AIAA 2015-0319

AIAA SciTech Forum
5-9 January 2015, Kissimmee, Florida
AIAA Guidance, Navigation, and Control Conference

Optimal Powered Descent Guidance with 6-DoF Line of Sight Constraints via Unit Dual Quaternions

Unsik Lee[*] and Mehran Mesbahi[†]
University of Washington, Seattle, WA 98195-2400

In this paper, we present a model predictive control (MPC) approach for powered descent guidance and control synthesis in the presence of line of sight and glide slope constraints. The design of an autonomous control algorithm for such a mission scenario is challenging due to fact that the constraints are coupled with rotational and translational motions of a lander spacecraft, leading to a complex configuration space. We approach this problem by representing the general dynamics of the rigid body in an uniform gravity field as a piece-wise affine system utilizing the unit dual quaternion parameterization. Such a parameterization enables a 6-DOF motion prediction algorithm while also allowing selecting a quadratic cost on the required acceleration commands in order to minimize propellant consumption. The constraints are imposed to ensure that the lander's position and orientation are not violating the constraints. A novel feature of this approach is the development of convex representable subset of unit dual quaternions that correspond to translational and rotational states satisfying predefined types of constraints with respect to a moving body frame. The stability and feasibility issues of the piece-wise affine MPC approach are also discussed. Simulation results are then presented to demonstrate the effectiveness of the proposed algorithm for powered descent guidance.

I. Introduction

A lander spacecraft will be required to perform pinpoint landings in next generation Mars missions. This will be accomplished by guiding the lander spacecraft to a given target on the surface of the planet with an accuracy of several hundred meters while meeting constraints on the rotational and translational motions during descent [1]. For example, it may be required to land on a scientifically interesting target surrounded by hazardous terrain or to land close to other prepositioned surface assets such as rovers, cargo, and fuel. The accuracy of the lander spacecraft's trajectory is influenced by the delivery error and knowledge uncertainty at the entry interface, environmental uncertainty, and the vehicle performance [2]. Further improvements in landing accuracy will require enhancements to heritage flight systems against environmental uncertainty which can degrade estimation accuracy and tracking performance. In this context, onboard vision sensors must maintain the line of sight with the surface target during descent. This type of constraint is also crucial for human missions since the crew is often responsible in making decisions in the last stage of landing in case of sensor failures or the discovery of unexpected terrain hazards. The line of sight vector pointing from the onboard sensors to the surface target relates to the coupled lander spacecraft attitude and its relative position with respect to the target. The topological complexity of the resulting lander configuration space leads to a challenging control synthesis problem for autonomous guidance.

In this paper, unlike the conventional means of representing the orientation and position of the lander spacecraft, we utilize the unit dual quaternion parameterization to simultaneously represent the orientation and position of the lander. This parameterization enables us to approach rotationally and translationally constrained zones in a unified fashion; more importantly, the resulting non-convex constraints can then be converted into computationally tractable convex constraints suitable for onboard computation.

Finding an energy optimal trajectory for the lander to the surface target in a uniform gravity field is referred to as the soft landing problem in the literature [3]. In general, there is no known closed form solution for soft landing with state and control constraints, and as a result, various numerical approaches have been proposed for this problem over the past several decades. For example, in [7] and [8], first-order necessary conditions are derived for the minimal fuel landing problem while satisfying throttle and thrust angle control constraints. With advances in aerospace technology, real-time onboard computation has gained attention in many aerospace fields, as the resulting autonomous systems are often more robust to disturbances and unpredictable operational environments. In this context, [9], [10] and [11] exploit the structure of the problem to design guidance algorithms with guaranteed convergence to the global optimum with a deterministic convergence criterion using a convex optimization framework.

[*] Postdoctoral Researcher, W. E. Boeing Department of Aeronautics & Astronautics, AIAA Member, unsik@uw.edu

[†] Professor, W. E. Boeing Department of Aeronautics & Astronautics, mesbahi@uw.edu

In this paper, we present a model predictive control algorithm for the constrained powered descent guidance within the framework of piece-wise affine (PWA) systems that can effectively approximate the nonlinear rotational and translational motion dynamics of a rigid body in a uniform gravity field. In the PWA-MPC formulation, constraints are imposed to ensure that the lander spacecraft does not violate its rotational and translational constraints during its descent. Specifically, we present an approach based on convex representable constraints in unit dual quaternions corresponding to translational and rotational states satisfying two types of constraints with respect to a moving body frame. The stability and feasibility issues of the corresponding piece-wise affine MPC are then discussed. Lastly, simulation results are presented to demonstrate the effectiveness of the proposed approach for powered descent guidance.

II. Preliminaries

In this section, we provide a brief background on unit quaternions and unit dual quaternions. More detailed discussions on these representations can be found in [4, 5, 6].

A. Unit Quaternions

The attitude of a rigid body, describing the relative orientation between a reference frame and a body-fixed frame, evolves on the special orthogonal group $\mathsf{SO}(3)$. The unit quaternion is a minimal parameterization of $\mathsf{SO}(3)$ free from singularities. A unit quaternion, as the three dimensional extension of a unit complex number, parameterizes a rotating axis with three perpendicular imaginary axes i, j, and k. We adopt the vector notation for unit quaternions as

$$\mathbf{q} = \begin{bmatrix} \mathbf{n}_x \sin\frac{\theta}{2} \\ \mathbf{n}_y \sin\frac{\theta}{2} \\ \mathbf{n}_z \sin\frac{\theta}{2} \\ \cos\frac{\theta}{2} \end{bmatrix}, \tag{1}$$

where $\mathbf{n}_x$, $\mathbf{n}_x$ and $\mathbf{n}_x$ denote three perpendicular components of the rotation axis $\mathbf{n}$, and θ denotes the rotation angle around this axis. Unit quaternions are closed for the following operations:

Quaternion Multiplication: Using the matrix notation, we can compute the product of two quaternions as

$$\mathbf{q} \otimes \mathbf{p} = \begin{bmatrix} q_0\boldsymbol{p} + p_0\boldsymbol{q} + \boldsymbol{q} \times \boldsymbol{p} \\ q_0 p_0 - \boldsymbol{q}^T\boldsymbol{p} \end{bmatrix}, \tag{2}$$

where $\mathbf{q} = [\boldsymbol{q}^T,\ q_0]^T$ and $\mathbf{p} = [\boldsymbol{p}^T,\ p_0]^T$. Another quaternion operation is the "unit quaternion conjugation" defined as $\mathbf{q}^* = [\,-\boldsymbol{q}^T\ q_0\,]^T$ in the vector notation, which facilitates the judicious definition of the "attitude difference/error" of $\mathbf{p}$ with respect to $\mathbf{q}$ as

$$\mathbf{q}_e \overset{\text{def}}{=} \mathbf{q}^* \otimes \mathbf{p} = \begin{bmatrix} q_0\boldsymbol{p} - p_0\boldsymbol{q} - \boldsymbol{q} \times \boldsymbol{p} \\ q_0 p_0 + \boldsymbol{q}^T\boldsymbol{p} \end{bmatrix}. \tag{3}$$

Note that the identity quaternion is expressed by $\mathbf{q_I} = [\,0\ 0\ 0\ 1\,]^T$. Quaternion multiplication is analogous in many ways to vector cross products. Thus, the quaternion multiplication can be expressed as the product of a skew symmetric matrix and a quaternion:

$$\begin{aligned} \mathbf{q} \otimes \mathbf{p} &= \begin{bmatrix} [\boldsymbol{q}]_\wedge + q_0\mathbf{I_3} & \boldsymbol{q} \\ -\boldsymbol{q}^T & q_0 \end{bmatrix} \begin{bmatrix} \boldsymbol{p} \\ p_0 \end{bmatrix} \overset{\text{def}}{=} [\mathbf{q}]_\otimes \mathbf{p} && (4) \\ &= \begin{bmatrix} [\boldsymbol{p}]_\wedge^T + p_0\mathbf{I_3} & \boldsymbol{p} \\ -\boldsymbol{p}^T & p_0 \end{bmatrix} \begin{bmatrix} \boldsymbol{q} \\ q_0 \end{bmatrix} \overset{\text{def}}{=} [\mathbf{p}]_\otimes^* \mathbf{q}, && (5) \end{aligned}$$

where $[\mathbf{q}]_\otimes$ and $[\mathbf{p}]_\otimes^*$ denote 4×4 skew symmetric matrices related to "$\mathbf{q}$" and "$\mathbf{p}$", respectively; $[\boldsymbol{a}]_\wedge$ denotes the cross product operator associated with the vector $\boldsymbol{a}$ defined as

$$[\boldsymbol{a}]_\wedge = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}. \tag{6}$$

Quaternion Cross Products:

The quaternion multiplication Eq. (2) can be relaxed to the newly defined *quaternion cross product* as

$$\mathbf{q} \times \mathbf{p} = \begin{bmatrix} q_0\boldsymbol{p} + p_0\boldsymbol{q} + \boldsymbol{q} \times \boldsymbol{p} \\ 0 \end{bmatrix} = \begin{bmatrix} [\boldsymbol{q}]_\wedge + q_0\mathbf{I}_3 & \boldsymbol{q} \\ 0_{1\times 3} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{p} \\ p_0 \end{bmatrix} \overset{\text{def}}{=} [\mathbf{q}]_\times \mathbf{p}, \tag{7}$$

$$= \begin{bmatrix} [\boldsymbol{p}]_\wedge^T + p_0\mathbf{I}_3 & \boldsymbol{p} \\ 0_{1\times 3} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{q} \\ q_0 \end{bmatrix} \overset{\text{def}}{=} [\mathbf{p}]_\times^* \mathbf{q}, \tag{8}$$

where $\mathbf{q}, \mathbf{p}$ denote quaternions and $[\cdot]_\wedge$ denotes a 4×4 linear operator matrix corresponding to the *quaternion cross product*. When quaternions are represented as elements in $\mathbb{R}^4$, their algebraic properties are extended with vector-based products such as the inner product. The following are examples of the extended algebraic properties of quaternions:

$$\mathbf{a} \otimes (\mathbf{b} + \mathbf{c}) = \mathbf{a} \otimes \mathbf{b} + \mathbf{a} \otimes \mathbf{c} \tag{9}$$

$$(\mathbf{a} \otimes \mathbf{b})^* = \mathbf{b}^* \otimes \mathbf{a}^* \tag{10}$$

$$(\gamma\mathbf{a}) \otimes \mathbf{b} = \mathbf{a} \otimes (\gamma\mathbf{b}) = \gamma(\mathbf{a} \otimes \mathbf{b}) \tag{11}$$

$$\mathbf{a} \otimes (\mathbf{b} \otimes \mathbf{c}) = (\mathbf{a} \otimes \mathbf{b}) \otimes \mathbf{c} \tag{12}$$

$$\mathbf{a}^T(\mathbf{b} \otimes \mathbf{c}) = c^T(\mathbf{b}^* \otimes \mathbf{a}) = \mathbf{b}^T(\mathbf{a} \otimes \mathbf{c}^*), \tag{13}$$

where $\gamma \in \mathbb{R}$. Let us present another interesting algebraic properties applicable to unit quaternions. This identity is inspired by the Binet-Cauchy identity [12].

Theorem 1 (Unit Quaternion Triple Identity)**.** *Assume that* $\mathbf{t}$ *and* $\mathbf{v}$ *are quaternions, and* $\mathbf{q}$ *is a unit quaternion. Then, for the vector inner product between two quaternion products, the following identity holds:*

$$(\mathbf{t} \otimes \mathbf{q})^T(\mathbf{y} \otimes \mathbf{q}) = (\mathbf{q} \otimes \mathbf{t})^T(\mathbf{q} \otimes \mathbf{y}) = \mathbf{t}^T\mathbf{y}, \tag{14}$$

where $\mathbf{t}, \mathbf{y} \in \mathbb{R}^4$ *and* $\mathbf{q} \in \mathcal{S}^3$.

