) that uses only continuous variables. We define the constraint (38b) with σ^* to be

$$h(\boldsymbol{z}) := -\min(g(\boldsymbol{z}), 0) \cdot c(\boldsymbol{z}) \leq 0. \tag{40}$$

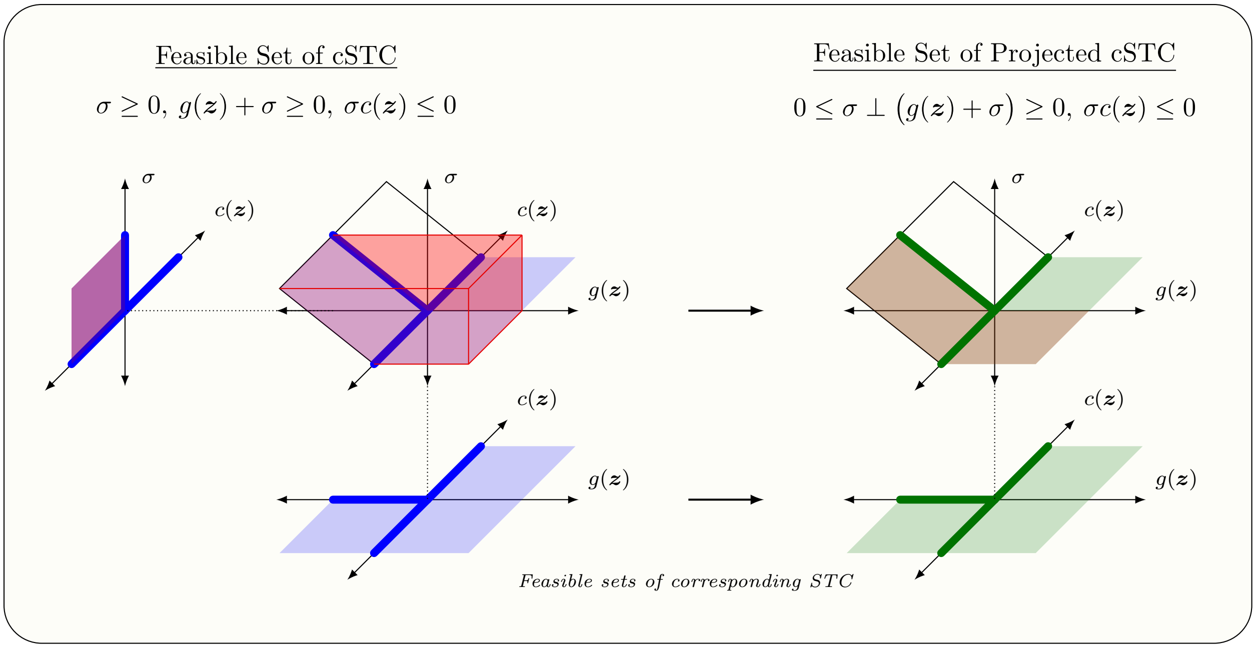

Feasible Set of cSTC
$\sigma \geq 0,\ g(\boldsymbol{z}) + \sigma \geq 0,\ \sigma c(\boldsymbol{z}) \leq 0$
Feasible Set of Projected cSTC
$0 \leq \sigma \perp \big(g(\boldsymbol{z}) + \sigma\big) \geq 0,\ \sigma c(\boldsymbol{z}) \leq 0$
σ
$c(\boldsymbol{z})$
$g(\boldsymbol{z})$
Feasible sets of corresponding STC

Figure 2: Geometric interpretation of cSTCs with an inequality constraint condition. The blue axes on the left represent the cSTCs in (37), while the green axes on the right represent the Projected cSTCs in (38). The red regions depict the feasible space of the auxiliary variable σ, with the *volume* observed in the central axes portraying the ambiguity noted for (37).

The geometry resulting from this analytical solution for σ^* is illustrated in the upper-rightmost axes of Figure 2. The removal of the ambiguity from (37) was accomplished by the additional complementarity constraint by effectively removing the volume of the red region seen in the central blue set of axes that denote the feasible space of (37).

