

Constrained Autonomous Precision Landing via Dual Quaternions and Model Predictive Control

Unsik Lee* and Mehran Mesbahi†

University of Washington, Seattle, WA 98195-2400

This paper addresses the problem of powered descent guidance and control for autonomous precision landing for next generation planetary missions. The precision landing algorithm aims to trace a fuel optimal trajectory while keeping geometrical constraints such as the line of sight to the target site. The design of an autonomous control algorithm managing such mission scenarios is challenging due to fact that critical geometrical constraints are coupled with the translational and rotational motions of the lander spacecraft, leading to a complex motion planning problem. We approach this problem within the model predictive control (MPC) framework by representing the dynamics of the rigid body in a uniform gravity field via a piece-wise affine system taking advantage of the unit dual quaternion parameterization. Such a parameterization in turn enables a 6-DOF motion planning in a unified framework while also admitting a quadratic cost on the required control commands in order to minimize propellant consumption. A novel feature of the proposed approach is the development of convex representable subsets in the configuration space in terms of unit dual quaternions. The stability and reachability issues of the corresponding piece-wise affine MPC approach are then discussed. Numerical simulations are presented to demonstrate the effectiveness of the proposed methodology for autonomous precision landing.

Nomenclature

\mathbb{R}	=	set of real numbers
\mathbb{R}^n	=	n -dimensional Euclidean space
$\mathbb{R}^{n \times m}$	=	set of $n \times m$ real matrices
\mathbb{S}_{++}^n	=	set of $n \times n$ positive definite matrices
$\text{SO}(3)$	=	special orthogonal group in dimension three
$\text{SE}(3)$	=	special Euclidean group in dimension three
\mathcal{S}^3	=	unit 3-sphere
\mathbf{I}_n	=	n -dimensional identity matrix
$\tilde{\mathbf{I}}_n$	=	n -dimensional anti-diagonal identity matrix
$\mathbf{0}_{n \times m}$	=	$n \times m$ zero matrix
$\det(\cdot)$	=	determinant of a given matrix
$[\mathbf{x}]_{\times}$	=	cross product operator associated with a vector $\mathbf{x} \in \mathbb{R}^3$
$\mathbf{x} \cdot \mathbf{y}$	=	dot product between vectors \mathbf{x} and \mathbf{y}
$\ \mathbf{x}\ $	=	Euclidean 2-norm of a vector or a quaternion x
$ x $	=	absolute value of a scalar x
$\text{diag}(x)$	=	diagonal matrix with the vector x on its diagonal
\mathbf{q}	=	unit quaternion as well as a pure quaternion
\mathbf{q}	=	vector part of the unit quaternion \mathbf{q}
q_o	=	scalar part of the unit quaternion \mathbf{q}
\mathbf{q}_I	=	identity unit quaternion
\mathbf{q}^*	=	conjugate unit quaternion
\otimes	=	product between two quaternions
$[\mathbf{q}]_{\otimes}$	=	quaternion product operator
$[\mathbf{q}]_{\otimes}^{\vee}$	=	transmuted quaternion product operator
\times	=	cross product between two quaternions
$[\mathbf{q}]_{\times}$	=	cross product operator
$[\mathbf{q}]_{\times}^{\vee}$	=	transmuted quaternion cross product operator
ϵ	=	dual unit for dual quaternions
$\tilde{\mathbf{q}}$	=	unit dual quaternion
$\tilde{\mathbf{q}}^*$	=	conjugate dual quaternion
\otimes	=	product between two dual quaternions
$[\tilde{\mathbf{q}}]_{\otimes}$	=	dual quaternion product operator
$[\tilde{\mathbf{q}}]_{\otimes}^{\vee}$	=	transmuted dual quaternion product operator
\times	=	vector product between two dual quaternions
$[\tilde{\mathbf{q}}]_{\times}$	=	dual quaternion cross product operator
$[\tilde{\mathbf{q}}]_{\times}^{\vee}$	=	transmuted dual quaternion cross product operator

$\ \tilde{\mathbf{q}}\ _{dq}$	=	norm of a dual quaternion
\mathcal{O}	=	ground fixed frame
\mathcal{B}	=	spacecraft body fixed frame
$\boldsymbol{\omega}_b$	=	angular velocity vector in pure quaternion
\mathbf{v}_b	=	velocity vector in pure quaternion
$\tilde{\boldsymbol{\omega}}$	=	velocity vector in dual quaternion form
\mathbf{r}_b	=	position vector in \mathcal{B} frame components
\mathbf{r}_o	=	position vector in \mathcal{O} frame components
θ	=	line of sight angle
ϕ	=	glide slope angle
ψ	=	general attitude angle
η	=	thrust direction angle

I. Introduction

The accuracy of planetary landing has been steadily improving over the past decades. The methods for achieving such improved levels of accuracy, however, have been predominately passive. For Mars landing missions, for example, the landing position error has largely improved through more accurate characterization of atmospheric uncertainties using data retrieved throughout previous missions. Other substantial factors leading to these position uncertainties at the landing site include uncertainties in navigating the spacecraft to the desired entry point in the atmosphere as well as measuring the vehicle aerodynamic coefficients [1]. In this direction, further improvements in planetary landing have been made through hypersonic entry guidance and enhancing navigational accuracy at the entry point with optical navigation [2]. These strategies have proved to be effective through two recent missions to Mars, namely, Phoenix and Mars Science Laboratory (MSL) missions in 2007 and 2011, respectively. Other sources of errors for landing algorithms such as map-tie error and wind drift are generally not predictable and require onboard real-time processing. Such uncertainties underscore the challenges for pinpoint/precision landing that is currently defined as landing within 100 m of the selected location and a retargeting capability [3]. For future planetary missions, precision landing will be of paramount importance as it will allow landing close to a scientifically rich target, potentially surrounded by hazardous terrain, or close to other pre-positioned surface assets, such as rovers, cargo, and fuel. For such a precision/pinpoint landing scenarios, the lander spacecraft should touch down softly and accurately close to the desired landing site while satisfying various state constraints in the powered descent stage. In this context, onboard guidance algorithm should also deal with environmental and operational uncertainties that generally degrade estimation accuracy and tracking performance. Such a requirement necessitates detecting and avoiding surface hazards such

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

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

as boulders and craters. Advances in sensor technology allows for accurate detection of these surface features. As such, the heritage flight system should enable a dynamic retargeting to avoid the hazards detected by the sensing systems [4]. Such a critical technology is referred to as the Autonomous Landing and Hazard Avoidance Technology (ALHAT) that is currently under development at the Jet Propulsion Laboratory (JPL) [5, 6, 7, 8]. One of the required technologies in ALHAT is that during the maneuver, the lander spacecraft should allow the sensors to see the landing site (when these sensors are not articulated) as well as including time and altitude margins for adjusting the trajectory based on the information gathered by the sensors [9]. In this venue, onboard vision sensors must maintain the line of sight with the surface targets during the descent phase. This type of constraint is also crucial for human missions since the crew is often responsible for making decisions during the last stage of landing in case of sensor failure or discovery of unexpected terrain hazards that the onboard sensors might have failed to detect. The line of sight vector pointing from the onboard sensors to the surface target relates to the coupled lander spacecraft's attitude and relative position with respect to the surface target. The topological complexity of the resulting lander configuration space leads to a number of challenges for autonomous landing.

In this paper, we utilize the unit dual quaternion parameterization to simultaneously represent the orientation and position of the lander. Such a parameterization has two main advantages: first, it provides a minimally globally nonsingular representation of $SE(3)$ for general motion of a rigid body without dependency on the underlying coordinate frame [10, 11]. During the last decade, this fact has resulted in the intermittent adoption of dual quaternions in the field of robot manipulators [12, 13] and computer graphics [14, 15]. Subsequently, dual quaternions have gained attention in aerospace engineering, e.g., rigid body control [16, 17, 18, 19], estimation [20, 21, 22], and inertial navigation [23], as an alternative minimal representation of the homogeneous transformation matrix. Second, as shown in this work, the dual quaternion parameterization enables an effective methodology for addressing motion planning in rotationally and translationally constrained configuration space. In fact, this parametrization facilitates convex representation of non-convex constraints typical in constrained landing missions, leading to the design of computationally tractable onboard algorithms for such complex missions; such a representation would not be feasible by adopting other representations, such as a combination of unit quaternions for attitude and three dimensional Euclidean vectors for position.

In the absence of state and control constraints, 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 optimal control literature. A closed form solution for a one dimensional (vertical) variation of this problem has been presented in [24, 25]. For Apollo missions, however, a quadratic polynomial (as a function of time) was employed for the synthesis of a feasible trajectory [26]. In general, there is no known closed form solution for the soft landing problem with state and control constraints, and as a result, various numerical approaches have been proposed for this problem over the past few decades. For example, in [27, 28, 29], guidance algorithms for the minimal fuel landing problem have been transformed to a two point boundary value problem and first-order necessary conditions for optimality have been derived while satisfying throttle and thrust angle control constraints.