Proof. The following proof utilizes the properties of quaternions. Using Eq. (13), we have

$$(\mathbf{t} \otimes \mathbf{q})^T(\mathbf{y} \otimes \mathbf{q}) = \mathbf{y}^T\left((\mathbf{t} \otimes \mathbf{q}) \otimes \mathbf{q}^*\right) \tag{15}$$

$$= \mathbf{y}^T\left(\mathbf{t} \otimes (\mathbf{q} \otimes \mathbf{q}^*)\right) \tag{16}$$

$$= \mathbf{y}^T\mathbf{t}, \tag{17}$$

and

$$(\mathbf{q} \otimes \mathbf{t})^T(\mathbf{q} \otimes \mathbf{y}) = \mathbf{y}^T\left(\mathbf{q}^* \otimes (\mathbf{q} \otimes \mathbf{t})\right) \tag{18}$$

$$= \mathbf{y}^T\left(\mathbf{q}^* \otimes \mathbf{q}) \otimes \mathbf{t}\right) \tag{19}$$

$$= \mathbf{y}^T\mathbf{t}, \tag{20}$$

concluding the proof. We note that the above identity remains valid when $\mathbf{t}, \mathbf{y}$ are pure quaternions, namely, when $\mathbf{t} = [\,\boldsymbol{t}^T\ 0\,], \mathbf{v} = [\,\boldsymbol{v}^T\ 0\,]$ with $\boldsymbol{t}, \boldsymbol{v} \in \mathbb{R}^3$. □

B. Unit Dual Quaternions

The concept of the "dual quaternion" was invented by W. Clifford to accommodate the additional information, namely, relative position, encoded in a "regular" quaternion. The dual quaternion $\tilde{\mathbf{q}}$ is comprised of two quaternions: $\mathbf{q}_1$ and $\mathbf{q}_2$, denoting the real and dual parts of $\tilde{\mathbf{q}}$, respectively, combined as

$$\tilde{\mathbf{q}} = \mathbf{q}_1 + \epsilon\,\mathbf{q}_2, \tag{21}$$

where ϵ denotes the "dual unit" satisfying the "unconventional" properties

$$\epsilon^2 = 0 \quad \text{and} \quad \epsilon \neq 0. \tag{22}$$

We shall adopt a convenient notation for unit dual quaternions. Since all operations on unit dual quaternions considered in this paper are linear, unit dual quaternions can be embedded in $\mathbb{R}^8$ Euclidean space as

$$\tilde{\mathbf{q}} = \mathbf{q}_1 + \epsilon\, \mathbf{q}_2, \quad \text{or} \quad \tilde{\mathbf{q}} = \begin{bmatrix} \mathbf{q}_1 \\ \mathbf{q}_2 \end{bmatrix}_{8\times 1} \tag{23}$$

where $\mathbf{q}_1$ and $\mathbf{q}_2$ are quaternions. For the set of *unit dual quaternions*, which is a subset of dual quaternions, the real part $\mathbf{q}_1$ is restricted to be a unit norm vector $\|\mathbf{q}_1\|_2 = 1$.

The unit dual quaternion is closed under multiplication, which follows from the property defined in Eq. (22). The corresponding operation is expressed as,

$$\tilde{\mathbf{q}} \tilde{\otimes} \tilde{\mathbf{p}} = \mathbf{q}_1 \otimes \mathbf{p}_1 + \epsilon \left(\mathbf{q}_1 \otimes \mathbf{p}_2 + \mathbf{q}_2 \otimes \mathbf{p}_1 \right) = \begin{bmatrix} \mathbf{q}_1 \otimes \mathbf{p}_1 \\ \mathbf{q}_1 \otimes \mathbf{p}_2 + \mathbf{q}_2 \otimes \mathbf{p}_1 \end{bmatrix}. \tag{24}$$

Note that dual quaternion multiplication, denoted by $\tilde{\otimes}$, is a linear operation; we can represent it as a multiplication between a matrix and a vector (dual quaternion) in conformance with quaternion multiplication as described in Eq. (4). Specifically,

$$\begin{aligned} \tilde{\mathbf{q}} \tilde{\otimes} \tilde{\mathbf{p}} &= \begin{bmatrix} [\mathbf{q}_1]_\otimes & 0_{4\times 4} \\ [\mathbf{q}_2]_\otimes & [\mathbf{q}_1]_\otimes \end{bmatrix} \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \end{bmatrix} \overset{\text{def}}{=} [\tilde{\mathbf{q}}]_{\tilde{\otimes}}\, \tilde{\mathbf{p}}, \\ &= \begin{bmatrix} [\mathbf{p}_1]^*_\otimes & 0_{4\times 4} \\ [\mathbf{p}_2]^2_\otimes & [\mathbf{p}_1]^*_\otimes \end{bmatrix} \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \end{bmatrix} \overset{\text{def}}{=} [\tilde{\mathbf{p}}]^*_{\tilde{\otimes}}\, \tilde{\mathbf{q}}. \end{aligned} \tag{25}$$

As an extension of the *quaternion cross product*, we also define

$$\begin{aligned} \tilde{\mathbf{q}} \tilde{\times} \tilde{\mathbf{p}} &= \begin{bmatrix} \mathbf{q}_1 \times \mathbf{p}_1 \\ \mathbf{q}_1 \times \mathbf{p}_2 + \mathbf{q}_2 \times \mathbf{p}_1 \end{bmatrix} = \begin{bmatrix} [\mathbf{q}_1]_\times & 0_{4\times 4} \\ [\mathbf{q}_2]_\times & [\mathbf{q}_1]_\times \end{bmatrix} \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \end{bmatrix} \overset{\text{def}}{=} [\tilde{\mathbf{q}}]_{\tilde{\times}}\, \tilde{\mathbf{p}} \\ &= \begin{bmatrix} [\mathbf{p}_1]^*_\times & 0_{4\times 4} \\ [\mathbf{p}_2]^*_\times & [\mathbf{p}_1]^*_\times \end{bmatrix} \begin{bmatrix} \mathbf{q}_1 \\ \mathbf{q}_2 \end{bmatrix} \overset{\text{def}}{=} [\tilde{\mathbf{p}}]^*_{\tilde{\times}}\, \tilde{\mathbf{q}} \end{aligned} \tag{26}$$

When the dual quaternion has a nonzero real part, its inverse can be obtained as $\tilde{\mathbf{q}}^{-1} = \|\tilde{\mathbf{q}}\|^{-1} \tilde{\mathbf{q}}^*$; on the other hand, its conjugate is defined as

$$\tilde{\mathbf{q}}^* = \begin{bmatrix} \mathbf{q}_1^* \\ \mathbf{q}_2^* \end{bmatrix}. \tag{27}$$

The norm of a dual quaternion is defined as $\|\tilde{\mathbf{q}}\|_{dq} = \tilde{\mathbf{q}}^* \otimes \tilde{\mathbf{q}} = \tilde{\mathbf{q}} \otimes \tilde{\mathbf{q}}^*$. In the case of unit dual quaternions, the norm of the dual quaternion yields

$$\|\tilde{\mathbf{q}}\|_{dq} = \begin{bmatrix} \mathbf{q}_\mathbf{I} \\ 0_{4\times 1} \end{bmatrix}, \tag{28}$$

where $\mathbf{q}_\mathbf{I}$ denotes an identity unit quaternion. Note that the norm of a dual quaternion in this sense is a dual quaternion, as distinct from the case of quaternions (the norm of a quaternion is a scalar).

C. Unit Dual Quaternions and $\mathsf{SE}(3)$

Suppose that the relationship between the inertially fixed frame $\mathcal{O}$ and the body frame $\mathcal{B}$ is as shown in Fig. 1. At each instance, the configuration space for the position and orientation of the rigid body is described by a 4×4 homogeneous transformation matrix. In particular, $\mathsf{SE}(3)$ is the set of all rigid body transformation in the three dimensional space:

$$\mathsf{SE}(3) = \left\{ T \in \mathbb{R}^{4\times 4} \,|\, T = \begin{bmatrix} R & t_o \\ 0 & 1 \end{bmatrix},\, R \in \mathsf{SO}(3),\, t_o \in \mathbb{R}^3 \right\}, \tag{29}$$

where t_o denotes a position vector to the body frame $\mathcal{B}$ with inertial frame components. As unit quaternions parameterize $\mathsf{SO}(3)$, a unit dual quaternion can be used to define a rigid body rotation ($\tilde{\mathbf{q}} \in \mathbf{S}^3 \times \mathbb{R}^3 \mapsto \mathsf{SE}(3)$). The

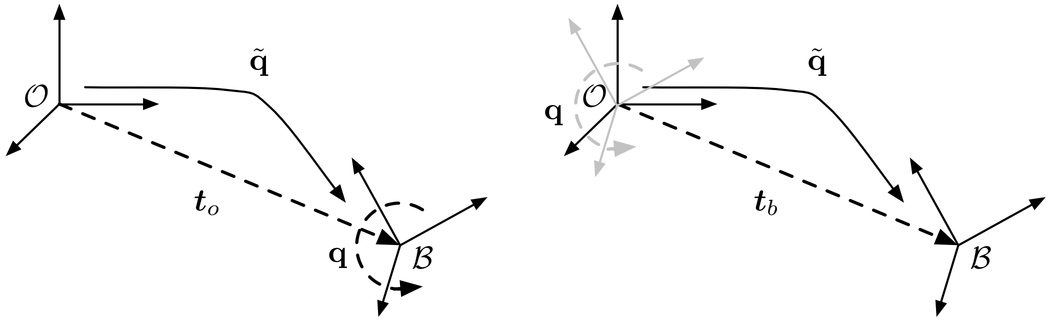

Figure 1. The geometric difference between the frame $\mathcal{B}$ and $\mathcal{O}$ can be expressed in two ways: a translation t_o followed by a rotation q [left figure] and a rotation q followed by a translation t_b [right figure], which are represented by the unit dual quaternion $\tilde{\mathrm{q}}$

geometric difference between the frame $\mathcal{B}$ with respect to the inertially fixed frame $\mathcal{O}$ can be expressed in two ways. A translation $\boldsymbol{t}_o$ (represented in frame $\mathcal{O}$) followed by a rotation $\mathbf{q}_1$ and a rotation $\mathbf{q}_1$ followed by a translation $\boldsymbol{t}_b$ (represented in frame $\mathcal{B}$), which are represented in the form of *unit dual quaternions* as

$$\tilde{\mathbf{q}} \overset{\text{def}}{=} \begin{bmatrix} \mathbf{q} \\ \frac{1}{2}\boldsymbol{t}_o \otimes \mathbf{q} \end{bmatrix} = \begin{bmatrix} \mathbf{q} \\ \frac{1}{2}\mathbf{q} \otimes \boldsymbol{t}_b \end{bmatrix}, \tag{30}$$

where $\boldsymbol{t}_o$ and $\boldsymbol{t}_b$ represent the translation vector $\boldsymbol{t}$ with respect to the frames $\mathcal{O}$ and $\mathcal{B}$, respectively. Note that Eq. (30) satisfies Eq. (28). Converting between $\boldsymbol{t}_o$ and $\boldsymbol{t}_b$ is governed by quaternion rotation operator as follows:

$$\boldsymbol{t}_o = \mathbf{q} \otimes \boldsymbol{t}_b \otimes \mathbf{q}^*, \tag{31}$$

$$\boldsymbol{t}_b = \mathbf{q}^* \otimes \boldsymbol{t}_o \otimes \mathbf{q}. \tag{32}$$

Remark 1. When using the position information extracted from the unit dual quaternion, it should be noted that although $\boldsymbol{t}_b$ and $\boldsymbol{t}_o$ are represented in two different frames, they can also be viewed in an inertially fixed frame; see Fig. (1). As a result, the position vector to the inertially fixed frame $\mathcal{O}$ from the body-fixed frame $\mathcal{B}$ is in fact $-\boldsymbol{t}_b$.