C. Distance-Triggered Line of Sight Constraints

We now consider the application of STCs to line of sight constraints. Line of sight (LOS) constraints restrict the permissible attitude and position of the vehicle such that the vector to the landing site is aligned with a particular boresight direction within some maximum angle. We denote the LOS angle by $\xi \in [0, 180°]$, and similarly denote its maximum allowable angle by $\xi_{\max}$. We highlight that this type of constraint has been considered in powered descent problems in past work,[23, 24] however it was applied at all times during a trajectory. The current methodology is required for powered descent problems with tight LOS bounds imposed by vision-based sensors for two reasons. First, the LOS angle is a constant offset from the sum of the glide slope and tilt angles. As a result, a small LOS angle can severely limit the maneuverability of the vehicle, thus limiting the set of feasible initial conditions. Second, long trajectories may require pointing to different regions of a planets surface to acquire navigation data, something that would typically require the solution of two problems that are properly joined. The use of STCs can handle these scenarios in a single optimization framework.

To formulate a distance-triggered LOS constraint, we propose a trigger function $g_l : \mathbb{R}^8_u \to \mathbb{R}$ of the form

$$g_l(\tilde{\boldsymbol{q}}) := d - \|2E_d\tilde{\boldsymbol{q}}\|_2 \tag{41}$$

where from (29) the second term is equivalent to the normed distance from the landing site. This function is seen to satisfy the trigger condition for distances *strictly greater than* $d \in \mathbb{R}$. We note that $d = 0$ recovers the previous LOS constraints considered in the literature.[23, 24]

To construct the constraint function, consider a unit vector in body coordinates, $\boldsymbol{p}_{\mathcal{B}} \in \mathbb{R}^3$, that defines the boresight of an optical sensor or window. The line of sight angle ξ is defined as the angle between $\boldsymbol{p}_{\mathcal{B}}$ and

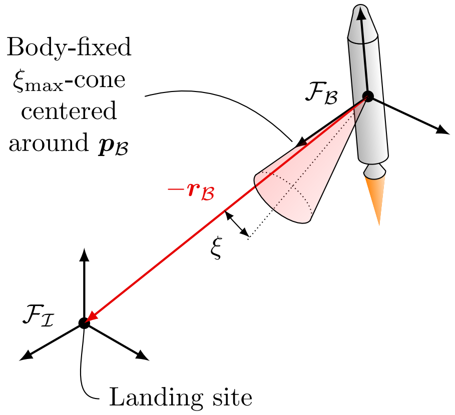

Body-fixed
$\xi_{\max}$-cone
centered
around $\boldsymbol{p}_{\mathcal{B}}$
$\mathcal{F}_{\mathcal{B}}$
$-\boldsymbol{r}_{\mathcal{B}}$
ξ
$\mathcal{F}_{\mathcal{I}}$
Landing site

Figure 3: A line of sight constraint to be conditionally imposed based on the distance from the landing site.

$-\boldsymbol{r}_{\mathcal{B}}$, and the line of sight constraint can be expressed as

$$\boldsymbol{r}_{\mathcal{B}}^T \boldsymbol{p}_{\mathcal{B}} + \|\boldsymbol{r}_{\mathcal{B}}\|_2 \cos \xi_{\max} \leq 0, \tag{42}$$

and is visualized in Figure 3. To write this constraint with dual quaternions, we note that the same trick from (30) can be used on the first term in (42) to yield

$$\boldsymbol{r}_{\mathcal{B}}^T \boldsymbol{p}_{\mathcal{B}} = (\boldsymbol{q} \otimes \boldsymbol{r}_{\mathcal{B}})^T (\boldsymbol{q} \otimes \boldsymbol{p}_{\mathcal{B}}) = \tilde{\boldsymbol{q}}^T M_l \tilde{\boldsymbol{q}}, \quad M_l = \begin{bmatrix} 0_{4\times 4} & [\boldsymbol{p}_{\mathcal{B}}]_{\otimes}^{*\,T} \\ [\boldsymbol{p}_{\mathcal{B}}]_{\otimes}^{*} & 0_{4\times 4} \end{bmatrix}$$

which in conjunction with (29) leads us to the following proposition.