With advances in computing technology, real-time onboard guidance and control has gained a lot of attention in many aerospace applications. In this context, [30, 31, 32, 33] have exploited the structure of certain classes of motion planning problems to design guidance algorithms via a convex optimization framework, with guaranteed convergence to the global optimum with a deterministic convergence criterion. In addition, convex optimization has been used to find collision free trajectories for formation reconfiguration problems [34, 35], and general robotic motion planning problems [36].

For a wider class of nonlinear (non-convex) motion planning problems, including controlling large angle rotational motions of a rigid body, an alternative real-time nonlinear model predictive control (NMPC) has been favored [37, 38]. In this setting, the nonlinear model of the plant is used to predict its future state over a predicted finite-horizon for the optimal control synthesis. Based on this prediction, a predefined objective function to penalize the state errors and con-

trol is then considered with respect to the next finite-horizon controls-only the first computed control action is then applied to the system. Such an NMPC approach, however, poses a number of computational challenges for real-time onboard implementation. Particularly for fast maneuvering aerospace applications, computational burden within the nonlinear MPC framework leads to a formidable barrier for real-time implementation. Following the general observation that (linear) MPC is particularly effective due to the fact that it involves solving tractable convex quadratic programs [39, 40], piece-wise affine (PWA) approximation has emerged as an alternative for replacing the nonlinear system model in NMPC [41, 42, 43, 44, 45]. In fact, a number of control synthesis problems for nonlinear plants have been successfully approached using such PWA approximations [46, 47]. Specifically, [48, 49] propose the sampled-data PWA system, where the switching action of the discrete state is determined at each sampling time according to a condition on the continuous state.

In this paper, we present a PWA-MPC algorithm for the powered descent guidance within the piece-wise affine (PWA) framework that effectively approximates the nonlinear rotational and translational rigid body dynamics in a uniform gravity field. Moreover, the rotationally and translationally constrained zones such as a line of sight and a glide slope constraints are represented using unit dual quaternions, enabling a 6-DOF motion prediction in an unified fashion while also admitting a quadratic cost on the required control commands in order to minimize the propellant consumption. Such constraints are imposed to ensure that the lander's trajectory does not violate the constraints during the powered descent phase of the mission. A novel feature of proposed approach is the development of convex representable subsets in the rotational and translational configuration space in terms of unit dual quaternions that correspond to general rigid body motions satisfying predefined geometrical constraints. The stability and reachability analysis for the PWA-MPC are subsequently discussed. Numerical simulations are then conducted to demonstrate the effectiveness of the proposed algorithm for autonomous precision landing.

The paper is organized as follows. In §II, background on unit quaternions and unit dual quaternions are discussed. In §III, the dual quaternion based rigid body dynamics are derived. In §IV, we define and parameterize four classes of rotationally and translationally constrained zones of importance for autonomous planetary landing; these zones are then convexified and incorporated into the proposed PWA-MPC framework. In §V, the problem of autonomous landing with constraints is then examined, followed by reformulation of the problem in terms of PWA-MPC in §VI. We provide a simulation example in §VII, followed by conclusions of this work in §VIII.

II. Preliminaries

In this section, we provide a brief background on unit quaternions and unit dual quaternions; the reader is also referred to [19]. More information on quaternions and dual quaternions can be found in [16, 17, 18, 21, 23].

A. Unit Quaternions

The attitude of a rigid body, describing the relative orientation between a reference frame and the body-fixed frame, evolves on the special orthogonal group $SO(3)$. The unit quaternion is a minimal parameterization of $SO(3)$ free from singularities. As such, spacecraft large angle control algorithms often favor such a parameterization; see [50, 51]. In this paper, we adopt the vector notation for unit quaternions as $\mathbf{q} = [n_x \sin \frac{\theta}{2}, n_y \sin \frac{\theta}{2}, n_z \sin \frac{\theta}{2}, \cos \frac{\theta}{2}]^T$, where n_x, n_y and n_z denote, respectively, the three perpendicular components of the rotation axis \mathbf{n} , and θ denotes the rotation angle around this axis. Note that from this definition, we have $\mathbf{q} \in S^3$, where S^3 denotes the 3-dimensional unit sphere. Unit quaternions are closed with respect to the following operations:

Quaternion Product: The product of two quaternions is represented using the vector notation as

$$\mathbf{q} \otimes \mathbf{p} = \begin{bmatrix} q_0 \mathbf{p} + p_0 \mathbf{q} + \mathbf{q} \times \mathbf{p} \\ q_0 p_0 - \mathbf{q}^T \mathbf{p} \end{bmatrix}, \quad (1a)$$

where $\mathbf{q} = [\mathbf{q}^T, q_0]^T$ and $\mathbf{p} = [\mathbf{p}^T, p_0]^T$. Another quaternion operation is the “unit quaternion conjugation” defined as $\mathbf{q}^* = [-\mathbf{q}^T, q_0]^T$ in the vector notation, which facilitates the judicious definition of the “attitude difference/error” of \mathbf{p} with respect to \mathbf{q} as $\mathbf{q}_e = \mathbf{p} \otimes \mathbf{q}^*$. Note that the identity quaternion is expressed by $\mathbf{q}_I = [0 \ 0 \ 0 \ 1]^T$. Quaternion product is analogous in many ways to vector cross products. Thus, the quaternion operation can be expressed as the product of a skew symmetric matrix and a quaternion,

$$\mathbf{q} \otimes \mathbf{p} \stackrel{\text{def}}{=} [\mathbf{q}]_{\otimes} \mathbf{p} \stackrel{\text{def}}{=} [\mathbf{p}]_{\otimes}^{\vee} \mathbf{q}, \quad (1b)$$

where $[\mathbf{q}]_{\otimes}$ and $[\mathbf{p}]_{\otimes}^{\vee}$ denote 4×4 skew symmetric matrices related to “ \mathbf{q} ” and “ \mathbf{p} ”, respectively.

Quaternion Cross Products: The quaternion product Eq. (1a) can be relaxed to the newly defined *quaternion cross product* as

$$\mathbf{q} \times \mathbf{p} = \begin{bmatrix} q_0 \mathbf{p} + p_0 \mathbf{q} + \mathbf{q} \times \mathbf{p} \\ 0 \end{bmatrix}, \quad (2a)$$

where \mathbf{q} and \mathbf{p} denote quaternions. This quaternion cross operation can also be expressed using matrix multiplication,

$$\mathbf{q} \times \mathbf{p} \stackrel{\text{def}}{=} [\mathbf{q}]_{\times} \mathbf{p} \stackrel{\text{def}}{=} [\mathbf{p}]_{\times}^{\vee} \mathbf{q}, \quad (2b)$$

where $[\cdot]_{\times}$ denotes a 4×4 linear operator matrix corresponding to the *quaternion cross product*. Note that this operation is only closed for pure quaternions. 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:

$$(\mathbf{a} \otimes \mathbf{b})^* = \mathbf{b}^* \otimes \mathbf{a}^* \quad (3a)$$

$$(\gamma \mathbf{a}) \otimes \mathbf{b} = \mathbf{a} \otimes (\gamma \mathbf{b}) = \gamma (\mathbf{a} \otimes \mathbf{b}) \quad (3b)$$

$$\mathbf{a} \otimes (\mathbf{b} + \mathbf{c}) = \mathbf{a} \otimes \mathbf{b} + \mathbf{a} \otimes \mathbf{c} \quad (3c)$$

$$\mathbf{a} \otimes (\mathbf{b} \otimes \mathbf{c}) = (\mathbf{a} \otimes \mathbf{b}) \otimes \mathbf{c} \quad (3d)$$

$$\mathbf{a}^T (\mathbf{b} \otimes \mathbf{c}) = \mathbf{c}^T (\mathbf{b}^* \otimes \mathbf{a}) = \mathbf{b}^T (\mathbf{a} \otimes \mathbf{c}^*), \quad (3e)$$

where $\gamma \in \mathbb{R}$. Another interesting algebraic property applicable to unit quaternions is inspired by the Binet-Cauchy identity [52].