III. Rigid Body Dynamics in Unit Dual Quaternions

The unit dual quaternion kinematic equation [4] is given as

$$\dot{\tilde{\mathbf{q}}} = \frac{1}{2}\tilde{\mathbf{q}}\tilde{\otimes}\tilde{\boldsymbol{\omega}} \quad \text{or} \quad \dot{\tilde{\mathbf{q}}} = \frac{1}{2}[\tilde{\boldsymbol{\omega}}]^*_{\tilde{\otimes}}\tilde{\mathbf{q}} \tag{33}$$

with

$$\tilde{\boldsymbol{\omega}} = \begin{bmatrix} \boldsymbol{\omega}_b \\ \boldsymbol{v}_b \end{bmatrix}_{8\times 1}, \tag{34}$$

where $\boldsymbol{\omega}_b = [\omega_b^T \ \ 0]^T$ denotes the angular velocity of the rigid body and $\boldsymbol{v}_b = [v_b^T \ \ 0]^T$ denotes the translational velocity defined by $v_b = \dot{t}_b + \omega_b \times t_b$. On the other hand, the translational and rotational motions of a fully actuated rigid body are expressed by the rate of change of linear and angular momentum through a set of differential equations [13]:

$$F = \left[\frac{d}{dt}(mv)\right]_{\mathcal{B}} = m\dot{v}_b + \omega_b \times mv_b, \tag{35}$$

$$T = \left[\frac{d}{dt}(J\omega)\right]_{\mathcal{B}} = J\dot{\omega}_b + \omega_b \times J\omega_b, \tag{36}$$

where $[d(\cdot)/dt]_{\mathcal{B}}$ denotes the time derivative represented in the body frame $\mathcal{B}$ and $m \in \mathbb{R}$ denotes the mass of the rigid body, $J \in \mathbb{R}^{3\times 3} = \mathtt{diag}(J_1,\ J_2,\ J_3)$ denotes its inertia matrix along the body frame, and $\omega_b, v_b \in \mathbb{R}^3$ denote, respectively, the angular and translational velocities of the rigid body represented in the body frame. Moreover, F and T denote the external translational force and torque acting on the rigid body, written in the body frame. Note that in order to focus on the feasibility of the dual quaternion-based algorithms, we assume that uncertainties and all external disturbances on the rigid body are negligible.

The following observation is needed shortly.

Lemma 1. *Let $\boldsymbol{J}$ be a real block matrix of the form*

$$\boldsymbol{J} = \begin{bmatrix} A & B \\ C & D \end{bmatrix}, \tag{37}$$

where A, B, C, D are $n \times n$ pair-wise commutative set of matrices. Then, det $(\boldsymbol{J})$ = det$(AD - BC)$.

Proof. See ref. [14] □

Proposition 1. *The rotational and translational motion dynamics Eqs. (36) and (35) can be represented in dual quaternions as*

$$\boldsymbol{J}\dot{\tilde{\boldsymbol{\omega}}} + \tilde{\boldsymbol{\omega}}\tilde{\times}\boldsymbol{J}\tilde{\boldsymbol{\omega}} = \tilde{\mathbf{F}} \tag{38}$$

where

$$\boldsymbol{J} = \left[\begin{array}{cc|cc} 0_{3\times3} & 0 & m\mathbf{I}_3 & 0 \\ 0 & 0 & 0 & 1 \\ \hline J & 0 & 0_{3\times3} & 0 \\ 0 & 1 & 0 & 0 \end{array}\right]_{8\times8} \quad \textit{and} \quad \tilde{\mathbf{F}} = \begin{bmatrix} F \\ 0 \\ T \\ 0 \end{bmatrix}. \tag{39}$$

Moreover, $\boldsymbol{J} \in \mathbb{R}^{8\times8}$ forms an invertible block anti-diagonal matrix.

Proof. Rewriting Eq. (38) with Eqs. (34) and (39) yields

$$\begin{bmatrix} m\dot{\boldsymbol{v}}_b \\ J\dot{\boldsymbol{\omega}}_b \end{bmatrix} + \begin{bmatrix} \boldsymbol{\omega}_b \\ \boldsymbol{v}_b \end{bmatrix} \tilde{\times} \begin{bmatrix} m\boldsymbol{v}_b \\ J\boldsymbol{\omega}_b \end{bmatrix} = \begin{bmatrix} \boldsymbol{F} \\ \boldsymbol{T} \end{bmatrix} \tag{40}$$

where $\boldsymbol{F} = [F^T, 0]^T$ and $\boldsymbol{T} = [T^T, 0]^T$. Noting that

$$\boldsymbol{v}_b \tilde{\times} m\boldsymbol{v}_b = 0,$$

we have

$$\begin{bmatrix} m\dot{v}_b \\ J\dot{\omega}_b \end{bmatrix} + \begin{bmatrix} \omega_b \times mv_b \\ \omega_b \times J\omega_b \end{bmatrix} = \begin{bmatrix} F \\ T \end{bmatrix}, \tag{41}$$

which is identical to Eqs. (35) and (36). Now, assume that $\boldsymbol{J}$ in Eq. (39) has the same form as Eq. (37). Since all 4×4 block matrices are commutative, from *Lemma 1*, we have

$$\det(\boldsymbol{J}) = \det(-BC) \tag{42}$$

$$= \det(B)\det(C), \tag{43}$$

where B, C denote the right upper and left lower block matrices in $\boldsymbol{J}$ whose determinants are not zero. Thus, $\boldsymbol{J}$ is always invertible. □

Eq. (33) and Eq. (38) constitute the kinematics and dynamics of the rigid body in terms of dual quaternions, considering *integrated* translational and rotational motion. Next, consider the gravitational force applied to the system. The gravitational force is defined as a constant force observed in the inertially fixed frame $\mathcal{O}$ as

$$\mathbf{g}_o = \begin{bmatrix} 0 \\ 0 \\ g \\ 0 \end{bmatrix}, \tag{44}$$

where g denotes a gravitational acceleration. Using the quaternion rotation operator, the gravitational acceleration observed in the body frame $\mathcal{B}$ is given in the form of a dual quaternion as

$$\tilde{\mathbf{g}}_b = \begin{bmatrix} m\left(\mathbf{q}^* \otimes \mathbf{g}_o \otimes \mathbf{q}\right) \\ 0_{4\times 1} \end{bmatrix} = m\tilde{\mathbf{I}}_8\left(\tilde{\mathbf{q}}^*\tilde{\otimes}\tilde{\mathbf{I}}_8\tilde{\mathbf{g}}_o\tilde{\otimes}\tilde{\mathbf{q}}\right) = m\tilde{\mathbf{I}}_8[\tilde{\mathbf{q}}^*\tilde{\otimes}\tilde{\mathbf{I}}_8\tilde{\mathbf{g}}_o]_{\tilde{\otimes}}\tilde{\mathbf{q}} \tag{45}$$

with

$$\tilde{\mathbf{g}}_0 = [\,\mathbf{g}_0^T \quad 0_{1\times 4}\,]^T \text{ and } \tilde{\mathbf{I}}_8 = \begin{bmatrix} 0_{4\times 4} & \mathbf{I}_4 \\ \mathbf{I}_4 & 0_{4\times 4} \end{bmatrix}. \tag{46}$$

The rotational and translational dynamics represented as dual quaternions are now given by

$$\boldsymbol{J}\dot{\tilde{\boldsymbol{\omega}}} + \tilde{\boldsymbol{\omega}}\tilde{\times}\boldsymbol{J}\tilde{\boldsymbol{\omega}} = \tilde{\mathbf{F}} + \tilde{\mathbf{g}}_b, \tag{47}$$

with Eq. (32).

IV. Rotationally and Translationally Constrained Zones

One of the challenging algorithmic aspects of the powered descent guidance problem is that translational and rotational dynamics of the spacecraft are generally taken into account independently. Thus it becomes difficult to handle situations where translational and rotational dynamics are coupled through constraints. For example, it is often required that the lander's onboard vision sensors must maintain the line of sight with the surface target during the lander descent phase. This type of constraint is affected by the lander's orientation and position during landing. In this section, we develop rotationally and translationally constrained zones in the rigid-body's configuration space. Such constraints are involved not only in the rigid-body's translational position but also in its orientation. Since this framework was originally inspired by a powered descent guidance algorithm for the Mars lander, let us formulate the unit dual quaternion-based constrained zones in the context of this application; Fig. 2 depicts the overall landing scenario.

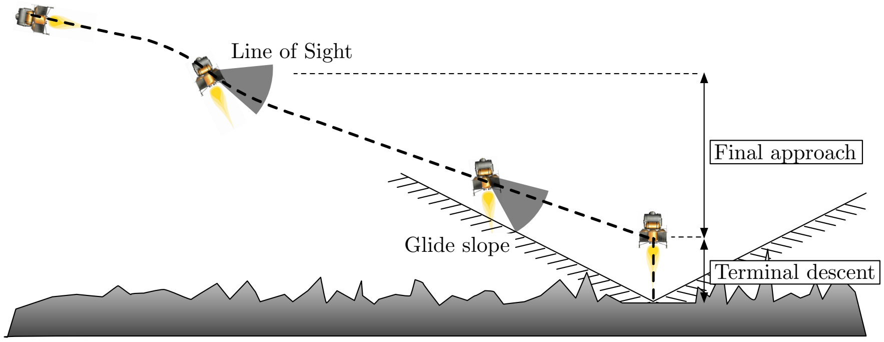

Figure 2. powered descent scenario in the presence of line of sight and glide slope constraints

A. Line of sight constraints

The line of sight constraint can be defined as a cone around a boresight vector in the body frame $\mathcal{B}$ (see Fig. 3). We assume that the frame $\mathcal{O}$ originates from the landing site. Consider now the situation where the body frame $\mathcal{B}$ is initially aligned with the frame $\mathcal{O}$ and subject to translation and rotation by $\tilde{\mathbf{q}}$. Thus, at a specific time, we can determine if the position vector to the target $-\boldsymbol{t}_b$ stays within an angle θ about the line of sight vector $\boldsymbol{y}_b$ in the frame $\mathcal{B}$ with the inner product

$$-\boldsymbol{t}_b \cdot \boldsymbol{y}_b \geq \|\boldsymbol{t}_b\| \cos\theta, \tag{48}$$

where $\boldsymbol{t}_b$ denotes a translation vector represented in the frame $\mathcal{B}$. Note that Eq. (48) holds for $-\pi \le \theta \le \pi$. One can show that the left hand side of the above equation can be expressed by a quadratic function as

$$
\begin{aligned}
\boldsymbol{t}_b \cdot \boldsymbol{y}_b &= (\mathbf{q}_1 \otimes \mathbf{t}_b)^T (\mathbf{q}_1 \otimes \mathbf{y}_b) && (49)\\
&= (\mathbf{q}_1 \otimes \mathbf{t}_b)^T [\mathbf{y}_b]^*_\otimes \, \mathbf{q}_1 && (50)\\
&= \begin{bmatrix} \mathbf{q}_1 \\ \frac{1}{2}\mathbf{q}_1 \otimes \boldsymbol{t}_b \end{bmatrix}^T \begin{bmatrix} 0_{4\times4} & [\mathbf{y}_b]^{*\,T}_\otimes \\ [\mathbf{y}_b]^*_\otimes & 0_{4\times4} \end{bmatrix} \begin{bmatrix} \mathbf{q}_1 \\ \frac{1}{2}\mathbf{q}_1 \otimes \boldsymbol{t}_b \end{bmatrix} \\
&= \tilde{\mathbf{q}}^T M_H \tilde{\mathbf{q}}, && (51)
\end{aligned}
$$

where $\tilde{\mathbf{q}}$ denotes a unit dual quaternion and the quaternion properties Eqs. (5) and (14) have been applied. We note that M_H forms an indefinite symmetric matrix. In addition, motivated by Eq. (14), we can find the following useful identities:

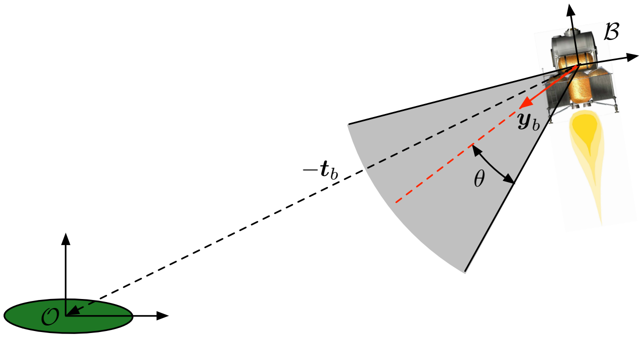

Figure 3. Illustration of the line of sight constraint. It is defined as a cone around the fixed boresight vector $\boldsymbol{y}_b$ in the body frame.

$$
\begin{aligned}
&\tilde{\mathbf{q}}^T \tilde{\mathbf{q}} = \|\tilde{\mathbf{q}}\|^2 = 1 + \frac{1}{4}\|\boldsymbol{t}_b\|^2, && (52)\\
&\tilde{\mathbf{q}}^T E_u \tilde{\mathbf{q}} = 1, \quad 4\,\tilde{\mathbf{q}}^T E_d \tilde{\mathbf{q}} = \|\boldsymbol{t}_b\|^2, \text{ and} && (53)\\
&\|\boldsymbol{t}_b\| = \|2\,E_d \tilde{\mathbf{q}}\| && (54)
\end{aligned}
$$

where

$$
E_u = \begin{bmatrix} \mathbf{I}_4 & 0_{4\times4} \\ 0_{4\times4} & 0_{4\times4} \end{bmatrix}, \quad E_d = \begin{bmatrix} 0_{4\times4} & 0_{4\times4} \\ 0_{4\times4} & \mathbf{I}_4 \end{bmatrix}. \tag{55}
$$

Rewriting Eq. (48) with the above identities in mind, we obtain

$$
f_1(\tilde{\mathbf{q}}) = \tilde{\mathbf{q}}^T M_H \tilde{\mathbf{q}} + \|2\,E_d \tilde{\mathbf{q}}\| \cos\theta \le 0, \tag{56}
$$

where we note that the unit dual quaternion $\tilde{\mathbf{q}}$ is defined as an open set over $\mathbb{R}^3$ since $\boldsymbol{t}_b \in \mathbb{R}^3$. Consider a closed subset of $\mathbb{R}^3$ as

$$
\|\boldsymbol{t}_b\| \le \delta, \tag{57}
$$

and let

$$
\tilde{\mathbf{q}}^T \tilde{\mathbf{q}} \le 1 + \frac{1}{4}\delta^2. \tag{58}
$$

Then, one can show that the left hand side of Eq. (56) is convex over the defined closed subset.

Proposition 2. *Suppose that the quadratic function $f : \tilde{\mathbf{q}} \to \mathbb{R}$ is given by*

$$
f_1(\tilde{\mathbf{q}}) = \tilde{\mathbf{q}}^T M_H \tilde{\mathbf{q}} + \|2\,E_d \tilde{\mathbf{q}}\| \cos\theta \le 0 \tag{56}
$$

with

$$\mathbf{dom}\, f_1 = \left\{ \tilde{\mathbf{q}} \in (\mathbf{S}^3 \times \mathbb{R}^3) \mid \tilde{\mathbf{q}}^T\tilde{\mathbf{q}} \;\le\; 1 + \frac{1}{4}\delta^2 \right\}, \tag{59}$$

where $\delta \in \mathbb{R}$. M_H and E_d are defined in Eqs. (51) and (55), respectively. Then f_1 is convex regardless of θ.

Proof. From the property of unit dual quaternions, Eq. (53), we have

$$\delta^2(\tilde{\mathbf{q}}^T E_u \tilde{\mathbf{q}}) - \delta^2 = 0. \tag{60}$$

Adding it into Eq. (56) yields

$$f_1 = \tilde{\mathbf{q}}^T M_H \tilde{\mathbf{q}} + \|2\, E_d \tilde{\mathbf{q}}\| \cos\theta + \delta^2(\tilde{\mathbf{q}}^T E_u \tilde{\mathbf{q}}) - \delta^2. \tag{61}$$

Note that f_1 is twice differentiable. Using the fact that

$$\tilde{\mathbf{q}}^T \left(\frac{d^2}{d\,\tilde{\mathbf{q}}^2} \|2E_d\tilde{\mathbf{q}}\| \right) \tilde{\mathbf{q}} = 0, \tag{62}$$

the quadratic form of the *Hessian* of f_1 with $\tilde{\mathbf{q}}$ now has the simple form of

$$\begin{aligned} \tilde{\mathbf{q}}^T(\nabla^2 f_1)\tilde{\mathbf{q}} &= 2(\tilde{\mathbf{q}}^T M_H \tilde{\mathbf{q}}) + 2\delta^2(\tilde{\mathbf{q}}^T E_u \tilde{\mathbf{q}}) && (63) \\ &= 2(\boldsymbol{t}_b \cdot \boldsymbol{y}_b + \delta^2). && (64) \end{aligned}$$

It is simple to show that $\boldsymbol{t}_b \cdot \boldsymbol{y}_b + \delta^2 \ge 0$ on $\mathbf{dom}\, f_1$. Since $\boldsymbol{y}_b$ is a unit vector, we observe that

$$-\|\boldsymbol{t}_b\|^2 \le \boldsymbol{t}_b \cdot \boldsymbol{y}_b \le \|\boldsymbol{t}_b\|^2. \tag{65}$$

Due to Eq. (57), we thereby conclude that $\nabla^2 f_1(\tilde{\mathbf{q}})$ is positive semidefinite and thus, f_1 is convex on $\mathbf{dom}\, f$. □

Remark 2. Similarly, we can show that in the case where the line of sight boresight vector $\boldsymbol{y}_b$ stays *outside of* θ around the direction to the target $\boldsymbol{t}_b$ with

$$\boldsymbol{t}_b \cdot \boldsymbol{y}_b \le \|\boldsymbol{t}_b\| \cos\theta \tag{66}$$

the corresponding constraint can be represented as a convex constraint on the same given domain. This can be achieved by observing that in Eq. (59), $1 \le \tilde{\mathbf{q}}^T\tilde{\mathbf{q}} \le 1 + \delta^2/4$, which allows the *Hessian* to remain positive semidefinite on its domain.

B. Glide Slope Constraints

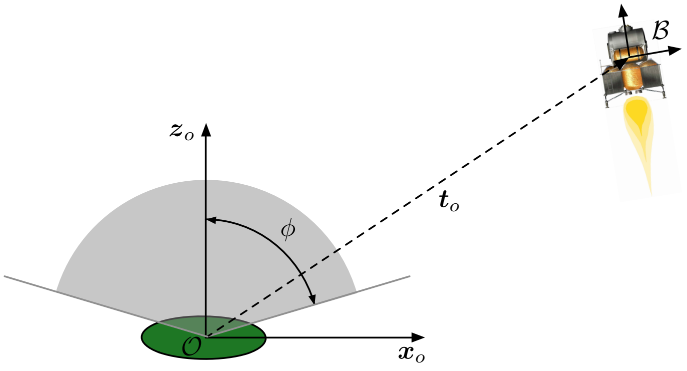

Figure 4. Illustration of the glide slope constraint. The constrained zone is defined as a cone around the inertially fixed vector $\boldsymbol{z}_o$ and angle ϕ.

The glide slope constraint is defined as a cone around the fixed vector $\boldsymbol{z}_o$ that lies in the frame $\mathcal{O}$. See the illustration in Fig. 4. Note that the glide slope constraint only depends on the position of the frame $\mathcal{B}$. Similar to the line of sight constraint, we find the condition with which the glide slope constraint is satisfied as

$$\boldsymbol{t}_o \cdot \boldsymbol{z}_o \ge \|\boldsymbol{t}_o\| \cos\phi, \tag{67}$$

where $\boldsymbol{z}_o$ and $\boldsymbol{t}_o$ denote a unitized y axis vector and a position vector to frame $\mathcal{B}$, respectively. Note that these vectors are represented with respect to frame $\mathcal{O}$. Subsequently, we assume without loss of generality that $0 < \phi \le \frac{1}{2}\pi$.

Proposition 3. *The glide slope constraint can be represented in terms of the unit dual quaternion* $\tilde{\mathbf{q}}$ *as*

$$f_2(\tilde{\mathbf{q}}) = -\tilde{\mathbf{q}}^T M_G \tilde{\mathbf{q}} + \|2\, E_d \tilde{\mathbf{q}}\| \cos\phi \le 0 \tag{68}$$

where

$$M_G = \begin{bmatrix} 0_{4\times4} & [\mathbf{z}_o]_{\otimes}{}^T \\ [\mathbf{z}_o]_{\otimes} & 0_{4\times4} \end{bmatrix}, \quad E_d = \begin{bmatrix} 0_{4\times4} & 0_{4\times4} \\ 0_{4\times4} & \mathbf{I}_4 \end{bmatrix} \tag{69}$$

with $\mathbf{z}_o = [\, \boldsymbol{z}_o^T \; 0]^T = [\,0\;0\;1\;0\,]^T$. *Then* $f_2 : \tilde{\mathbf{q}} \to \mathbb{R}$ *is convex on* $\mathbf{dom}\, f_2 = \{\tilde{\mathbf{q}} \in (\mathbf{S}^3 \times \mathbb{R}^3) \mid \tilde{\mathbf{q}}^T \tilde{\mathbf{q}} \le 1 + \frac{1}{4}\delta^2\}$.

Proof. The left hand side of Eq. (67) is rewritten as follows

$$\begin{aligned} \boldsymbol{t}_o \cdot \boldsymbol{z}_o &= (\boldsymbol{t}_o \otimes \mathbf{q}_1)^T (\boldsymbol{z}_o \otimes \mathbf{q}_1) && (70)\\ &= (\boldsymbol{t}_o \otimes \mathbf{q}_1)^T [\mathbf{z}_o]_{\otimes}\, \mathbf{q}_1 && (71)\\ &= \begin{bmatrix} \mathbf{q}_1 \\ \frac{1}{2}\boldsymbol{t}_o \otimes \mathbf{q}_1 \end{bmatrix}^T \begin{bmatrix} 0_{4\times4} & [\mathbf{z}_o]_{\otimes}{}^T \\ [\mathbf{z}_o]_{\otimes} & 0_{4\times4} \end{bmatrix} \begin{bmatrix} \mathbf{q}_1 \\ \frac{1}{2}\boldsymbol{t}_o \otimes \mathbf{q}_1 \end{bmatrix} \\ &= \tilde{\mathbf{q}}^T M_G \tilde{\mathbf{q}}, && (72) \end{aligned}$$

where M_G denotes an indefinite symmetric matrix. Thus we obtain Eq. (68). The rest of the proof is analogous to the proof in *Proposition* 2. Over the set $\mathbf{dom}\, f_2$, we can rewrite Eq. (68) as

$$f_2(\tilde{\mathbf{q}}) = \tilde{\mathbf{q}}^T M_G \tilde{\mathbf{q}} + \|2\, E_d \tilde{\mathbf{q}}\| \cos\phi + \delta^2 (\tilde{\mathbf{q}}^T E_u \tilde{\mathbf{q}}) - \delta^2. \tag{73}$$

Moreover, the quadratic form of the *Hessian* of f_2 yields

$$\tilde{\mathbf{q}}^T (\nabla^2 f_2) \tilde{\mathbf{q}} = 2(-\boldsymbol{t}_o \cdot \boldsymbol{y}_o + \delta^2). \tag{74}$$

Since $\|t_o\| = \|t_b\|$, we have

$$-\boldsymbol{t}_o \cdot \boldsymbol{y}_o + \delta^2 \ge 0 \tag{75}$$

on $\mathbf{dom}\, f_2$ from Eq. (57). Note that $\boldsymbol{y}_o$ is a unit vector. Thus, $\nabla^2 f_2(\tilde{\mathbf{q}})$ is positive semi-definite and $f_2(\tilde{\mathbf{q}})$ is convex on $\mathbf{dom}\, f_2$, which concludes the proof. □

Remark 3. The aforementioned constraint forms an analogous formulation in terms of unit dual quaternions. This is because a unit dual quaternion can be represented either with the inertial frame component or with the body frame component, and both representations lead to the same quantity, as noted from Eq. (30).