Proposition 3. *The line of sight constraint* (42) *is expressed in terms of the dual quaternion as*

$$c_l(\tilde{\boldsymbol{q}}) := \tilde{\boldsymbol{q}}^T M_l \tilde{\boldsymbol{q}} + \|2 E_d \tilde{\boldsymbol{q}}\|_2 \cos \xi_{\max} \leq 0 \tag{43}$$

Moreover, the function $c_l : \mathbb{R}^8_u \to \mathbb{R}$ *defined above is convex over the domain* $\textbf{dom}\, c_l = \mathbb{R}^8_u \bigcap \left\{ \tilde{\boldsymbol{q}} \,|\, \tilde{\boldsymbol{q}}^T \tilde{\boldsymbol{q}} \leq 1 + \frac{1}{4}\Delta^2 \right\}$*, where* $\|\boldsymbol{r}_{\mathcal{I}}\|_2 \leq \Delta$ *is an upper bound on the distance from the landing site.*

Proof. See[23, 24] for the proof. □

The distance triggered line of sight constraint can thus be expressed using (40) as

$$h_l(\tilde{\boldsymbol{q}}) := -\min(g_l(\tilde{\boldsymbol{q}}), 0) \cdot c_l(\tilde{\boldsymbol{q}}) \leq 0. \tag{44}$$

Remark III.1. *The formulation presented here is equally applicable to time-triggered constraints for problems where either the final or ignition time is a variable. In this case, it is a simple matter to construct the trigger function to model an arbitrary interval of time as a function of the ignition time* t_0*, the final time* t_f *and the current time* t*.*

Remark III.2. *The state-triggered constraints are formulated in a way that avoids using iterative schemes that update the constrained temporal intervals based on the value of the trigger function from a previous iteration. The advantage lies primarily in allowing the optimization process to understand how adjusting the state will enable or disable the constraint, a feature not enjoyed by these heuristic methods.*

D. Problem Statement

We conclude this section with a full statement of the problem to be solved. We state the problem as a free-final time continuous optimal control problem subject to nonlinear dynamics and both state and control constraints. We wish to find the burn time, $t_f \in \mathbb{R}_{++}$ and the piecewise continuous thrust commands $\boldsymbol{u}_{\mathcal{B}}(t)$ for $t \in [0, t_f]$ that solve the following optimal control problem.

$$\begin{array}{llll}
\displaystyle\min_{t_f,\,\boldsymbol{u}_{\mathcal{B}}(\cdot)} & & -\,m(t_f) & \\
\text{subject to:} & \dot{m} & = -\alpha\|\boldsymbol{u}_{\mathcal{B}}\|_2 & (20) \\
 & \dot{\tilde{\boldsymbol{q}}} & = \dfrac{1}{2}\tilde{\boldsymbol{q}} \otimes \tilde{\boldsymbol{\omega}} & (19) \\
 & \boldsymbol{J}\dot{\tilde{\boldsymbol{\omega}}} & = \Phi\boldsymbol{u}_{\mathcal{B}} + \tilde{\boldsymbol{g}}_{\mathcal{B}} - \big(\tilde{\boldsymbol{\omega}} \oslash \boldsymbol{J} + \dot{\boldsymbol{J}}E_r\big)\tilde{\boldsymbol{\omega}} & (23) \\
 & u_{\min} & \leq \|\boldsymbol{u}_{\mathcal{B}}\|_2 \leq u_{\max} & (24) \\
 & \|\boldsymbol{u}_{\mathcal{B}}\|_2 & \leq \bar{\boldsymbol{z}}_{\mathcal{B}}^T\boldsymbol{u}_{\mathcal{B}} & (25) \\
 & \dfrac{\Delta\boldsymbol{u}_{\mathcal{B}}}{\Delta t} & \leq \Delta\boldsymbol{u}_{\max} & (26) \\
 & m_{\text{dry}} & \leq m & (27) \\
 & c_g(\tilde{\boldsymbol{q}}) & \leq 0 & (31) \\
 & c_t(\tilde{\boldsymbol{q}}) & \leq 0 & (34) \\
 & \|\boldsymbol{\omega}_{\mathcal{B}}\|_\infty & \leq \omega_{\max} & (35) \\
 & h_l(\tilde{\boldsymbol{q}}) & \leq 0 & (44)
\end{array} \qquad \text{(Problem 1)}$$

Finally, we note that the *baseline problem* is Problem 1 with constraint (44) removed.

IV. Solution Method

This section will briefly describe the solution method for Problem 1. In,[28] a method was presented to convert a general free-final time nonlinear continuous-time optimal control problem into a sequence of fixed-final time convex discrete-time parameter optimization subproblems. The general formulation was specified for a powered descent guidance problem using Cartesian variables, however the steps remain the same when using dual quaternions. The algorithm works by iteratively solving these subproblems until a converged solution is attained. Each iteration can be decomposed into two main steps; a propagation step and a solve step. The propagation step is responsible for obtaining a convex approximation to Problem 1, while the solve step solves this subproblem to full optimality using well-studied algorithms.[40–43] The solve step's solution is then used during the next iteration's propagation step to obtain an improved approximation of Problem 1. Upon convergence, the algorithm is designed to obtain solutions that adhere exactly to the nonlinear dynamics of the problem, while approximating constraint feasibility and local optimality. For brevity, we do not repeat the steps to obtain a convex approximation of Problem 1 here. We refer the reader to[28] for these details, but provide a summary of the key steps and assumptions made for the current work.