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

$$(\mathbf{r} \otimes \mathbf{q})^T (\mathbf{y} \otimes \mathbf{q}) = (\mathbf{q} \otimes \mathbf{r})^T (\mathbf{q} \otimes \mathbf{y}) = \mathbf{r}^T \mathbf{y}, \quad (4)$$

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

Proof. See [19] for the proof. \square

B. Dual Quaternions

The concept of “*dual quaternion*” is due to Clifford [10] who aimed to encode an additional piece of information, namely, relative position, 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 $\tilde{\mathbf{q}} = \mathbf{q}_1 + \epsilon \mathbf{q}_2$, where ϵ denotes the “dual unit” satisfying the “unconventional” properties $\epsilon^2 = 0$ and $\epsilon \neq 0$. 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 $\tilde{\mathbf{q}} = \mathbf{q}_1 + \epsilon \mathbf{q}_2$, or $\tilde{\mathbf{q}} = [\mathbf{q}_1^T \ \mathbf{q}_2^T]^T_{8 \times 1}$, where \mathbf{q}_1 and \mathbf{q}_2 are quaternions. The dual quaternion is closed under (appropriately defined) product.

Dual Quaternion Products: Using the dual unit definition, the corresponding product operation is expressed as $\tilde{\mathbf{q}} \otimes \tilde{\mathbf{p}} = \mathbf{q}_1 \otimes \mathbf{p}_1 + \epsilon (\mathbf{q}_1 \otimes \mathbf{p}_2 + \mathbf{q}_2 \otimes \mathbf{p}_1)$. Note that the dual quaternion product, denoted by \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. (1b). Specifically,

$$\tilde{\mathbf{q}} \otimes \tilde{\mathbf{p}} \stackrel{\text{def}}{=} [\tilde{\mathbf{q}}]_{\otimes} \tilde{\mathbf{p}} \stackrel{\text{def}}{=} [\tilde{\mathbf{p}}]_{\otimes}^{\vee} \tilde{\mathbf{q}}, \quad (5a)$$

where $[\cdot]_{\otimes}$ and $[\cdot]_{\otimes}^{\vee}$ denote an 8×8 linear operator matrix corresponding to the dual quaternion product.

Dual Quaternion Cross Products: Similarly, the dual quaternion product can be relaxed to the dual quaternion cross product as $\tilde{\mathbf{q}} \times \tilde{\mathbf{p}} = \mathbf{q}_1 \times \mathbf{p}_1 + \epsilon (\mathbf{q}_1 \times \mathbf{p}_2 + \mathbf{q}_2 \times \mathbf{p}_1)$, where $\tilde{\mathbf{q}}$ and $\tilde{\mathbf{p}}$ denote dual quaternions. This dual quaternion cross product operation can also be expressed in terms of matrix multiplication as,

$$\tilde{\mathbf{q}} \times \tilde{\mathbf{p}} \stackrel{\text{def}}{=} [\tilde{\mathbf{q}}]_{\times} \tilde{\mathbf{p}} \stackrel{\text{def}}{=} [\tilde{\mathbf{p}}]_{\times}^{\vee} \tilde{\mathbf{q}}, \quad (6a)$$

where $[\cdot]_{\times}$ and $[\cdot]_{\times}^{\vee}$ denote an 8×8 linear operator (matrix) corresponding to the *dual quaternion cross product*. Note that this operation is only closed for the pure dual quaternion or *Plücker coordinates* [53].

When the dual quaternion has a nonzero real part, its inverse can be obtained as $\tilde{\mathbf{q}}^{-1} = \|\tilde{\mathbf{q}}\|^{-1} \tilde{\mathbf{q}}^*$, where $\tilde{\mathbf{q}}^*$ denotes the dual quaternion conjugate defined as $\tilde{\mathbf{q}}^* = [\mathbf{q}_1^{*T} \ \mathbf{q}_2^{*T}]^T$. 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 $\|\tilde{\mathbf{q}}\|_{dq} = [\mathbf{q}_I^T \ 0_{4 \times 1}^T]^T$, where \mathbf{q}_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 SE(3)

For the set of *unit dual quaternions*, which is a subset of dual quaternions, the first part \mathbf{q}_1 is restricted to be a unit norm vector $\|\mathbf{q}_1\|_2 = 1$. Suppose that the relationship between the fixed frame \mathcal{O} and the body

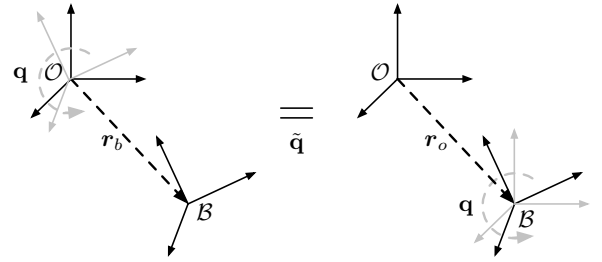


Figure 1. The geometric difference between the frame \mathcal{B} and \mathcal{O} can be expressed in two ways: a rotation \mathbf{q} followed by a translation \mathbf{r}_b [left panel] and a translation \mathbf{r}_o followed by a rotation \mathbf{q} [right panel], both represented by the unit dual quaternion $\tilde{\mathbf{q}}$

frame \mathcal{B} is as shown in Figure 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, $\text{SE}(3)$ is the set of all rigid body transformations in the three dimensional space,

$$\text{SE}(3) = \left\{ T \in \mathbb{R}^{4 \times 4} \mid T = \begin{bmatrix} R & \mathbf{r}_o \\ 0 & 1 \end{bmatrix}, R \in \text{SO}(3), \mathbf{r}_o \in \mathbb{R}^3 \right\},$$

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

The geometric difference between the frame \mathcal{B} with respect to the fixed frame \mathcal{O} can be expressed in two ways: a translation \mathbf{r}_o (represented in frame \mathcal{O}) followed by a rotation \mathbf{q}_1 and a rotation \mathbf{q}_1 followed by a translation \mathbf{r}_b (represented in frame \mathcal{B}); see Figure 1. This observation can be represented in the form of *unit dual quaternions* as

$$\tilde{\mathbf{q}} \stackrel{\text{def}}{=} \begin{bmatrix} \mathbf{q} \\ \frac{1}{2} \mathbf{r}_o \otimes \mathbf{q} \end{bmatrix} = \begin{bmatrix} \mathbf{q} \\ \frac{1}{2} \mathbf{q} \otimes \mathbf{r}_b \end{bmatrix}, \quad (7)$$

where \mathbf{r}_o and \mathbf{r}_b represent the translation vector \mathbf{r} with respect to the frames \mathcal{O} and \mathcal{B} , respectively. Note that Eq. (7) satisfies the dual quaternion norm constraint. Converting between \mathbf{r}_o and \mathbf{r}_b is governed by the quaternion rotation operators as follows,

$$\mathbf{r}_o = \mathbf{q} \otimes \mathbf{r}_b \otimes \mathbf{q}^* \quad \text{and} \quad \mathbf{r}_b = \mathbf{q}^* \otimes \mathbf{r}_o \otimes \mathbf{q}. \quad (8a)$$

D. Rigid Body Dynamics in Unit Dual Quaternions

The unit dual quaternion kinematic equation is given with a dual velocity $\tilde{\mathbf{q}}$ as [11]

$$\dot{\tilde{\mathbf{q}}} = \frac{1}{2} \tilde{\mathbf{q}} \otimes \tilde{\boldsymbol{\omega}} = \frac{1}{2} [\tilde{\boldsymbol{\omega}}]_{\otimes}^{\vee} \tilde{\mathbf{q}} \quad \text{with} \quad \tilde{\boldsymbol{\omega}} = \begin{bmatrix} \boldsymbol{\omega}_b^T & \mathbf{v}_b^T \end{bmatrix}_{8 \times 1}^T, \quad (9)$$

where $\boldsymbol{\omega}_b = [\boldsymbol{\omega}_b^T \ 0]^T$ denotes the angular velocity of the rigid body and $\mathbf{v}_b = [\mathbf{v}_b^T \ 0]^T$ denotes the translational velocity defined by $\mathbf{v}_b = \dot{\mathbf{r}}_b + \boldsymbol{\omega}_b \times \mathbf{r}_b$. 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 [54]:

$$\mathbf{F} = \frac{d}{dt}(m\mathbf{v}) = m\dot{\mathbf{v}}_b + \boldsymbol{\omega}_b \times m\mathbf{v}_b, \quad (10a)$$

$$\mathbf{T} = \frac{d}{dt}(J\boldsymbol{\omega}) = J\dot{\boldsymbol{\omega}}_b + \boldsymbol{\omega}_b \times J\boldsymbol{\omega}_b, \quad (10b)$$

where $d(\cdot)/dt$ denotes the time derivative represented in the body frame \mathcal{B} , $m \in \mathbb{R}$ denotes the mass of the rigid body, $J \in \mathbb{S}_{++}^3$ denotes its inertia matrix along the body frame, and $\boldsymbol{\omega}_b, \mathbf{v}_b \in \mathbb{R}^3$ denote, respectively, the angular and translational velocities of the rigid body represented in the body frame. Moreover, \mathbf{F} and \mathbf{T} denote the external translational force and torque acting on the rigid body, written in the body frame, respectively. Note that in order to focus on the reachability 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 to derive a rigid body dynamics in dual quaternions.