V. Constrained Powered Descent Guidance Problem

In this section, we formulate the fuel optimal powered descent guidance problem subject to rotational and translational constraints. The problem formulation consists of a cost function of states and controls to be minimized, the spacecraft rotational and translational dynamics in terms of dual quaternions, coupled rotational and translational state constraints, and general control- and state-constrained bounds. Without loss of generality, we assume that the desired position and attitude of the lander is the origin (in both translation and rotation).

Thus, the problem is formulated as follows:

$$\min \int_0^{t_f} \begin{bmatrix} \tilde{\boldsymbol{\omega}} \\ \tilde{\mathbf{q}} \end{bmatrix}^T Q \begin{bmatrix} \tilde{\boldsymbol{\omega}} \\ \tilde{\mathbf{q}} \end{bmatrix} + \tilde{\mathbf{F}}^T R \tilde{\mathbf{F}}\, dt \tag{76}$$

subject to:

$$\boldsymbol{J}\dot{\tilde{\boldsymbol{\omega}}} + \tilde{\boldsymbol{\omega}} \tilde{\times} \boldsymbol{J}\tilde{\boldsymbol{\omega}} = \tilde{\mathbf{F}} + \tilde{\mathbf{g}}_b, \tag{47}$$

$$\dot{\tilde{\mathbf{q}}} = \frac{1}{2}\tilde{\mathbf{q}} \tilde{\otimes} \tilde{\boldsymbol{\omega}}, \tag{33}$$

$$f_1(\tilde{\mathbf{q}}) = \tilde{\mathbf{q}}^T M_H \tilde{\mathbf{q}} + \|2\, E_d \tilde{\mathbf{q}}\| \cos\theta \le 0, \tag{56}$$

$$f_2(\tilde{\mathbf{q}}) = -\tilde{\mathbf{q}}^T M_G \tilde{\mathbf{q}} + \|2\, E_d \tilde{\mathbf{q}}\| \cos\phi \le 0, \tag{68}$$

$$\|\tilde{\mathbf{q}}\|^2 \le \zeta_{\tilde{\mathbf{q}}} \tag{77}$$

$$\|\omega_i\| \le \zeta_{\tilde{\boldsymbol{\omega}}_i}, \quad \|F_i\| \le \zeta_{F_i}, \quad i = 1, .., 8, \tag{78}$$

$$\tilde{\mathbf{q}}(0) = \tilde{\mathbf{q}}_0, \quad \tilde{\boldsymbol{\omega}}(0) = \tilde{\boldsymbol{\omega}}_0, \tag{79}$$

where Q and R denote the weights on states and controls, $\zeta_{\tilde{\mathbf{q}}} = 1 + \frac{1}{4}\delta^2$ denotes the boundary of feasible positions from Eq. (58), and $\zeta_{\tilde{\boldsymbol{\omega}}}$ and $\zeta_{\tilde{\mathbf{F}}}$ denote the bounds on velocities and external forces, respectively. Note that the domains of f_1 and f_2, are parameterized by the constraint $\|\tilde{\mathbf{q}}\| \leq \zeta_{\tilde{\mathbf{q}}}$. In this formulation we consider the rotational and translational motion constraints Eqs. (56),(68) as hard constraints which must be satisfied during the descent time interval .

VI. Controller Design

In this section, based on the spacecraft dynamics and constraint parameterizations in unit dual quaternions presented in the previous section, we formulate a piece-wise affine model predictive controller and present sufficient feasibility and stability conditions for the resulting closed loop system. The piece-wise affine model predictive control computes a sub-optimal trajectory to minimize a given cost function and satisfy constraints over a finite and receding horizon based on sampled piece-wise affine system- obtained by approximating a nonlinear continuous dynamics.

A. Piece-wise Affine Model

This section describes the piece-wise affine model approximation of the aforementioned nonlinear dynamic models written in dual quaternions for the subsequent development of the PWA-MPC controller. The continuous time differential equations Eqs. (33) and (38) are discretized in time with sampling period Δt. The equations are calculated as

$$\boldsymbol{J}\tilde{\boldsymbol{\omega}}(t+1) = \boldsymbol{J}\tilde{\boldsymbol{\omega}}(t) - \Delta t\,[\tilde{\boldsymbol{\omega}}(t)]_{\tilde{\times}}(\boldsymbol{J}\tilde{\boldsymbol{\omega}}(t)) + \Delta t\,\tilde{\mathbf{F}}(t) + \Delta t\tilde{\mathbf{I}}_8[m\tilde{\mathbf{q}}^*\tilde{\otimes}\tilde{\mathbf{I}}_8\tilde{\mathbf{g}}_o]_{\tilde{\otimes}}\tilde{\mathbf{q}} \tag{47d}$$

$$-\frac{\Delta t}{2}[\tilde{\mathbf{q}}(t)]_{\tilde{\otimes}}\tilde{\boldsymbol{\omega}}(t+1) + \tilde{\mathbf{q}}(t+1) = \tilde{\mathbf{q}}(t), \tag{33d}$$

where $\boldsymbol{J}$ denotes an inertia matrix in the body frame, $[\cdot]_{\tilde{\times}}$ and $[\cdot]_{\tilde{\otimes}}$ are defined in the previous section. Note that we have assumed that the time sequence of the discretized equations is

$$\tilde{\mathbf{F}}(t) \implies \tilde{\boldsymbol{\omega}}(t+1)\ ,\ \tilde{\mathbf{q}}(t+1) \tag{80}$$

in order for the system to be controllable. By defining the new state variables as

$$x(t) = \begin{bmatrix} \tilde{\boldsymbol{\omega}}(t) \\ \tilde{\mathbf{q}}(t) \end{bmatrix}_{16\times 1} \quad \text{and} \quad u(t) = \tilde{\mathbf{F}}(t), \tag{81}$$

these equations can be represented as

$$M(t)x(t+1) = A'(t)x(t) + B'u(t) \tag{82}$$

with

$$M(t) = \begin{bmatrix} \boldsymbol{J} & \mathbf{0}_{8\times 8} \\ -\frac{\Delta t}{2}[\tilde{\mathbf{q}}(t)]_{\tilde{\otimes}} & \mathbb{I}_8 \end{bmatrix}, \quad A'(t) = \begin{bmatrix} \boldsymbol{J} - \Delta t\,[\boldsymbol{J}\tilde{\boldsymbol{\omega}}(t)]^*_{\tilde{\times}} & \Delta t\tilde{\mathbf{I}}_8[m\tilde{\mathbf{q}}^*\tilde{\otimes}\tilde{\mathbf{I}}_8\tilde{\mathbf{g}}_o]_{\tilde{\otimes}} \\ \mathbf{0}_{8\times 8} & \mathbb{I}_8 \end{bmatrix}, \quad B' = \begin{bmatrix} \Delta t\mathbb{I}_8 \\ \mathbf{0}_{8\times 8} \end{bmatrix}. \tag{83}$$

Moreover, Eq. (82) can be represented as a standard pice-wise affine system as

$$\begin{aligned} x(t+1) &= M(t)^{-1}A'(t)x(t) + M(t)^{-1}B'u(t) \\ &= A_t x(t) + B_t u(t) \end{aligned} \tag{84}$$

Note that the matrix M is invertible by **Lemma 1** and the system Eq. (84) is stabilizable.

B. PWA-MPC

In the PWA-MPC setup, once a sub-optimal state and control trajectory has been computed, the control corresponding to the first discrete time interval is implemented. The optimization horizon then recedes by one time step and the calculation process is repeated at each time step with the current state as the initial condition. This loop generates a feedback action that can effectively compensate the uncertainties and disturbances. Furthermore, at each time step, constraints and even the system model are allowed to be time-varying.

In this section, we introduce a piece-wise affine model predictive control strategy that generates a piece-wise constant control to stabilize the above-mentioned discrete time system. In particular, we consider the following discrete piece-wise affine system sampled at time instant t as

$$x_{t+1} = A_t x_t + B_t u_t, \tag{85}$$

where $x_t \in \mathbb{R}^n$ and $u_t \in R^m$ denote the state vector and the control vector, respectively. The matrices A_t and B_t are sampled and remain constant during $[t, t + \Delta t]$. This PWA system is subject to the constraints

$$x_t \in \mathcal{X}, \quad \text{and} \quad u_t \in \mathcal{U} \tag{86}$$

where $\mathcal{X}$ and $\mathcal{U}$ are assumed to be convex sets.

For the problem of regulation to the origin, we consider a model predictive control to minimize the cost function $J_t : \mathbb{R}^n \times \mathbb{R}^{Nm} \to \mathbb{R}_+$ defined by

$$J_t = \sum_{i=1}^{N-1} \left(x^T_{(t|i)} Q x_{(t|i)} + u^T_{(t|i)} R u_{(t|i)} \right) + x^T_{(t|N)} P x_{(t|N)}, \tag{87}$$

where $x_{(t|i)} \in \mathbb{R}^n$ denotes the state vector, $u_{(t|i)} \in \mathbb{R}^m$ denotes the control input, $N \in \mathbb{Z}_+$ denotes the finite horizon and $Q, R, P \in \mathbb{S}^n_+$ denote weighting matrices on state, control input, and final time state, respectively. Then the PWA-MPC solves the following optimization problem:

$$\min_{U_t} \quad J_t(x_{(t|0)}, U_t) \tag{88}$$

$$\text{subject to} \quad x_{(t|i+1)} = A_t x_{(t|i)} + B_t u_{(t|i)}, \quad i = 0, ..., N-1 \tag{88a}$$

$$x_{(t|i)} \in \mathcal{X}_i, \quad i = 0, ..., N-1 \tag{88b}$$

$$x_{(t|N)} \in \mathcal{X}_f, \tag{88c}$$

$$u_{(t|i)} \in \mathcal{U}, \quad i = 0, ..., N-1 \tag{88d}$$

$$x_{(t|0)} = x_t(0), \tag{88e}$$

where $U_t = [u_{t|0}, ..., u_{t|N-1}]$ denotes the control vector to be optimized at time t, whereas $X_t = [x_{t|0}, ..., x_{x|N}]$ denotes the predicted state vector corresponding to U_t; $x_{(t|0)} = x(t)$ denotes the first state fed-back by a measurement at time t and $\mathcal{X}_i$ and $\mathcal{U}$ denote the given convex sets for states and control, respectively. The matrices A_t and B_t are assumed to be bounded and sampled at each time instance t. Given $x_{(t|0)}$, if the optimization problem is feasible, a set of optimal controls with the finite horizon N is found as

$$U_t^* = [u^*_{t|0}, u^*_{t|1}, ..., u^*_{t|N-2}, u^*_{t|N-1}]^T, \tag{89}$$

and for $i = 0, ..., N$, the optimal predicted-state trajectory $X_t^* = [x^*_{t|1}, ..., x^*_{t|N}]^T$ is obtained by applying U_t^* to the system. Then, the first element of U_t^* is applied to the system as

$$u_t^* = u^*_{t|0}. \tag{90}$$

The corresponding closed-loop PWA system of Eq. (88a) is thereby given as

$$x_{t+1} = A_t x_t + B_t u_t^* = f_{t|cl}(x_t, u_t^*), \quad x_t = x(t). \tag{91}$$

The optimization problem is then repeated at time $t + 1$ over one step receded horizon, based on the newly available measurement $x_{t+1|0} = x_{t+1}(0)$. Note that the cost function Eq. (87) is convex. When state constraints Eq. (88b) and control constraints Eq. (88d) are convex, the optimization problem is convex and can efficiently be solved by available convex programming solvers [15].