In,[44] several methods for the propagation step are compared in a Monte Carlo campaign, and the methods that appear suitable for real-time implementations were discussed. In particular, the trade study of direct transcription methods in[44] compared methods that parameterize the control trajectory to global pseudospectral methods (that parameterize the state and control trajectories). It was found that the former group of methods result in a more sparse optimization, leading to faster solution times. When the accuracy of the solutions was taken into account, a piecewise linear approximation of the control signal was seen to perform the best over the scenarios tested. As a result of this analysis, we proceed in this work with this assumption in the propagation step.

We discretize the time interval $[0, t_f] \subset \mathbb{R}_+$ into $N-1$ evenly spaced temporal intervals such that $0 = t_0 < t_1 < \ldots < t_k < \ldots < t_{N-1} = t_f$. The control trajectory can then be expressed by a set of N vectors $\boldsymbol{u}_{\mathcal{B},k} \in \mathbb{R}^3$ such that $\boldsymbol{u}_{\mathcal{B},k} = \boldsymbol{u}_{\mathcal{B}}(t_k)$. The continuous-time signal is reconstructed using the piecewise linear interpolation scheme between these discrete values. Each solve step therefore obtains the value of the control signal at discrete temporal points along the trajectory.

During the propagation step, the nonlinear dynamics in (19), (20) and (23) are approximated by a first order Taylor series centered about the previous iteration's solve step solution. We point out that due to the chosen discretization method,[28,44] the behaviour of the original nonlinear dynamics during inter-node temporal intervals is captured even in the discrete solution. All constraints in Problem 1 are enforced only at the discrete temporal points, and non-convex state and control constraints are approximated by a first order Taylor series expansion centered about the previous iteration's solve step solution.

V. Numerical Examples

This section provides example solutions to both the baseline problem and Problem 1 discussed in §III. These examples are intended to demonstrate the main contribution of this paper, namely the distance-triggered line of sight constraint. The scenario is modeled as a lunar landing whereby the vehicle must provide a line of sight to the nominal landing site until it has reached a distance of 200m from the landing site. The problem parameters are given in Table 1, and we assume that the inertia matrix is computed as an affine function of mass according to

$$J(m) = \begin{bmatrix} \alpha_x m + \beta_x & 0 & 0 \\ 0 & \alpha_y m + \beta_y & 0 \\ 0 & 0 & \alpha_z m + \beta_z \end{bmatrix}, \tag{45}$$

where $\boldsymbol{\alpha} = \begin{bmatrix}\alpha_x \; \alpha_y \; \alpha_z\end{bmatrix}^T$, $\boldsymbol{\beta} = \begin{bmatrix}\beta_x \; \beta_y \; \beta_z\end{bmatrix}^T \in \mathbb{R}^3_{++}$ are vehicle specific parameters. For an appropriate choice of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, this model amounts to a linear interpolation between the wet and dry inertia matrices. It is important to note that the use of this model in conjunction with the solution strategy discussed in §IV means that all thrust commands are computed with an understanding of how mass depletion will alter the inertia of the vehicle and affect rotation induced by gimbaling the engine.

Table 1: Problem parameters for the solution of Problem 1.

Parameter	**Value**	**Units**	**Parameter**	**Value**	**Units**
$\boldsymbol{\alpha}$	[1.85 1.85 1.83]	m^2	$\boldsymbol{\beta}$	[7605 7605 13395]	$\mathrm{kg\,m}^2$
m_{wet}	3250	kg	m_{dry}	2100	kg
$\boldsymbol{r}_{\mathcal{I}}(0)$	[250 150 433]	m	$\boldsymbol{v}_{\mathcal{I}}(0)$	$[-30\; 0\; -15]$	m/s
$\boldsymbol{r}_{\mathcal{I}}(t_f)$	[0 0 30]	m	$\boldsymbol{v}_{\mathcal{I}}(t_f)$	$[0\; 0\; -1]$	m/s
$\boldsymbol{r}_u$	$-0.25 \cdot \boldsymbol{z}_{\mathcal{B}}$	m	$\boldsymbol{\omega}_{\mathcal{B}}(0)$, $\boldsymbol{\omega}_{\mathcal{B}}(t_f)$	[0 0 0]	°/s
I_{sp}	225.0	s	$\boldsymbol{q}(t_f)$	[0 0 0 1]	-
$\theta_{\max}$	80.0	°	N	20	-
$\omega_{\max}$	28.6	°/s	$\xi_{\max}$	30	°
$\gamma_{\max}$	75.0	°	$\boldsymbol{p}_{\mathcal{B}}$	$[0.906\; 0\; -0.423]$	-
$\delta_{\max}$	20.0	°	d	200	m
$T_{\min}$	6000	N	$T_{\max}$	22,500	N