Proposition 1. *The rotational and translational dynamics Eqs. (10a)-(10b) can be represented in dual quaternions as*

$$\mathbf{J}\dot{\tilde{\mathbf{q}}} + \tilde{\boldsymbol{\omega}} \times \mathbf{J}\tilde{\mathbf{q}} = \tilde{\mathbf{F}} \quad (11a)$$

where

$$\mathbf{J} = \left[\begin{array}{cc|cc} 0_{3 \times 3} & 0 & m\mathbf{I}_3 & 0 \\ 0 & 0 & 0 & 1 \\ \hline J & 0 & 0_{3 \times 3} & 0 \\ 0 & 1 & 0 & 0 \end{array} \right]_{8 \times 8} \quad \text{and} \quad \tilde{\mathbf{F}} = \begin{bmatrix} F \\ 0 \\ T \\ 0 \end{bmatrix}. \quad (11b)$$

Moreover, $\mathbf{J} \in \mathbb{R}^{8 \times 8}$ forms an invertible matrix.

Proof. See [19] for the proof. \square

Eqs. (9) and (11a) constitute the kinematics and dynamics of the rigid body in terms of dual quaternions, considering *coupled* translational and rotational motion. Next, consider the gravitational effect on the dynamics. The gravitational acceleration is assumed to be constant as observed in the fixed frame \mathcal{O} . Then, the gravitation force is given as

$$\mathbf{g}_o = [0 \ 0 \ mg \ 0]^T, \quad (12a)$$

where m denotes the mass at time of measurement and g denotes the gravitational acceleration. Using the quaternion rotation operator Eq. (8a), the gravitational force observed in the body frame \mathcal{B} is given as $\mathbf{g}_b = m(\tilde{\mathbf{q}}^* \otimes \mathbf{g}_o \otimes \tilde{\mathbf{q}})$. This equation can be represented in terms of dual quaternions as

$$\tilde{\mathbf{g}}_b = m\tilde{\mathbf{I}}_8(\tilde{\mathbf{q}}^* \otimes \tilde{\mathbf{I}}_8\tilde{\mathbf{g}}_o \otimes \tilde{\mathbf{q}}) = m\tilde{\mathbf{I}}_8[\tilde{\mathbf{q}}^* \otimes \tilde{\mathbf{I}}_8\tilde{\mathbf{g}}_o]_{\otimes} \tilde{\mathbf{q}} \quad (12b)$$

with

$$\tilde{\mathbf{g}}_o = [\mathbf{g}_o^T \ 0_{1 \times 4}]^T \quad \text{and} \quad \tilde{\mathbf{I}}_8 = \begin{bmatrix} 0_{4 \times 4} & \mathbf{I}_4 \\ \mathbf{I}_4 & 0_{4 \times 4} \end{bmatrix}. \quad (12c)$$

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

$$\mathbf{J}\dot{\tilde{\mathbf{q}}} + \tilde{\boldsymbol{\omega}} \times \mathbf{J}\tilde{\mathbf{q}} = \tilde{\mathbf{F}} + \tilde{\mathbf{g}}_b, \quad (12d)$$

and the dual quaternion kinematics Eq. (9). Note that in this set of equations, all vector components are represented in the body fixed frame.

III. Rotationally and Translationally Constrained Zones

Some of the challenging algorithmic aspects of the powered descent guidance problem stems from the fact that the translational and rotational dynamics of the spacecraft are often taken into account independently in their respective configuration spaces. 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 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, using unit dual quaternions as a single configuration space parameterizing the orientation and position of a rigid-body, we develop the machinery for a unified representation of rotationally and translationally constrained zones. Since this framework was originally inspired by the powered descent guidance problem for a Mars lander, let us formulate the unit dual quaternion-based constrained zones in the context of this application; Figure 2 depicts the overall landing scenario.

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 Figure 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 $-\mathbf{r}_b$ stays within an angle θ about the line of sight vector \mathbf{y}_b in the frame \mathcal{B} via the inequality,

$$-\mathbf{r}_b \cdot \mathbf{y}_b \geq \|\mathbf{r}_b\| \cos \theta, \quad (13)$$

where \mathbf{r}_b denotes a translation vector represented in the frame \mathcal{B} . Note that Eq. (13) holds for $-\pi \leq \theta \leq \pi$. One can show that the left hand-side of the above equation can be expressed as the quadratic form,

$$\begin{aligned} \mathbf{r}_b \cdot \mathbf{y}_b &= (\mathbf{q}_1 \otimes \mathbf{r}_b)^T (\mathbf{q}_1 \otimes \mathbf{y}_b) \\ &= (\mathbf{q}_1 \otimes \mathbf{r}_b)^T [\mathbf{y}_b]_{\otimes}^{\vee} \mathbf{q}_1 \\ &= \begin{bmatrix} \mathbf{q}_1 \\ \frac{1}{2} \mathbf{q}_1 \otimes \mathbf{r}_b \end{bmatrix}^T \begin{bmatrix} 0_{4 \times 4} & [\mathbf{y}_b]_{\otimes}^{\vee T} \\ [\mathbf{y}_b]_{\otimes}^{\vee} & 0_{4 \times 4} \end{bmatrix} \begin{bmatrix} \mathbf{q}_1 \\ \frac{1}{2} \mathbf{q}_1 \otimes \mathbf{r}_b \end{bmatrix} \\ &= \tilde{\mathbf{q}}^T M_H \tilde{\mathbf{q}}, \end{aligned} \quad (14)$$

where $\tilde{\mathbf{q}}$ denotes a unit dual quaternion and the quaternion properties Eqs. (1b) and (4) have been applied. We note that the matrix M_H in Eq. (14) is generally an indefinite symmetric matrix. In addition, motivated by Eq. (4), we can find the following useful identities,

$$\tilde{\mathbf{q}}^T \tilde{\mathbf{q}} = \|\tilde{\mathbf{q}}\|^2 = 1 + \frac{1}{4} \|\mathbf{r}_b\|^2, \quad (15a)$$

$$\tilde{\mathbf{q}}^T E_u \tilde{\mathbf{q}} = 1, \quad 4 \tilde{\mathbf{q}}^T E_d \tilde{\mathbf{q}} = \|\mathbf{r}_b\|^2, \quad \text{and} \quad (15b)$$

$$\|\mathbf{r}_b\| = \|2 E_d \tilde{\mathbf{q}}\| \quad (15c)$$

where

$$E_u = \begin{bmatrix} \mathbf{I}_4 & 0_{4 \times 4} \\ 0_{4 \times 4} & 0_{4 \times 4} \end{bmatrix}, \quad E_d = \begin{bmatrix} 0_{4 \times 4} & 0_{4 \times 4} \\ 0_{4 \times 4} & \mathbf{I}_4 \end{bmatrix}. \quad (15d)$$

Rewriting Eq. (13) with the above identities in mind, we obtain the parameterization of the zone due to the line of sight constraint as

$$f_1(\tilde{\mathbf{q}}) = \tilde{\mathbf{q}}^T M_H \tilde{\mathbf{q}} + \|2 E_d \tilde{\mathbf{q}}\| \cos \theta \leq 0. \quad (16)$$

We note that in Eq. (15a) the inner product of two unit dual quaternions $\tilde{\mathbf{q}}^T \tilde{\mathbf{q}}$ is uniquely defined by the norm of a vector $\mathbf{r}_b \in \mathbb{R}^3$. Now consider a closed subset of \mathbb{R}^3 as

$$\|\mathbf{r}_b\| \leq \delta, \quad (17)$$

and let

$$\tilde{\mathbf{q}}^T \tilde{\mathbf{q}} \leq 1 + \frac{1}{4} \delta^2. \quad (18)$$

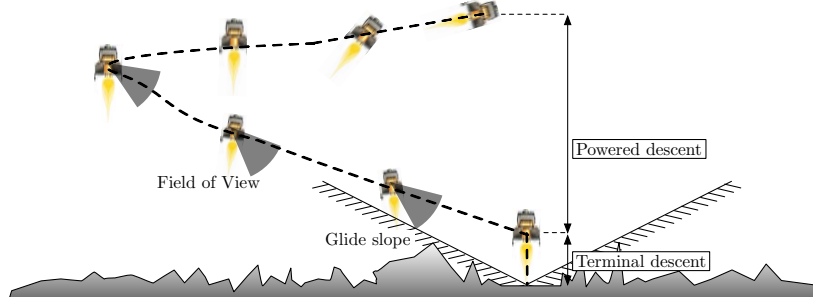


Figure 2. Powered descent guidance scenario in the presence of line of sight and glide slope constraints

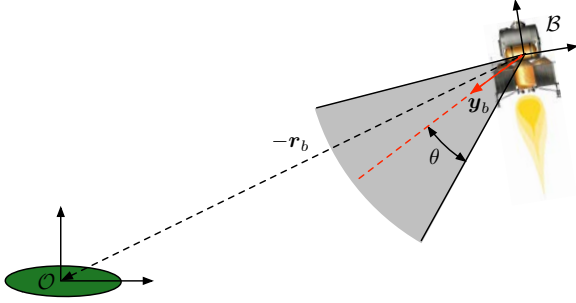


Figure 3. Illustration of the line of sight constraint; this constraint is defined as a cone around the fixed boresight vector y_b in the body frame.