C. Feasibility and stability Analysis

In the PWA-MPC approach, the optimization problem Eq. (88) is solved over a finite horizon repeatedly at each time step. Distinct from the properties inherent from an infinite-horizon closed loop controller synthesis procedure such as a LQR, the PWA-MPC approach faces at least two technical issues. First, the PWA-MPC procedure may lead to a situation where after a few steps, the finite horizon optimal control problem may become infeasible, i.e., there required U_t in Eq. (88) at certain time step may not exist. Second, the computed controls may not converge to the origin, i.e., the closed loop system is not guaranteed to be asymptotically stable.

In general, stability and feasibility are not automatically ensured by the PWA-MPC approach, and additional conditions are required. Such conditions hinge on how the terminal weight P in Eq. (87) and the terminal constraint set $\mathcal{X}_f$ are chosen such that PWA-MPC procedure does not encounter the aforementioned stability and feasibility issues.

In this section, we investigate the conditions for the persistent feasibility and the uniform asymptotic stability of the origin for the closed system using the PWA-MPC laws described above. First we address the necessary ingredients for the persistent feasibility issue.

Assumption 1. *The state and input constraint sets $\mathcal{X}$ and $\mathcal{U}$ are convex*

$$\mathcal{X} = \{x \in \mathbb{R}^n | M(x) \leq 0\}, \quad \text{and} \quad \mathcal{U} = \{u \in \mathbb{R}^m | N(u) \leq 0\}, \tag{92}$$

where $M : \mathbb{R}^n \to \mathbb{R}$ and $N : \mathbb{R}^m \to \mathbb{R}$ denote convex functions.

Definition 1 (Reachability)**.** *At time instance t, the set $\mathcal{X}_i \subseteq \mathcal{X}$ denotes the set of states $x_{(t|i)}$ for which $\mathcal{X}_{i+1}$ is reachable, i.e.,*

$$\mathcal{X}_i = \{x \in \mathcal{X} \,|\, \exists u_{(t|i)} \in \mathcal{U} \text{ such that } A_t x + B_t u_{(t|i)} \in \mathcal{X}_{i+1}\}, \quad i = 1, ..., N. \tag{93}$$

In particular, for $i = N$, $\mathcal{X}_N = \mathcal{X}_f$ denotes a terminal set.

Definition 2 (Persistence)**.** *The collection $\mathcal{X}_0, .., \mathcal{X}_i, .., \mathcal{X}_f$ for the PWA-MPC is persistently feasible.*

The next definition complements the notion of persistence.

Definition 3 (Control Invariance)**.** *A set $\mathcal{C} \subseteq \mathcal{X}$ is said to be a control invariant set for the system Eq. (88a) if for all permissible control inputs, the one step state evolution remains in $\mathcal{C}$, i.e.,*

$$\mathcal{C} = \{x \in \mathcal{C} \subseteq \mathcal{X} \,|\, \forall u \in \mathcal{U} \text{ such that } A_t x + B_t u \in \mathcal{C}\}. \tag{94}$$

Proposition 4. *Consider the PWA-MPC Eq. (88) with the finite horizon $N \geq 1$. If the terminal state set $\mathcal{X}_f$ is control invariant for the system Eqs. (88a)-(88e), then the PWA-MPC is persistently feasible.*

Proof. If $\mathcal{X}_f = \mathcal{X}_N$ is a control invariant set for the system, we have

$$\mathcal{X}_{N-1} = \{x \in \mathcal{X} \,|\, \forall u_{(t|i)} \in \mathcal{U} \text{ such that } A_t x + B_t u_{(t|i)} \in \mathcal{X}_N = \mathcal{C}\}, \tag{95}$$

which implies that $\mathcal{X}_N \subseteq \mathcal{X}_{N-1}$ is also a control invariant set by **Definition 3**. By induction, we conclude that the control invariance of $\mathcal{X}_N$ implies the control invariance of $\mathcal{X}_{N-1}, \mathcal{X}_{N-2}, \ldots, \mathcal{X}_0$. Thus, the PWA-MPC is persistently feasible. □

The stability of proposed PWA-MPC is investigated by using a Lyapunov technique. In particularly, we show that if the terminal set is appropriately chosen, the cost function J_t is in fact a Lyapunov function. This is analogous to the notion of cost-to-go in optimal control.

Theorem 2 (Uniform Asymptotic Stability for Discrete-Time Systems[16, 17])**.** *Given the closed-loop system Eq. (91), consider $\mathcal{X}_f \in \mathbb{R}^n$ to be the terminal set containing the equilibrium state $x = 0$. Assume that there exists a Lyapunov function $J(x_t) : \mathbb{R}^n \times \mathbb{Z}_+ \to R_+$ in $x \in \mathcal{X}_f$ such that*

1. *$J(x_t) = 0$ and $J(x_t)) > 0$, $\forall x \in \Omega \setminus \{0\}$*
2. *The rate of change, $J(x_{t+1}) - J(x_t) \leq -\alpha(x_t)$, $\forall x \in \Omega \setminus \{0\}$,*

where $\alpha(x_t)$ denotes a positive definite function. Then the origin $x = 0$ is asymptotically stable in $\mathcal{X}_f$.

Definition 4. *We denote by $J_t^*(x_t)$ the value of the cost function $J_t(x_t, U_t)$ in Eq. (87) at time t when the control law U_t^* and initial states x_t have been applied to Eq. (88a).*

Assumption 2. *An assumption needed for our subsequent analysis is that the difference between two PWA systems (A_t, B_t) and (A_{t+1}, B_{t+1}) is small. That is, given the control input for the PWA system sampled at t, the one step evolution of the system stays in $\mathcal{X}_f$ when the system is sampled at $t + 1$, i.e.,*

$$u_t^* : f_{t|cl}(x_t, u_t^*) \in \mathcal{X}_f \Longrightarrow u_t^* : f_{t+1|cl}(x_{t+1}, u_t^*) \in \mathcal{X}_f \tag{96}$$

Proposition 5. *Consider the PWA system Eq. (85) subject to constraints Eq. (86). Suppose that PWA-MPC law Eqs. (87)-(90) have been applied under the* **Assumptions 1** *and* **2**. *Moreover, let the corresponding PWA-MPC be persistently feasible. If for the initial condition $x_{(t|0)} \in \mathcal{X}_0$ one has*

$$x^T_{(t+1|N)} P x_{(t+1|N)} + \sum_{i=0}^{N-1} x^T_{(t+1|i)} Q x_{(t+1|i)} - x^T_{(t|N)} P x_{(t|N)} \leq 0, \tag{97}$$

then the closed-loop system Eq. (91) converges to the origin asymptotically.

Proof. First, since we assume that the system is persistently feasible, we have $\mathcal{X}_0$, $\mathcal{X}_1$,..., $\mathcal{X}_{N-1}$ are control invariant sets and the PWA-MPC problem remains feasible. For $x_{(t|0)} \in \mathcal{X}_0$, consider the optimized cost function $J_t^*(x_{(t|0)}, U_t^*)$ at time t as

$$J_t^*(x_{(t|0)}, U_t^*) = x_{(t|N)}^T P x_{(t|N)} + \sum_{i=0}^{N-1} x_{(t|i)}^T Q x_{(t|i)} + {u^*}_{(t|i)}^T R u^*_{(t|i)}, \tag{98}$$

where $U_t^* = [u^*_{t|0}, ..., u^*_{t|N-1}]^T$ denotes an optimal control sequence to minimize the cost function subject to constraints. After the implementation of the first control $u^*_{t|0}$, we obtain

$$x_{(t|1)} = A_t x_{(t|0)} + B_t u^*_{(t|0)}, \tag{99}$$

where we assume that this predicted state $x_{(t|1)}$ using the optimal control $u^*_{(t|0)}$ is equal to the measured state $x_{(t+1|0)}$ at the next time instance $t+1$, as $x_{(t|1)} \longrightarrow x_{(t+1|0)}$. Consider now the PWA-MPC problem at time instance $t+1$. Our aim is to find the upper bound on $J^*_{t+1}(x_{t+1}, U_{t+1})$ as we impose the same control sequence U_t^* to the PWA system at $t+1$. Note that due to **Assumption 2**, the set of control U_t^* is still feasible. Moreover, U_{t+1} has the form

$$U_{t+1} = [\, u^*_{(t|0)}, u^*_{(t|2)}, ..., u^*_{(t|N-2)}, u^*_{(t|N-1)} \,]^T. \tag{100}$$

Note that since two systems are sampled at t and $t+1$, the trajectories generated by the U_t do not overlap. Therefore, U_{t+1} is not optimal and $J^*_{t+1}(x_{t+1}, U_{t+1})$ is upper bounded by $J_{t+1}(x_{t+1}, U_t^*)$ as

$$\begin{aligned} J^*_{t+1}(x_{t+1}, U_{t+1}) \leq J_{t+1}(x_{t+1}, U_t^*) =& x^T_{(t+1|N)} P x_{(t+1|N)} + \sum_{i=0}^{N-1} x^T_{(t+1|i)} Q x_{(t+1|i)} \\ &+ J_t^*(x_t, U_t) - x^T_{(t|N)} P x_{(t|N)} - \sum_{i=0}^{N-1} x^T_{(t|i)} Q x_{(t|i)}. \end{aligned} \tag{101}$$

It now follows that if the condition Eq. (97) is satisfied, it follows that

$$J^*_{t+1}(x_{t+1}, U_{t+1}) - J_t^*(x_t, U_t) \leq - \sum_{i=0}^{N-1} x^T_{(t|i)} Q x_{(t|i)}, \tag{102}$$

where $Q \in \mathbb{S}^n_+$ ensures that $J_t^*(x_t, U_t)$ strictly decreases along the trajectory generated by the PWA-MPC Eq. (91) for $x(t) \in \mathcal{X}_0 \setminus \{0\}$. Since the cost function satisfies all requirements for being Lyapunov function, using **Theorem 2** it follows that the origin in $\mathcal{X}_f$ is asymptotically stable. □

Thus, the stability condition requires finding an appropriate terminal cost P that satisfies the inequality Eq. (97) given as

$$x^T_{(t+1|N)} P x_{(t+1|N)} + \sum_{i=0}^{N-1} x^T_{(t+1|i)} Q x_{(t+1|i)} - x^T_{(t|N)} P x_{(t|N)} \leq 0. \tag{97}$$

Since the PWAMPC is persistently feasible, we have

$$x_{(t+1|N)} \text{ and } x_{(t|N)} \in \mathcal{X}_f. \tag{103}$$

In the case of linear time invariant systems, i.e., $(A_t, B_t) = (A_{t+1}, B_{t+1})$, the state trajectories corresponding to the same optimal control U_t^* overlap; thus we have

$$x_{(t+1|N)} = x_{(t|N+1)}, \text{ and } x_{(t+1|N)} = A x_{(t|N)} + B u_{(t|N)}, \tag{104}$$

Due to Eq. (103), there exists a sequential PWA-MPC law, and thereby $u^*_{(t|N)}$ satisfies

$$x_{(t+1|N)} = x_{(t|N+1)} = A x_{(t|N)} + B u^*_{(t|N)} = A_{(t|cl)} x_{(t|N)}. \tag{105}$$

By plugging the above equation into the stability condition Eq. (97), we obtain

$$x^T_{(t|N)} \left[A^T_{(t|cl)} P A_{(t|cl)} + \sum_{i=0}^{N-1} A^T_{(t|cl)} Q A_{(t|cl)} - P \right] x_{(t|N)} \leq 0, \tag{106}$$

and P can be found by solving the discrete Lyapunov equation $A^T_{(t|cl)} P A_{(t|cl)} + \sum_{i=0}^{N-1} A^T_{(t|cl)} Q A_{(t|cl)} - P = 0$.