All problems are solved using SDPT3[43] and CVX.[45] We solve both the baseline problem (Problem 1 with constraint (44) removed) and Problem 1 for comparison. The solution is initialized with the *straight-line interpolation* method detailed in,[28] in which each state is linearly interpolated between its boundary values, and the thrust is chosen to oppose the force of gravity.

Converged trajectories are shown in Figure 4, while corresponding state and control trajectories are given in Figure 5a and Figure 5b, respectively. In all figures, the black dots represent the discrete solution values (both states and controls) from the optimization process. The solid curves are the trajectories obtained by integrating these control signals through the nonlinear dynamics in (19), (20) and (23).

In Figure 4, the light blue lines represent the boresight vector $\boldsymbol{p}_{\mathcal{B}}$ at the discrete temporal points used for the optimization procedure. In particular, the upper-right South-Zenith projection shows how both the attitude and position of the vehicle are changed from the baseline scenario to account for the pointing constraint enforced at this distance from the landing site. Figure 5a illustrates the distance from the landing site versus the line of sight angle for each trajectory. This figure demonstrates that the line of sight angle is maintained below its desired bound while the vehicle is far enough from the landing site (i.e., when the trigger condition is active). Since the baseline problem does not enforce the line of sight constraint, the corresponding angles are seen to violate the desired limit during the initial portion of the descent. Figure 5b indicates, at times, large deviations in the thrust commands between the two problem instances as the solution to Problem 1 must accommodate for the additional line of sight constraint.