Then, one can show that the left hand-side of Eq. (16) is convex over the defined closed subset. Let us expand on this observation in the following proposition.

Proposition 2. Consider the set defined through the quadratic function $f : \tilde{\mathbf{q}} \rightarrow \mathbb{R}$ via

$$f_1(\tilde{\mathbf{q}}) = \tilde{\mathbf{q}}^T M_H \tilde{\mathbf{q}} + \|2 E_d \tilde{\mathbf{q}}\| \cos \theta \leq 0, \quad (19a)$$

with

$$\text{dom } f_1 = \left\{ \tilde{\mathbf{q}} \in (\mathbb{S}^3 \times \mathbb{R}^3) \mid \tilde{\mathbf{q}}^T \tilde{\mathbf{q}} \leq 1 + \frac{1}{4} \delta^2 \right\}, \quad (19b)$$

where $\delta \in \mathbb{R}$, the matrices M_H and E_d are defined as in Eqs. (14) and (15d), respectively. Then f_1 is convex regardless of θ .

Proof. From the property of unit dual quaternions, Eq. (15b), we have

$$\delta^2 (\tilde{\mathbf{q}}^T E_u \tilde{\mathbf{q}}) - \delta^2 = 0. \quad (20a)$$

Using this equality in Eq. (19a) yields

$$f_1 = \tilde{\mathbf{q}}^T M_H \tilde{\mathbf{q}} + \|2 E_d \tilde{\mathbf{q}}\| \cos \theta + \delta^2 (\tilde{\mathbf{q}}^T E_u \tilde{\mathbf{q}}) - \delta^2. \quad (20b)$$

Note that f_1 is twice differentiable and

$$\begin{aligned} \tilde{\mathbf{q}}^T \left(\frac{d^2}{d \tilde{\mathbf{q}}^2} \|2 E_d \tilde{\mathbf{q}}\| \right) \tilde{\mathbf{q}} &= \frac{4 \tilde{\mathbf{q}}^T E_d \tilde{\mathbf{q}}}{\|2 E_d \tilde{\mathbf{q}}\|} - \frac{16 \tilde{\mathbf{q}}^T E_d \tilde{\mathbf{q}} \tilde{\mathbf{q}}^T E_d^T \tilde{\mathbf{q}}}{\|2 E_d \tilde{\mathbf{q}}\|^3} \\ &= \frac{4 \tilde{\mathbf{q}}^T E_d \tilde{\mathbf{q}}}{\|2 E_d \tilde{\mathbf{q}}\|} - \frac{4 \tilde{\mathbf{q}}^T E_d \tilde{\mathbf{q}}}{\|2 E_d \tilde{\mathbf{q}}\|} = 0. \end{aligned} \quad (20c)$$

Thus the quadratic form of the Hessian of f_1 has the form

$$\begin{aligned} \tilde{\mathbf{q}}^T (\nabla^2 f_1) \tilde{\mathbf{q}} &= 2(\tilde{\mathbf{q}}^T M_H \tilde{\mathbf{q}}) + 2\delta^2 (\tilde{\mathbf{q}}^T E_u \tilde{\mathbf{q}}) \\ &= 2(\mathbf{r}_b \cdot \mathbf{y}_b + \delta^2). \end{aligned} \quad (20d)$$

We now note that $\mathbf{r}_b \cdot \mathbf{y}_b + \delta^2 \geq 0$ on $\text{dom } f_1$. Since \mathbf{y}_b is a unit vector, we observe that

$$-\|\mathbf{r}_b\|^2 \leq \mathbf{r}_b \cdot \mathbf{y}_b \leq \|\mathbf{r}_b\|^2. \quad (20e)$$

Due to Eq. (17), we thereby conclude that $\nabla^2 f_1(\tilde{\mathbf{q}})$ is positive semidefinite and thus, f_1 is convex on $\text{dom } f$. \square

Remark 1. One can show an analogous statement for the case where the line of sight boresight vector \mathbf{y}_b has to stay *outside of* θ around the direction to the target \mathbf{r}_b as

$$-\mathbf{r}_b \cdot \mathbf{y}_b \leq \|\mathbf{r}_b\| \cos \theta. \quad (21)$$

The corresponding constraint can also be represented as a convex constraint on the same domain. This can be achieved by observing that in Eq. (19b), $1 \leq \tilde{\mathbf{q}}^T \tilde{\mathbf{q}} \leq 1 + \delta^2/4$, which allows the Hessian to remain positive semidefinite on this domain.

B. Glide Slope Constraints

The glide slope constraint is defined as a cone around the fixed vector \mathbf{z}_o that lies in the frame \mathcal{O} ; see the illustration in Figure 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

$$\mathbf{r}_o \cdot \mathbf{z}_o \geq \|\mathbf{r}_o\| \cos \phi, \quad (22)$$

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

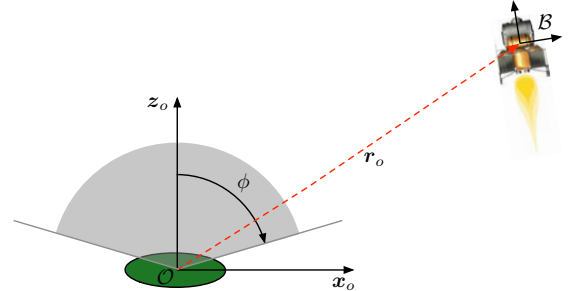


Figure 4. Illustration of the glide slope constraint; the constrained zone is defined as a cone around the ground fixed vector \mathbf{z}_o and angle ϕ .

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

$$f_2(\tilde{\mathbf{q}}) = -\tilde{\mathbf{q}}^T M_G \tilde{\mathbf{q}} + \|2 E_d \tilde{\mathbf{q}}\| \cos \phi \leq 0, \quad (23a)$$

where

$$M_G = \begin{bmatrix} 0_{4 \times 4} & [\mathbf{z}_o]_{\otimes}^T \\ [\mathbf{z}_o]_{\otimes} & 0_{4 \times 4} \end{bmatrix}, \quad E_d = \begin{bmatrix} 0_{4 \times 4} & 0_{4 \times 4} \\ 0_{4 \times 4} & \mathbf{I}_4 \end{bmatrix} \quad (23b)$$

with $\mathbf{z}_o = [\mathbf{z}_o^T \ 0]^T = [0 \ 0 \ 1 \ 0]^T$. Moreover, $f_2 : \tilde{\mathbf{q}} \rightarrow \mathbb{R}$ is convex on $\text{dom } f_2 = \{\tilde{\mathbf{q}} \in (\mathbb{S}^3 \times \mathbb{R}^3) \mid \tilde{\mathbf{q}}^T \tilde{\mathbf{q}} \leq 1 + \frac{1}{4} \delta^2\}$.