However, the PWA system is time varying, i.e., $(A_t, B_t) \neq (A_{t+1}, B_{t+1})$, and the state trajectories corresponding to the same optimal control U^*_t do not overlap. We thus assume that the PWA system is slowly varying such that

$$\|P^{\frac{1}{2}} x_{(t+1|N)}\|^2 = \|P^{\frac{1}{2}} x_{(t|N+1)}\|^2 + \delta_N \|x_{(t|N+1)}\|^2, \tag{107}$$

$$\|Q^{\frac{1}{2}} x_{(t+1|i)}\|^2 = \|Q^{\frac{1}{2}} x_{(t|i+1)}\|^2 + \delta_i \|x_{(t|i+1)}\|^2, \tag{108}$$

where $P^{\frac{1}{2}} x_{(\cdot)}$ and $Q^{\frac{1}{2}} x_{(\cdot)}$ denote the weighted states by matrices P and Q, respectively. We note that $\delta_0, ..., \delta_{N-1}$ and δ_N denote the positive constants representing the 2-norm square difference between the two states; see Fig. 5. We define the bound on δ_i's as

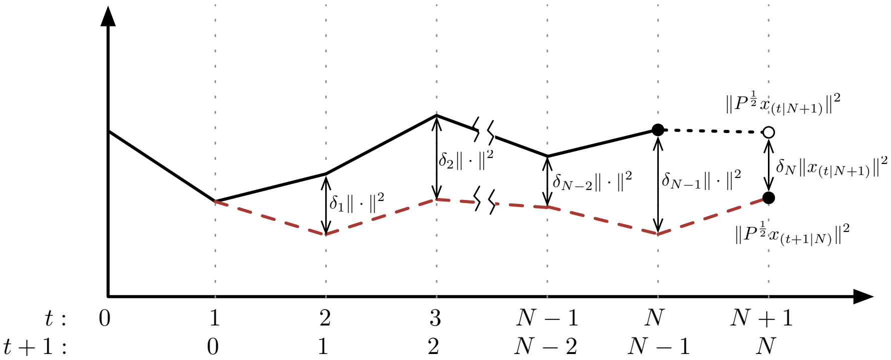
$\|P^{\frac{1}{2}} x_{(t|N+1)}\|^2$
$\delta_2 \| \cdot \|^2$
$\delta_{N-2} \| \cdot \|^2$
$\delta_{N-1} \| \cdot \|^2$
$\delta_N \|x_{(t|N+1)}\|^2$
$\delta_1 \| \cdot \|^2$
$\|P^{\frac{1}{2}} x_{(t+1|N)}\|^2$
t : 0 1 2 3 $N-1$ N $N+1$
$t+1$: 0 1 2 $N-2$ $N-1$ N

Figure 5. 2-norm square difference between weighted states over finite horizon

$$\max(\delta_0, ..., \delta_{N-1}) \doteq \delta_{max}. \tag{109}$$

By substituting Eqs. (107)-(109) and the PWA-MPC law Eq. (105) into the stability condition Eq. (97), we obtain

$$x^T_{(t+1|N)} P x_{(t+1|N)} + \sum_{i=0}^{N-1} x^T_{(t+1|i)} Q x_{(t+1|i)} - x^T_{(t|N)} P x_{(t|N)} \tag{110}$$

$$\leq x^T_{(t|N)} \left[A^T_{(t|cl)} P A_{(t|cl)} + \sum_{i=0}^{N-1} A^T_{(t|cl)} Q A_{(t|cl)} - P + \delta \mathbf{I}_n \right] x_{(t|N)}, \tag{111}$$

where $\delta = \delta_N + N\delta_{max}$ and the inequality stems from the fact that

$$\sum_{i}^{N-1} \delta_i \leq N\delta_{max}. \tag{112}$$

Solving the discrete Lyapunov equation

$$A^T_{(t|cl)} P A_{(t|cl)} + \sum_{i=0}^{N-1} A^T_{(t|cl)} Q A_{(t|cl)} - P + \delta \mathbf{I}_n = 0, \tag{113}$$

yields the solution to the desired weighting matrix P.

D. Constrained powered Descent Guidance via PWAMPC

In this section, we formulate the constrained powered descent guidance problem using the design technique described in the previous section. The optimization problem to solve PWA-MPC is defined as follows:

$$\min_{U_t} \quad J_t(x_{(t|0)}, U_t) = \sum_{k=1}^{N-1} \left(x^T_{(t|i)} Q x_{(t|i)} + u_{(t|i)} R u_{(t|i)} \right) + x^T_{(t|N)} P x_{(t|N)} \tag{114}$$

$$\begin{aligned}
\text{subject to} \quad & x_{(t|i+1)} = A_t x_{(t|i)} + B_t u_{(t|i)}, \quad i = 0, ..., N-1 && (84)\\
& x^T_{(t|i+1)} M'_H x_{(t|i+1)} + \|2\, E'_d x_{(t|i+1)}\| \cos\theta \leq 0, && (56)\\
& -x^T_{(t|i+1)} M'_G x_{(t|i+1)} + \|2\, E'_d x_{(t|i+1)}\| \cos\phi \leq 0, && (68)\\
& \|x_{(t|i+1)}\|^2 \leq \zeta_x && (115)\\
& \|u_{(t|i)}\|^2 \leq \zeta_u, && (116)\\
& x_{(t|0)} = x_0(t), && (117)
\end{aligned}$$

with

$$x_{(t|i)} = \begin{bmatrix} \tilde{\boldsymbol{\omega}}_{(t|i)} \\ \tilde{\mathbf{q}}_{(t|i)} \end{bmatrix}_{16\times1} \quad \text{and} \quad u_{(t|i)} = \tilde{\mathbf{F}}_{(t|i)} = \begin{bmatrix} \boldsymbol{F}_{(t|i)} \\ \boldsymbol{T}_{(t|i)} \end{bmatrix}, \tag{118}$$

where N denotes the prediction and the control horizon; ζ_x and ζ_u on the other hand denote the state and control bounds, respectively. Since the two types of constraints are acting on the position, the corresponding matrices are re-defined as

$$M'_H = \begin{bmatrix} 0 & 0 \\ 0 & M_H \end{bmatrix}, \quad M'_G = \begin{bmatrix} 0 & 0 \\ 0 & M_G \end{bmatrix}, \quad E'_d = \begin{bmatrix} 0 & 0 \\ 0 & E_d \end{bmatrix}. \tag{119}$$

We denote as $U^*_t = [u^*_{(t|0)}, ..., u^*_{(t|N-1)}]$ the sequence of optimal control laws over the finite horizon with a predicted model. The first control input $u^*_{(t|0)}$ is then applied to the system. Subsequently, at the next time step $t+1$, the optimization problem is solved over a shifted finite horizon based on the updated system model A_{t+1}, b_{t+1} and the new initial condition x_{t+1}.

VII. Numerical Simulations

In this section, we present a simulation example to demonstrate the viability of the proposed PWA-MPC approach for the Mars pinpoint landing problem in the presence of constraints. In the scenario considered, it has been assumed that the Mars lander is required not to violate two types of rotational and translational constraints. The simulation is carried out using Matlab's generic nonlinear solver with an interior point algorithm. Even though the simulation was not setup using a custom solver optimized for landing problems, the rapid execution time and the convergence guarantees make the proposed approach a viable candidate for onboard guidance control for Mars landing. The spacecraft's physical properties are specified as:

$$\text{Moment of inertia: } J = \texttt{diag}[\,553.3, \quad 779.8, \quad 371.4\,]\,\text{kg}\cdot\text{m}^2, \tag{120}$$

$$\text{Mass: } m = 770.07\,\text{kg}, \tag{121}$$

$$\text{Gravity on Mars: } g_o = [\,0, \quad 0, \quad -3.7114\,]\,\text{m/s}^2. \tag{122}$$

The Mars lander configuration with the line of sight vector is depicted in Fig. (6). The initial and final conditions are given in Table 1. Detailed parameters for four rotation and translation related constraints are also presented. State bounds are reasonably defined and addressed in Table 2 as well. Based on the initial and desired attitudes and positions, the corresponding initial and desired conditions in unit dual quaternions are computed as

$$\begin{aligned}
\tilde{\mathbf{q}}(0) &= \begin{bmatrix} \mathbf{q}(0) \\ \frac{1}{2}\boldsymbol{t}_o(0) \otimes \mathbf{q}(0) \end{bmatrix} && (123)\\
&= [-0.38,\ 0.49,\ -0.40,\ 0.67,\ 187.31,\ -93.75,\ 269.61,\ 336.95]^T \\
\tilde{\mathbf{q}}(t_f) &= [0,\ 0,\ 0,\ 1.00,\ 0,\ 0,\ 0,\ 0]^T, && (124)
\end{aligned}$$

where both initial and desired attitudes $\mathbf{q}(0), \mathbf{q}(t_f)$ and positions $\boldsymbol{t}_o(0), \boldsymbol{t}_o(t_f)$ are randomly chosen from a set of corresponding dual quaternions that satisfy all convex constraints.

In Fig. (7), the overall trajectory of the Mars lander is depicted along with the representation of the cone for the line of sight constraint. The desired position for landing is denoted by a blue circle in this figure. We note that different from the point mass model, the shape of the overall trajectory is affected by the initial and desired attitudes. This is due to the fact that the rotational and translational motions are combined with dual quaternions and the shortest path between two points on the unit dual quaternion manifold is exhibited as a screw motion in $\mathsf{SE}(3)$. By weighting the second part of the dual quaternion pertaining to the lander's translational dynamics, one can dictate the convergence rate between the rotational and translational motions. Fig. (8) depicts the time histories for the position and attitude in unit quaternions of the Mars lander observed with respect to an inertially fixed frame while Fig. (9) depicts the time histories for translational and angular velocities in the body-fixed frame. Note that the translational velocities for the first 100 seconds have been restricted by their upper and lower bounds given in Table 2. The translational forces and torques acting along the body-fixed axes are represented in Fig. (10). Note that the translational forces never vanish since there exists a gravitational force acting downwards and forces acting along $\boldsymbol{z}_b$ compensate such a gravitational force to maintain the desired altitude in the later part of the simulation.

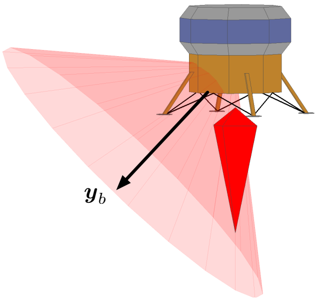

Figure 6. Mars lander configuration with the line of sight vector $\boldsymbol{y}_b$

Δt	1 second
Predition horizon	10
Simulation time	100 seconds
Initial Attitude $\mathbf{q}(0)$	[-0.3841, 0.4913, -0.4009, 0.6710]
Initial Position $\boldsymbol{t}_o(0)$	[700, -400, 520] m
Initial Velocity $\boldsymbol{v}_b(0)$	[-9.15, 19.34, -29.16] m/s
Initial Angular Velocity $\boldsymbol{\omega}_b(0)$	[0, 0, 0] rad/s
line of sight	$\boldsymbol{y}_b$ = [0, $\sqrt{2}/2$, -$\sqrt{2}/2$], $\theta = 50°$
Glide slope	$\boldsymbol{z}_o$ = [0, 0, 1], $\phi = 65°$

Table 1. Constrained powered descent simulation parameters

$\|\boldsymbol{t}_b\|$	$\leq$ 1000 m
$\lvert F_x\rvert, \lvert F_y\rvert, \lvert F_z\rvert$	$\leq$ 5000 N
$\lvert M_x\rvert, \lvert M_y\rvert, \lvert M_z\rvert$	$\leq$ 5 N $\cdot$ m
$\lvert v_x\rvert, \lvert v_y\rvert, \lvert v_z\rvert$	$\leq$ 40 m/s
$\lvert \omega_x\rvert, \lvert \omega_y\rvert, \lvert \omega_z\rvert$	$\leq$ 0.04 rad/s

Table 2. Simulation state bounds

Fig. (11) shows the trace of the deviation angle over time from the line of sight vector $\boldsymbol{y}_b$ to the desired landing location. As we expect from the configuration of the Mars lander shown in Fig. 6, pure vertical descent arises around the boundary of the line of sight constraint cone. The glide slope angle constraint is defined as 65° and Fig. 12 represents the glide slope angle over time.