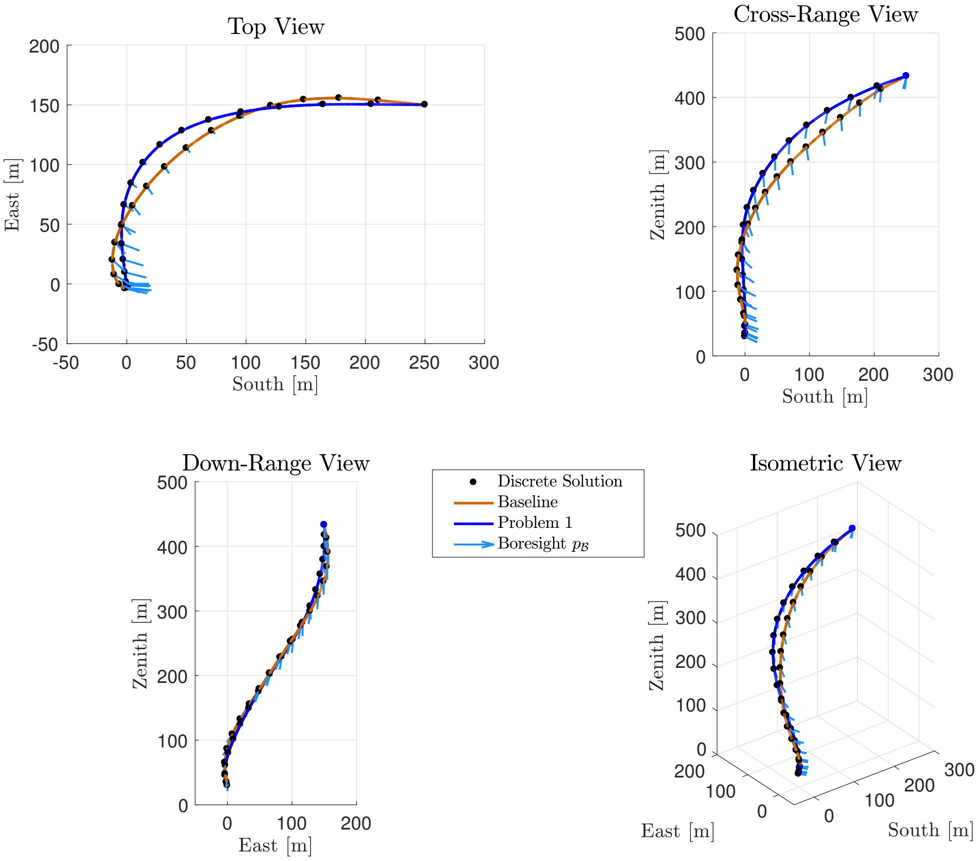


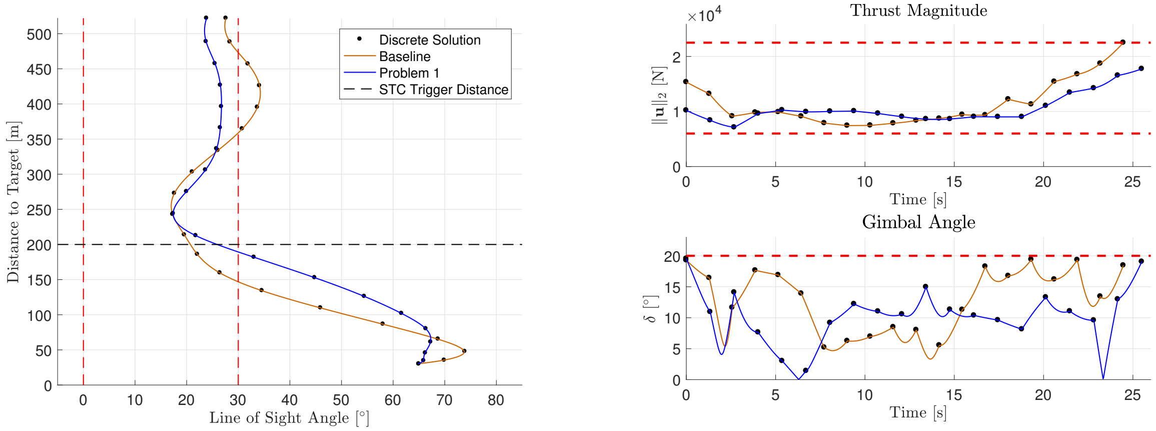
Figure 4: Converged trajectories for both the baseline problem (brown curve) and Problem 1 (blue curve). For Problem 1, a line of sight constraint is enforced conditionally on the distance from the landing site. The light blue line represents the boresight vector $\boldsymbol{p}_{\mathcal{B}}$ in both cases.

VI. Conclusions

In this paper, the 6-degree-of-freedom powered descent guidance problem with state-triggered constraints was formulated. Using dual quaternions to represent the equations of motion, the kinematics, dynamics, and state and control constraints for a variable mass and variable inertia rigid body with a single engine configuration were presented. This work introduced the inequality form of state-triggered constraints with an application to distance-triggered line of sight constraints. The formulation enables the use of continuous optimization tools to generate and study feasible trajectories that are subject to pointing constraints only during certain portions of the descent. The solution method leverages recent work in developing iterative methods to solve nonlinear and nonconvex problems in a way that is amenable to real-time and on-board computation. A numerical example demonstrates these methods for a representative lunar landing scenario, and highlights the utility and impact of state-triggered constraints on the resulting state and control trajectories.

Acknowledgements

This research has been supported by NASA grant NNX17AH02A and was partially carried out at the Johnson Space Center. Government sponsorship acknowledged.



(a) Altitude versus line of sight angle.

(b) Thrust magnitude and gimbal angle commands.

Figure 5: State-triggered constraint phase plot and thrust commands for the converged trajectories. Constraint boundaries are represented by dashed red lines.

References

[1] J. S. Meditch, "On the Problem of Optimal Thrust Programming For a Lunar Soft Landing," IEEE Transactions on Automatic Control, vol. 9, no. 4, pp. 477–484, 1964.

[2] D. Lawden, "Optimal Trajectories for Space Navigation," in Optimal Trajectories for Space Navigation, London: Butterworths, 1963.

[3] T. N. Edelbaum, "Optimal Space Trajectories," tech. rep., Analytical Mechanics Associates, Inc., Jericho, NY, 1969.

[4] J.-P. Marec, Optimal Space Trajectories. Amsterdam: Elsevier Scientific Publishing Company, 1979.

[5] A. R. Klumpp, Apollo guidance, navigation, and control: Apollo lunar-descent guidance. Charles Stark Draper Laboratory, 1971.