Proof. The left hand-side of Eq. (22) is rewritten as follows

$$\begin{aligned}
 \mathbf{r}_o \cdot \mathbf{z}_o &= (\mathbf{r}_o \otimes \mathbf{q}_1)^T (\mathbf{z}_o \otimes \mathbf{q}_1) \\
 &= (\mathbf{r}_o \otimes \mathbf{q}_1)^T [\mathbf{z}_o]_{\otimes} \mathbf{q}_1 \\
 &= \begin{bmatrix} \mathbf{q}_1 \\ \frac{1}{2} \mathbf{r}_o \otimes \mathbf{q}_1 \end{bmatrix}^T \begin{bmatrix} 0_{4 \times 4} & [\mathbf{z}_o]^T \\ [\mathbf{z}_o]_{\otimes} & 0_{4 \times 4} \end{bmatrix} \begin{bmatrix} \mathbf{q}_1 \\ \frac{1}{2} \mathbf{r}_o \otimes \mathbf{q}_1 \end{bmatrix} \\
 &= \tilde{\mathbf{q}}^T M_G \tilde{\mathbf{q}}, \tag{24a}
 \end{aligned}$$

where M_G denotes an indefinite symmetric matrix. Thus we obtain Eq. (23a). The rest of the proof is analogous to the proof in Proposition 2. Over the set $\text{dom } f_2$, we can rewrite Eq. (23a) as

$$f_2(\tilde{\mathbf{q}}) = -\tilde{\mathbf{q}}^T M_G \tilde{\mathbf{q}} + \|2 E_d \tilde{\mathbf{q}}\| \cos \phi + \delta^2 (\tilde{\mathbf{q}}^T E_u \tilde{\mathbf{q}}) - \delta^2. \tag{24b}$$

Moreover, the quadratic form of the Hessian on f_2 yields

$$\tilde{\mathbf{q}}^T (\nabla^2 f_2) \tilde{\mathbf{q}} = 2(-\mathbf{r}_o \cdot \mathbf{y}_o + \delta^2). \tag{24c}$$

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

$$-\mathbf{r}_o \cdot \mathbf{y}_o + \delta^2 \geq 0, \tag{24d}$$

on $\text{dom } f_2$ from Eq. (17). Note that \mathbf{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 $\text{dom } f_2$, concluding the proof. \square

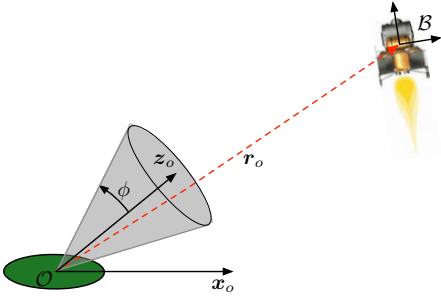


Figure 5. Illustration of the general lower and upper bounded glide slope constraint; the constrained zone is defined as a cone around the ground fixed vector \mathbf{z}_o and angle ϕ .

Remark 2. The aforementioned constraint forms an analogous formulation in terms of unit dual quaternions. This is due to the fact that 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 by Eq. (7).

Remark 3. For a landing application, we typically have a lower bounded glide slope constraint with $\mathbf{z}_o = [0, 0, 1]$. This is because, for the planetary landing, most hazards are found on the ground. Such a constraint can easily adapt to an upper and lower bounded glide slope constraint; see Figure 5.

C. General Spacecraft Attitude Constraints

The spacecraft rotational maneuvers are generally restricted to a certain angle about the (ground) fixed axes. For example, there may be a radar sensor onboard that has to be pointed to the ground to detect the spacecraft's altitude. The case in point is the sky crane system used in the Mars Science Laboratory (MSL) mission by NASA that was required to keep changes in the angle ψ minimal to detect the precise distance above the ground while maneuvering. Such a rotational constraint can be captured by the angle between the ground fixed vector \mathbf{z}_o and the body-fixed vector \mathbf{z}_b ; see Figure 6. In this formulation, we can now define an inequality representing the constraint as,

$$\mathbf{z}_o \cdot [\mathbf{z}_b]_{\mathcal{O}} \geq \cos \psi, \tag{25}$$

where \mathbf{z}_o denotes a unitized ground fixed vector and $[\mathbf{z}_b]_{\mathcal{O}}$ denotes a unitized body-fixed vector \mathbf{z}_b represented in the inertial frame \mathcal{O} .

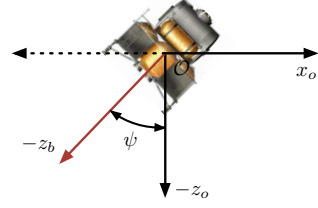


Figure 6. General spacecraft attitude constraints

Proposition 4. The aforementioned general spacecraft attitude constraint can be formulated in terms of the unit dual quaternion $\tilde{\mathbf{q}}$ as

$$f_3(\tilde{\mathbf{q}}) = \tilde{\mathbf{q}}^T M_A \tilde{\mathbf{q}} + \cos \psi \leq 0, \tag{26a}$$

where

$$M_A = \begin{bmatrix} [\mathbf{z}_o]_{\otimes} [\mathbf{z}_b]_{\otimes}^{\vee} & 0_{4 \times 4} \\ 0_{4 \times 4} & 0_{4 \times 4} \end{bmatrix}_{8 \times 8}, \tag{26b}$$

and $\mathbf{z}_o = [\mathbf{z}_o^T \ 0]^T$ and $\mathbf{z}_b = [\mathbf{z}_b^T \ 0]^T$. Moreover, in such a setup, $f_3 : \tilde{\mathbf{q}} \rightarrow \mathbb{R}$ on $\text{dom } f_3 = \{\tilde{\mathbf{q}} \in (\mathcal{S}^3 \times \mathbb{R}^3) \mid \tilde{\mathbf{q}}^T \tilde{\mathbf{q}} \leq 1 + \frac{1}{4} \delta^2\}$ is convex.

Proof. First note that \mathbf{z}_b is observed in the body frame \mathcal{B} . By Eq. (8a), it can be represented with the components of the inertial frame \mathcal{O} . Then according to the quaternion rotation operator Eq. (8a), it follows that

$$\mathbf{z}_o^T (\mathbf{q}_1 \otimes \mathbf{z}_b \otimes \mathbf{q}_1^*) \geq \cos \psi. \tag{27a}$$

Using the Unit Quaternion Triple Identity presented in §II, we have

$$(\mathbf{z}_o \otimes \mathbf{q}_1)^T (\mathbf{q}_1 \otimes \mathbf{z}_b \otimes \mathbf{q}_1^*) \geq \cos \psi \tag{27b}$$

$$(\mathbf{z}_o \otimes \mathbf{q}_1)^T (\mathbf{q}_1 \otimes \mathbf{z}_b) \geq \cos \psi \tag{27c}$$

$$\mathbf{q}_1^T ((\mathbf{z}_o \otimes \mathbf{q}_1) \otimes \mathbf{z}_b^*) \geq \cos \psi \tag{27d}$$

$$\mathbf{q}_1^T (\mathbf{z}_o \otimes (\mathbf{q}_1 \otimes \mathbf{z}_b)) \leq -\cos \psi \tag{27e}$$

$$\mathbf{q}_1^T [\mathbf{z}_o]_{\otimes} [\mathbf{z}_b]_{\otimes}^{\vee} \mathbf{q}_1 \leq -\cos \psi \tag{27f}$$

$$\tilde{\mathbf{q}}^T M_A \tilde{\mathbf{q}} + \cos \psi \leq 0, \tag{27g}$$

where $\tilde{\mathbf{q}}$ denotes a unit dual quaternion and M_A is defined by

$$M_A = \begin{bmatrix} [\mathbf{z}_o]_{\otimes} [\mathbf{z}_b]_{\otimes}^{\vee} & 0_{4 \times 4} \\ 0_{4 \times 4} & 0_{4 \times 4} \end{bmatrix}_{8 \times 8}. \tag{27h}$$

Note that $\mathbf{z}_o, \mathbf{z}_b \in \mathbb{R}^3$ are pure quaternions; as $\mathbf{z}_b^* = -\mathbf{z}_b$ and $\mathbf{z}_o, \mathbf{z}_b \in \mathbb{R}^3$, we have,

$$\begin{aligned}
 [\mathbf{z}_o]_{\otimes} [\mathbf{z}_b]_{\otimes}^{\vee} &= \begin{bmatrix} [\mathbf{z}_o]_{\times} & \mathbf{z}_o \\ -\mathbf{z}_o^T & 0 \end{bmatrix} \begin{bmatrix} [\mathbf{z}_b]_{\times}^T & \mathbf{z}_b \\ -\mathbf{z}_b^T & 0 \end{bmatrix} \\
 &= \begin{bmatrix} [\mathbf{z}_o]_{\times} [\mathbf{z}_b]_{\times}^T - \mathbf{z}_o \mathbf{z}_b^T & (\mathbf{z}_o \times \mathbf{z}_b) \\ (\mathbf{z}_o \times \mathbf{z}_b)^T & -\mathbf{z}_o^T \mathbf{z}_b \end{bmatrix}. \tag{27i}
 \end{aligned}$$