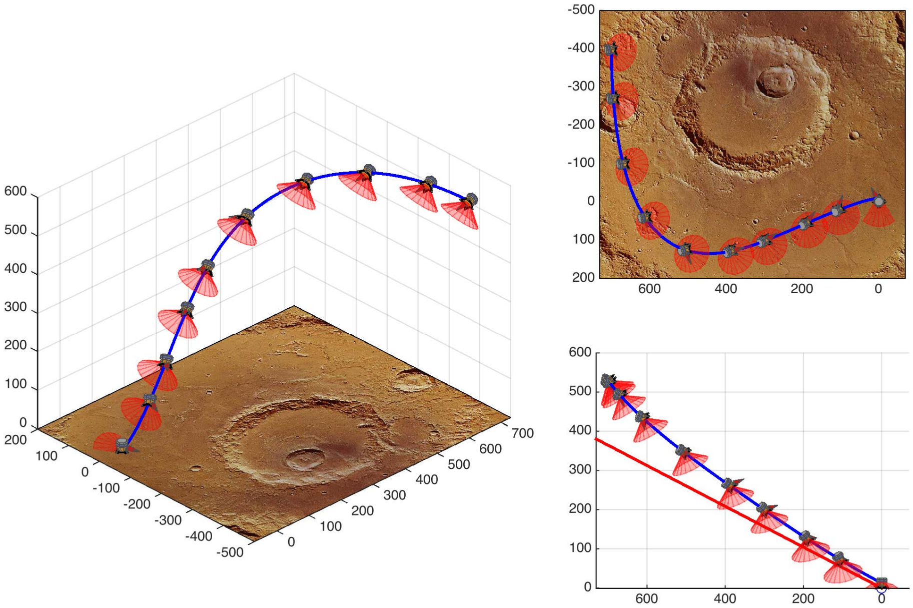

Figure 7. Overall trajectory of Mars lander's powered descent for pin-point landing along with the representation of the line of sight cone and glide slope.

VIII. Conclusions and future work

In this paper, we utilized unit dual quaternions that parameterize rotational and translational motion of a rigid spacecraft to develop an efficient algorithm for powered descent guidance in the presence of translational and rotational constraints. Subsequently, we derived a piece-wise affine model (PWA) for the rotational and translational control synthesis based on model predictive control (MPC). The resulting PWA-MPC approach was shown to provide a unifying framework through which various classes of constraints can addressed for powered descent, in addition to a wide array of problems for autonomous spacecraft operations.

References

[1] R. D. Braun, and R. M. Manning, "Mars Exploration Entry, Descent and Landing Challenges," *Journal of Spacecraft and Rockets,* vol. 44, no. 2, pp. 310-323, 2007.

[2] B. A. Steinfeldt, M. J. Grant, D. A. Matz and R. D. Braun, "Guidance, Navigation, and Control System Performance Trades for Mars Pinpoint Landing," *Journal of Spacecraft and Rockets,* vol. 47, no. 1, pp. 188-198, 2010.

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

[4] A. T. Yang, "Application of Quaternion Algebra and Dual Numbers to the Analysis of Spatial Mechanisms," Ph.D dissertation, Columbia University, 1963.

[5] J. M. McCarthy, *An Introduction to Theoretical Kinematics*, MIT Press, 1990.

[6] Y. Wu, X. Hu, D. Hu, and J. Lian, "Strapdown Inertial Navigation System Algorithms Based on Dual Quaternions," *IEEE Transaction on Aerospace and Electronic Systems*, vol. 41, no. 1. pp. 110-132, 2005.

[7] R. R. Sostaric, and J. R. Rea, "Powered Descent Guidance Methods for the Moon and Mars," *AIAA Guidance, Navigation, and Control Conference and Exhibit,* AIAA Paper 2005-6287, 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] 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.

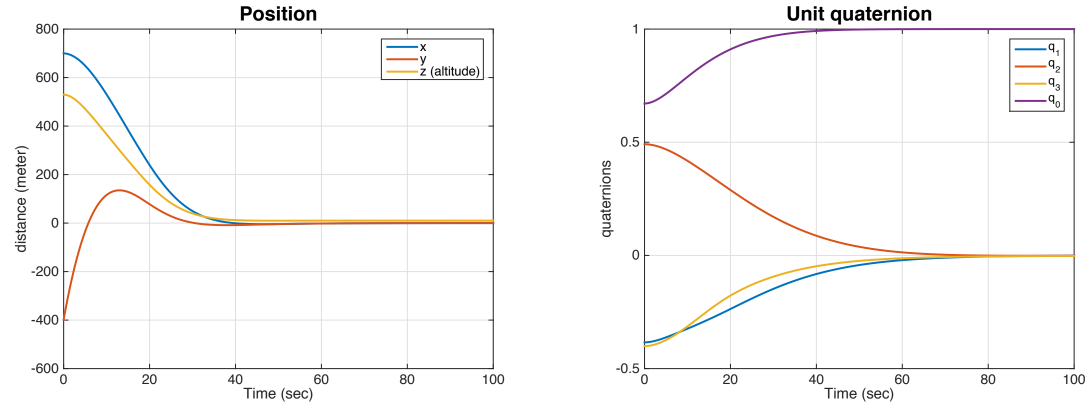

Figure 8. Time histories for the position and attitude of Mars lander observed in an inertially fixed frame.

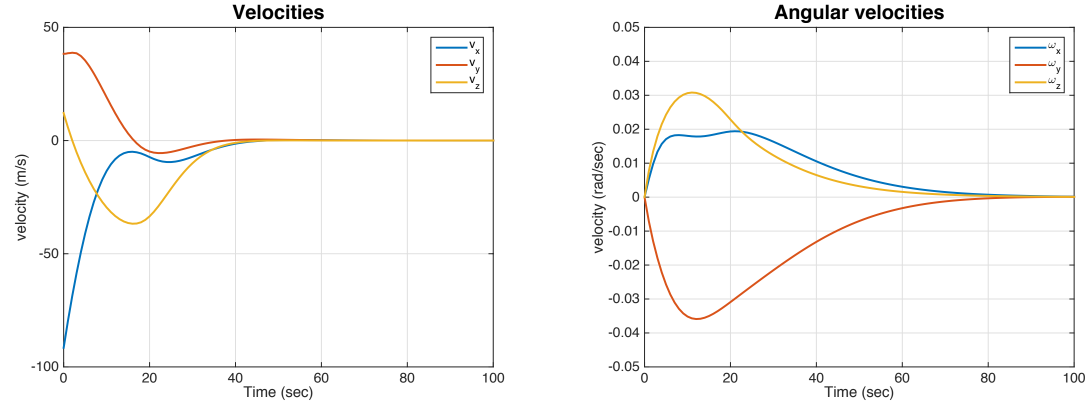

Figure 9. Time histories for the translational and angular velocities in the body-fixed frame.

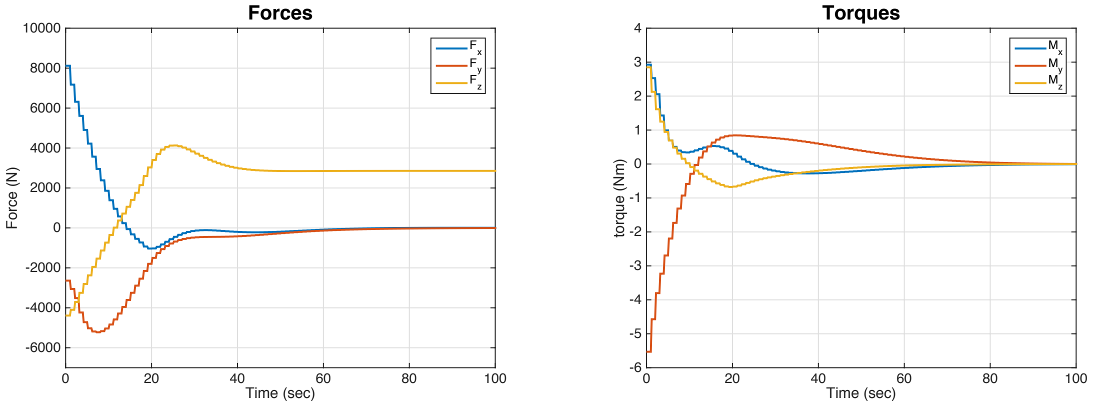

Figure 10. Time histories for the translational forces and torques expressed in the body-fixed frame.

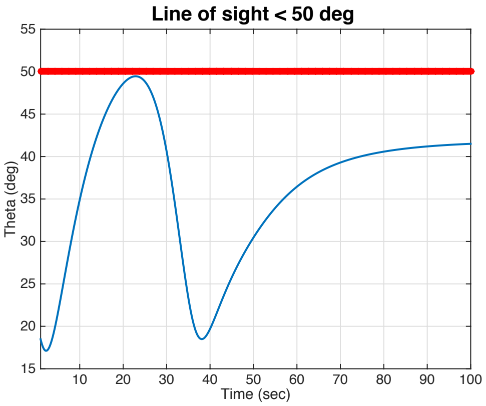

Figure 11. Trace of deviating angle over time from line of sight vector

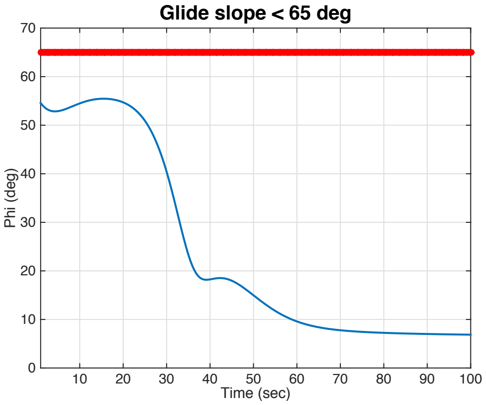

Figure 12. Glide slop angle over time

[10] B. Acikmese and S. R. Ploen, “A Powered Descent Guidance Algorithm for Mars Pinpoint Landing,” *AIAA Guidance, Navigation, and Control Conference,* San Francisco, CA, 2005.

[11] B. Acikmese, L. Blackmore, D. P. Scharf and A. Wolf, “Enhancements on the Convex Programming Based Powered Descent Guidance Algorithm for Mars Landing,” *AIAA/AAS Astrodynamics Specialist Conference and Exhibit,* Honolulu, Hawaii, 2008.

[12] S. Liu, and G. Trenkler, “Hadamard, Khatri-Rao, Kronecker and other Matrix Products,” *International Journal of Information and Systems Sciences,* vol. 4, no. 1, pp. 160-177, 2008.

[13] R. F. Stengel, *“Flight Dynamics,”* Princeton University Press, 2004.

[14] J. R. Silvester, “Determinants of Block Matrices,” *The Mathematical Gazette,* Vol. 84, No. 501 , pp. 460-467, Nov., 2000.

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

[16] R.E. Kalman and J.E. Bertram, “Control system analysis and design via the second method of Lyapunov I: Continuous-time systems,” *Transactions of the ASME. Series D, Journal of Basic Engineering,* vol. 82, pp. 371-393, 1960.

[17] R.E. Kalman and J.E. Bertram, “Control system analysis and design via the second method of Lyapunov II: Discrete-time systems,” *Transactions of the ASME. Series D, Journal of Basic Engineering,* vol. 82, pp. 94-400, 1960.