[6] C. N. D'Souza, "An Optimal Guidance Law for Planetary Landing," in AIAA Guidance, Navigation, and Control Conference, (New Orleans, LA), pp. 1376—-1381, 1997.

[7] U. Topcu, J. Casoliva, and K. D. Mease, "Fuel Efficient Powered Descent Guidance for Mars Landing," in Guidance, Navigation, and Control Conference and Exhibit, (San Francisco, CA), 2005.

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

[9] F. Najson and K. Mease, "A Computationally Non-Expensive Guidance Algorithm for Fuel Efficient Soft Landing," in AIAA Guidance, Navigation, and Control Conference and Exhibit, Guidance, Navigation, and Control and Co-located Conferences, (San Francisco, CA), aug 2005.

[10] J. R. Rea, An Investigation of Fuel Optimal Terminal Descent. Ph.D. Dissertation, University of Texas at Austin, 2009.

[11] J. Rea and R. Bishop, "Analytical Dimensional Reduction of a Fuel Optimal Powered Descent Subproblem," in AIAA Guidance, Navigation, and Control Conference, (Toronto, ON), 2010.

[12] A. B. Acikmese and S. Ploen, "A Powered Descent Guidance Algorithm for Mars Pinpoint Landing," in AIAA Guidance, Navigation, and Control Conference and Exhibit, (San Francisco, CA), 2005.

[13] S. Ploen, B. Acikmese, and A. Wolf, "A Comparison of Powered Descent Guidance Laws for Mars Pinpoint Landing," in AIAA/AAS Astrodynamics Specialist Conference and Exhibit, (Keystone, CO), 2006.

[14] B. Acikmese and S. R. Ploen, "Convex Programming Approach to Powered Descent Guidance for Mars Landing," Journal of Guidance, Control, and Dynamics, vol. 30, no. 5, pp. 1353–1366, 2007.

[15] J. M. Carson, B. Açıkmeşe, L. Blackmore, and A. Wolf, "Capabilities of Convex Powered-Descent Guidance Algorithms for Pinpoint and Precision Landing," in 2011 IEEE Aerospace Conference, (Big Sky, MT), pp. 1–8, 2011.

[16] J. M. Carson, B. Acikmese, and L. Blackmore, "Lossless Convexification of Powered-Descent Guidance with Non-Convex Thrust Bound and Pointing Constraints," in Proceedings of the 2011 American Control Conference, (San Francisco, CA), pp. 2651–2656, 2011.

[17] B. Acikmese, J. M. Carson, and L. Blackmore, "Lossless Convexification of Nonconvex Control Bound and Pointing Constraints of the Soft Landing Optimal Control Problem," IEEE Transactions on Control Systems Technology, vol. 21, no. 6, pp. 2104–2113, 2013.

[18] L. Blackmore, B. Acikmese, and D. P. Scharf, "Minimum-Landing-Error Powered-Descent Guidance for Mars Landing Using Convex Optimization," Journal of Guidance, Control, and Dynamics, vol. 33, no. 4, pp. 1161–1171, 2010.

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

[20] L. Blackmore, B. Açıkmeşe, and J. M. Carson, “Lossless convexification of control constraints for a class of nonlinear optimal control problems,” Systems and Control Letters, vol. 61, no. 8, pp. 863–870, 2012.

[21] M. W. Harris and B. Acikmeşe, “Lossless Convexification of Non-Convex Optimal Control Problems for State Constrained Linear Systems,” Automatica, vol. 50, no. 9, pp. 2304–2311, 2014.

[22] U. Lee and M. Mesbahi, “Dual Quaternions, Rigid Body Mechanics, and Powered-Descent Guidance,” in Proceedings of the IEEE Conference on Decision and Control, (Maui, HI), pp. 3386–3391, 2012.

[23] U. Lee and M. Mesbahi, “Optimal Powered Descent Guidance with 6-DoF Line of Sight Constraints via Unit Dual Quaternions,” in AIAA Guidance, Navigation, and Control Conference, (Kissimmee, FL), 2015.

[24] U. Lee and M. Mesbahi, “Constrained Autonomous Precision Landing via Dual Quaternions and Model Predictive Control,” Journal of Guidance, Control, and Dynamics, vol. 40, no. 2, pp. 292–308, 2017.

[25] M. Szmuk, B. Acikmese, and A. W. Berning, “Successive Convexification for Fuel-Optimal Powered Landing with Aerodynamic Drag and Non-Convex Constraints,” in AIAA Guidance, Navigation, and Control Conference, (San Diego, CA), 2016.