Thus M_A forms a symmetric matrix. From the fact that \mathbf{z}_o and \mathbf{z}_b are unit vectors, one can find $([\mathbf{z}_o]_{\otimes} [\mathbf{z}_b]_{\otimes}^{\vee})^T ([\mathbf{z}_o]_{\otimes} [\mathbf{z}_b]_{\otimes}^{\vee}) = \mathbf{I}_4$, and we obtain

$$([\mathbf{z}_o]_{\otimes} [\mathbf{z}_b]_{\otimes}^{\vee}) \mathbf{v} = \lambda \mathbf{v} \tag{27j}$$

$$([\mathbf{z}_o]_{\otimes} [\mathbf{z}_b]_{\otimes}^{\vee})^T ([\mathbf{z}_o]_{\otimes} [\mathbf{z}_b]_{\otimes}^{\vee}) \mathbf{v} = \lambda ([\mathbf{z}_o]_{\otimes} [\mathbf{z}_b]_{\otimes}^{\vee}) \mathbf{v}, \tag{27k}$$

$$(\mathbf{I}_4 - \lambda_p^2 \mathbf{I}_4) \mathbf{v} = 0, \tag{27l}$$

where λ denotes an eigenvalue of $([\mathbf{z}_o]_{\otimes} [\mathbf{z}_b]_{\otimes}^{\vee})$, and \mathbf{v} is the corresponding eigenvector. The eigenvalues of $([\mathbf{z}_o]_{\otimes} [\mathbf{z}_b]_{\otimes}^{\vee})$, on the other hand, are

$$\lambda = -1, -1, 1, 1. \tag{27m}$$

Now, rewrite the function appearing in Eq. (26a) as

$$f_3 = \tilde{\mathbf{q}}^T M_A \tilde{\mathbf{q}} + \cos \psi + \tilde{\mathbf{q}}^T E_u \tilde{\mathbf{q}} - 1, \quad (27n)$$

where E_u is presented in Eq. (15b). The quadratic form on the Hessian of f_3 is then,

$$\begin{aligned} \tilde{\mathbf{q}}^T (\nabla^2 f_3) \tilde{\mathbf{q}} &= 2\mathbf{q}^T ([\mathbf{z}_o]_{\otimes} [\mathbf{z}_b]_{\otimes}^{\vee}) \mathbf{q} + 2\mathbf{q}^T \mathbf{q} \\ &= 2\mathbf{q}^T ([\mathbf{z}_o]_{\otimes} [\mathbf{z}_b]_{\otimes}^{\vee} + \mathbf{I}_4) \mathbf{q}. \end{aligned} \quad (27o)$$

Since the eigenvalues of $[\mathbf{z}_o]_{\otimes} [\mathbf{z}_b]_{\otimes}^{\vee}$ are greater than -1 , we conclude that $[\mathbf{z}_o]_{\otimes} [\mathbf{z}_b]_{\otimes}^{\vee} + \mathbf{I}_4 \succeq 0$. Thus,

$$\nabla^2 f_3(\tilde{\mathbf{q}}) \succeq 0, \quad (27p)$$

and $f_3(\tilde{\mathbf{q}})$ is convex. \square

Note that this constraint is convex on the entire domain of the unit dual quaternion $\tilde{\mathbf{q}} \in (\mathbf{S}^3 \times \mathbb{R}^3)$.

D. Thrust Direction Constraints

When the spacecraft is treated as a point mass with a single thrust vector representing the net thrust, it can be assumed that the net thrust on the spacecraft can be pointed arbitrarily. However, in the case of a

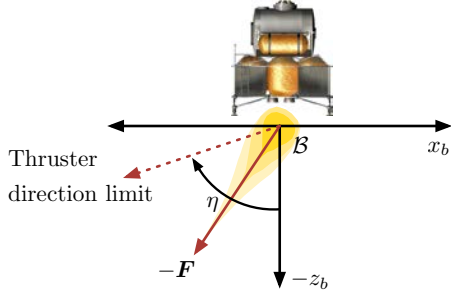


Figure 7. Net thrust direction constraint

rigid body model, there should be an extra pointing constraint on the net thrust since in reality, the thrusters can not be assembled on all sides of the body. This may imply that the thrust direction should not deviate more than the angle η from the z body axis as depicted in Figure 7. In this section, this type of constraint is represented as,

$$\mathbf{z}_b \cdot \frac{\mathbf{F}}{\|\mathbf{F}\|} \geq \cos \eta, \quad (28a)$$

where \mathbf{z}_b denotes a body-fixed z axis and \mathbf{F} denotes the net thrust vector represented in the body-fixed frame \mathcal{B} . For the presented application, we make the reasonable assumption that $-90^\circ \leq \eta \leq 90^\circ$.

Proposition 5. *The thrust direction constraint can be represented in terms of the net thrust vector \mathbf{F} as*

$$f_4(\mathbf{F}) = -\mathbf{z}_b^T \mathbf{F} + \cos \eta \|\mathbf{F}\| \leq 0, \quad (28b)$$

where $f_4 : \mathbf{F} \rightarrow \mathbb{R}$ is convex.

Proof. This type of constraint on the net thrust vector \mathbf{F} defines a convex set as it comes in the standard form of a second order cone constraint [55]. \square

IV. Autonomous Landing with Constraints

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). The problem is formulated as,

$$\min \int_0^{t_f} \begin{bmatrix} \tilde{\omega} \\ \tilde{\mathbf{q}} \end{bmatrix}^T Q \begin{bmatrix} \tilde{\omega} \\ \tilde{\mathbf{q}} \end{bmatrix} + \tilde{\mathbf{F}}^T R \tilde{\mathbf{F}} dt \quad (29a)$$

$$\text{subject to } \mathbf{J} \dot{\tilde{\omega}} + \tilde{\omega} \times \mathbf{J} \tilde{\omega} = \tilde{\mathbf{F}} + \tilde{\mathbf{g}}_b, \quad (12d)$$

$$\dot{\tilde{\mathbf{q}}} = \frac{1}{2} \tilde{\mathbf{q}} \otimes \tilde{\omega}, \quad (9)$$

$$f_1(\tilde{\mathbf{q}}) = \tilde{\mathbf{q}}^T M_H \tilde{\mathbf{q}} + \|2 E_d \tilde{\mathbf{q}}\| \cos \theta \leq 0, \quad (19a)$$

$$f_2(\tilde{\mathbf{q}}) = -\tilde{\mathbf{q}}^T M_G \tilde{\mathbf{q}} + \|2 E_d \tilde{\mathbf{q}}\| \cos \phi \leq 0, \quad (23a)$$

$$f_3(\tilde{\mathbf{q}}) = \tilde{\mathbf{q}}^T M_A \tilde{\mathbf{q}} + \cos \psi \leq 0 \quad (26a)$$

$$f_4(\mathbf{F}) = -\mathbf{z}_b^T \mathbf{F} + \cos \eta \|\mathbf{F}\| \leq 0, \quad (28b)$$

$$\|\tilde{\mathbf{q}}\|^2 \leq \zeta_{\tilde{\mathbf{q}}} \quad (29b)$$

$$\|\omega_i\| \leq \zeta_{\omega_i}, \quad i = 1, \dots, 8, \quad (29c)$$

$$\|F_i\| \leq \zeta_{F_i}, \quad i = 1, \dots, 8, \quad (29d)$$

$$\tilde{\mathbf{q}}(0) = \tilde{\mathbf{q}}_0, \quad \tilde{\omega}(0) = \tilde{\omega}_0, \quad (29e)$$

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. (18), and $\zeta_{\tilde{\omega}}$ and $\zeta_{\tilde{\mathbf{F}}}$ denote the bounds on velocities and external forces, respectively. Note that the feasible domains of the functions 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 constraints Eqs. (19a) and (23a) as hard constraints that must be satisfied during the descent phase of the mission.

Remark 4. In the problem formulation above, the dynamics in the spacecraft mass rate as well as the corresponding time varying inertia matrix will be ignored in order to focus on the geometrical constraints. Nevertheless, we note that these constraints can seamlessly be integrated into the problem formulation by updating the related states during each iteration in the PWA model; see section VI.

V. Piece-wise Affine Model Predictive Control

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 PWA-MPC computes a sub-optimal trajectory to minimize a given cost function and satisfy constraints over a finite and receding horizon based on sampled PWA system obtained by approximating a nonlinear continuous dynamics.