[26] M. Szmuk, U. Eren, and B. Acikmese, “Successive Convexification for Mars 6-DoF Powered Descent Landing Guidance,” in AIAA Guidance, Navigation, and Control Conference, (Grapevine, TX), 2017.

[27] M. Szmuk and B. Acikmese, “Successive Convexification for 6-DoF Mars Rocket Powered Landing with Free-Final-Time,” in AIAA Guidance, Navigation, and Control Conference, (Orlando, FL), 2018.

[28] M. Szmuk, T. P. Reynolds, and B. Açıkmeşe, “Successive convexification for real-time 6-dof powered descent guidance with state-triggered constraints,” arXiv e-prints, November 2018. arXiv:1811.10803.

[29] Y. Mao, M. Szmuk, and B. Acikmese, “Successive Convexification of Non-Convex Optimal Control Problems and its Convergence Properties,” in 2016 IEEE 55th Conference on Decision and Control, (Las Vegas, NV), pp. 3636–3641, 2016.

[30] Y. Mao, D. Dueri, M. Szmuk, and B. Açıkmeşe, “Successive Convexification of Non-Convex Optimal Control Problems with State Constraints,” IFAC-PapersOnLine, vol. 50, no. 1, pp. 4063–4069, 2017.

[31] Y. Mao, M. Szmuk, and B. Acikmese, “Successive Convexification: A Superlinearly Convergent Algorithm for Non-convex Optimal Control Problems,” arXiv e-prints, 2018. arXiv:1804.06539.

[32] N. Filipe and P. Tsiotras, “Adaptive Position and Attitude Tracking Controller for Satellite Proximity Operations using Dual Quaternions,” Advances in the Astronautical Sciences, vol. 150, no. 4, pp. 2313–2332, 2014.

[33] A. Valverde and P. Tsiotras, “Modeling of Spacecraft-Mounted Robot Dynamics and Control Using Dual Quaternions,” in American Control Conference, pp. 670–675, 2018.

[34] W. Clifford, Mathematical Paper. Macmillan and Company, 1882.

[35] U. Lee and M. Mesbahi, “Dual Quaternion Based Spacecraft Rendezvous With Rotational and Translational Field of View,” in AIAA/AAS Astrodynamics Specialist Conference, (San Diego, CA), pp. 1–20, 2014.

[36] J. M. McCarthy, Introduction to Theoretical Kinematics. Cambridge, MA: The MIT Press, 1990.

[37] W. T. Thomson, Introduction to Space Dynamics. Toronto, ON: Dover Publications, Inc., 1986.

[38] M. Szmuk, T. P. Reynolds, D. Malyuta, B. Acikmese, M. Mesbahi, and J. M. Carson, “A Tutorial on Successive Convexification for Real-Time Rocket Landing Guidance with State-Triggered Constraints,” in AIAA SciTech Forum, (San Diego, CA), 2019.

[39] R. W. Cottle, J.-S. Pang, and R. E. Stone, Linear Complementarity Problem. San Diego, CA: Academic Press Limited, 1992.

[40] A. Domahidi, E. Chu, and S. Boyd, “ECOS: An SOCP solver for embedded systems,” in European Control Conference, pp. 3071–3076, 2013.

[41] D. Dueri, B. Açıkmeşe, D. P. Scharf, and M. W. Harris, “Customized Real-Time Interior-Point Methods for Onboard Powered-Descent Guidance,” Journal of Guidance, Control, and Dynamics, vol. 40, no. 2, pp. 197–212, 2016.

[42] D. Dueri, J. Zhang, and B. Açikmese, “Automated Custom Code Generation for Embedded, Real-time Second Order Cone Programming,” IFAC Proceedings Volumes, vol. 47, no. 3, pp. 1605–1612, 2014.

[43] K. C. Toh, M. J. Todd, and R. H. Tütüncü, “SDPT3: A Matlab software package for semidefinite programming, Version 1.3,” Optimization Methods and Software, vol. 11, no. 1-4, pp. 545–581, 1999.

[44] D. Malyuta, T. P. Reynolds, M. Szmuk, M. Mesbahi, B. Acikmese, and J. M. Carson, “Discretization Performance and Accuracy Analysis for the Powered Descent Guidance Problem,” in AIAA SciTech Forum, (San Diego, CA), 2019.

[45] M. Grant and S. Boyd, “CVX: Matlab Software for Disciplined Convex Programming, version 2.1,” 2014.