A. Piece-wise Affine Model

This section describes the PWA model approximation of the aforementioned nonlinear dynamic system written in dual quaternions for the subsequent development of the PWA-MPC. First, the continuous time differential equations Eqs. (9) and (11a) are discretized in time with the sampling period Δt . These equations assume the form

$$\begin{aligned} \mathbf{J} \tilde{\omega}(t+1) &= \mathbf{J} \tilde{\omega}(t) - \Delta t [\mathbf{J} \tilde{\omega}(t)]_{\times}^{\vee} \tilde{\omega}(t) + \Delta t \tilde{\mathbf{F}}(t) \\ &\quad + \Delta t \tilde{\mathbf{I}}_8 [m \tilde{\mathbf{q}}^* \otimes \tilde{\mathbf{I}}_8 \tilde{\mathbf{g}}_o]_{\otimes} \tilde{\mathbf{q}}, \end{aligned} \quad (30a)$$

$$\tilde{\mathbf{q}}(t+1) = \tilde{\mathbf{q}}(t) - \frac{\Delta t}{2} [\tilde{\mathbf{q}}(t)]_{\otimes} \tilde{\omega}(t+1), \quad (30b)$$

where \mathbf{J} denotes an inertia matrix in the body frame, and $[\cdot]_{\times}$ and $[\cdot]_{\otimes}$ are defined in §II.B. Note that the general rigid body motion is associated with a pair of differential equations that are discretized in the context of MPC; the discretization sequence adopted is,

$$\tilde{\mathbf{F}}(t) \implies \tilde{\omega}(t+1) \implies \tilde{\mathbf{q}}(t+1) \quad (31)$$

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

$$x(t) \stackrel{\text{def}}{=} \begin{bmatrix} \tilde{\omega}(t)^T & \tilde{\mathbf{q}}(t)^T \end{bmatrix}_{16 \times 1}^T \quad \text{and} \quad u(t) \stackrel{\text{def}}{=} \tilde{\mathbf{F}}(t), \quad (32)$$

these equations can be represented as

$$M_t x(t+1) = A'_t x(t) + B' u(t), \quad (33a)$$

with

$$A'_t = \begin{bmatrix} \mathbf{J} - \Delta t [\mathbf{J} \tilde{\omega}(t)]^\vee & \Delta t \tilde{\mathbf{I}}_8 [m \tilde{\mathbf{q}}^* \otimes \tilde{\mathbf{I}}_8 \tilde{\mathbf{g}}_o] \otimes \\ \mathbf{0}_{8 \times 8} & \mathbf{I}_8 \end{bmatrix}, \quad (33b)$$

$$M_t = \begin{bmatrix} \mathbf{J} & \mathbf{0}_{8 \times 8} \\ -\frac{\Delta t}{2} [\tilde{\mathbf{q}}(t)] \otimes & \mathbf{I}_8 \end{bmatrix}, \quad B' = \begin{bmatrix} \Delta t \mathbf{I}_8 \\ \mathbf{0}_{8 \times 8} \end{bmatrix}. \quad (33c)$$

Then, Eq. (33a) can be represented as a standard discrete piece-wise affine system as

$$\begin{aligned} x(t+1) &= M_t^{-1} A'_t x(t) + M_t^{-1} B' u(t) \\ &= A_t x(t) + B_t u(t). \end{aligned} \quad (34)$$

Note that M_t^{-1} always exists since three sub-block matrices of M_t are invertible [56].

B. PWA-MPC

In the PWA-MPC setup, once an 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 computational step is repeated with the current state as an initial condition. This loop generates a feedback action that can effectively compensate uncertainties and disturbances. Furthermore, at each time step, constraints and the system model can potentially be time-varying at the expense of weakening theoretical guarantees. In this section, we introduce a PWA-MPC strategy that generates a piece-wise constant control to stabilize the aforementioned discrete time system. In particular, we consider the following discrete PWA system sampled at time instant t as

$$f_t(x_t, u_t) = A_t x_t + B_t u_t, \quad (35)$$

where the matrices A_t and B_t are assumed to be constant during the time interval $[t, t+1]$. Based on this sampled system, we consider the following prediction model within the finite horizon N . The state predicted at step $i+1$ is then defined by

$$x_{(t|i+1)} = A_t x_{(t|i)} + B_t u_{(t|i)}, \quad i = 0, \dots, N-1, \quad (36)$$

where $x_{(t|i)} \in \mathbb{R}^n$ and $u_{(t|i)} \in \mathbb{R}^m$ denote the state vector and the control vector at i th step, while $x_{(t|i+1)}$ denotes the predicted state computed from $x_{(t|i)}$ and $u_{(t|i)}$. This PWA system is subject to constraints

$$x_{(t|i)} \in \mathcal{X} \quad \text{and} \quad u_{(t|i)} \in \mathcal{U}, \quad \text{for all } i, t, \quad (37)$$

where \mathcal{X} and \mathcal{U} are assumed to be convex sets representing state and control constraints, respectively. For the problem of regulation to the origin, we consider a MPC to minimize the following cost function. At the given time t , $\mathcal{C}_t : \mathbb{R}^n \times \mathbb{R}^{Nm} \rightarrow \mathbb{R}_+$ is defined by

$$\mathcal{C}_t = \sum_{i=0}^{N-1} \left(q(x_{(t|i)}) + r(u_{(t|i)}) \right) + p(x_{(t|N)}), \quad (38a)$$

with

$$q(x_{(t|i)}) = x_{(t|i)}^T Q x_{(t|i)}, \quad (38b)$$

$$r(u_{(t|i)}) = u_{(t|i)}^T R u_{(t|i)}, \quad (38c)$$

$$p(x_{(t|N)}) = x_{(t|N)}^T P x_{(t|N)}, \quad (38d)$$

where $x_{(t|i)} \in \mathbb{R}^n$ denotes the state vector at the i th step, $u_{(t|i)} \in \mathbb{R}^m$ denotes the control input, $N \in \mathbb{Z}_+$ denotes the finite horizon, and $Q, P \in \mathbb{S}_{++}^n, R \in \mathbb{S}_{++}^m$ denote the weighting matrices on state, final

time state, and control input, respectively. At time t , the PWA-MPC solves the following optimization problem for $i = 0, \dots, N-1$:

$$\min_{U_t} \mathcal{C}_t(x_{(t|0)}, U_t) \quad (39a)$$

$$\text{subject to } x_{(t|i+1)} = A_t x_{(t|i)} + B_t u_{(t|i)}, \quad (39b)$$

$$x_{(t|i)} \in \mathcal{X}, \quad (39c)$$

$$x_{(t|N)} \in \mathcal{X}_f, \quad (39d)$$

$$u_{(t|i)} \in \mathcal{U}, \quad (39e)$$

$$x_{(t|0)} = x_t(0), \quad (39f)$$

where $U_t = [u_{(t|0)}, \dots, u_{(t|N-1)}]$ denotes the control vector to be optimized at time t , whereas $X_t = [x_{(t|0)}, \dots, x_{(t|N)}]$ denotes the predicted state vector corresponding to U_t ; $x_{(t|0)}$ denotes the first state fed-back by the measurement at time t and \mathcal{X} and \mathcal{U} denote the given convex sets for states and control, respectively. In this setup, \mathcal{X}_f denotes the terminal set and the condition $x_{(t|N)} \in \mathcal{X}_f$ is required for stability guarantees. 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 as

$$U_t^* = [u_{(t|0)}^*, u_{(t|1)}^*, \dots, u_{(t|N-2)}^*, u_{(t|N-1)}^*], \quad (40)$$

and for $i = 0, \dots, N$, the optimal predicted-state trajectory

$$X_t^* = [x_{(t|1)}^*, \dots, x_{(t|N)}^*] \quad (41)$$

is obtained by applying U_t^* to the system. Then, the first element of U_t^* is implemented for the actual plant as

$$u_t^* = u_{(t|0)}^*. \quad (42)$$

The corresponding closed-loop PWA system of Eq. (39b) at time instance t is thereby given as

$$x_{(t|1)} = A_t x_{(t|0)} + B_t u_t^* = f_{(t|cl)}(x_t, u_t^*), \quad (43)$$

where $x_t = x_{(t|0)}$. The optimization problem is then repeated at time $t+1$ over one step receded horizon, based on the newly available measurement $x_{(t+1|0)} = x_{t+1}$. Note that the cost function Eq. (38a) is convex. When state constraints Eq. (39c) and control constraints Eq. (39e) are convex, the optimization problem is convex and hence can be efficiently solved by available convex programming solvers [55].

C. Reachability and Stability Analysis

In the PWA-MPC approach, the optimization problem Eq. (39a) is sequentially solved over a finite horizon. Distinct from the properties inherent from an infinite-horizon feedback control design such as a linear quadratic regulator (LQR), the PWA-MPC approach possesses feasibility and stability issues that need to be addressed explicitly. First, the PWA-MPC procedure may lead to a situation where after a few steps the corresponding finite horizon optimal control problem becomes infeasible. Second, the computed controls may not converge to the origin, e.g., the closed loop system is not automatically guaranteed to be asymptotically stable in the PWA-MPC setup. We note that in general, feasibility and stability of PWA-MPC are not ensured by increasing the number of finite horizons since the prediction model is time varying. As such the analysis of PWA-MPC requires additional conditions to address these issues. The stability of PWA-MPC hinges upon how the terminal weight has been parameterized through the matrix P in Eq. (38a); moreover, the terminal constraint set \mathcal{X}_f needs to be chosen such that PWA-MPC is guaranteed to have a stabilizing control about the desired equilibrium. On the other hand, the feasibility of PWA-MPC dictates a bound on the maximum time step Δt since discretized dynamics is used for control computation. In order to effectively approximate the continuous dynamics, we impose an extra constraint in order to bound the variation in the state during each time interval. In this bound, we can include discretization error and sensor uncertainties, allowing an additional flexibility on how fast the discretized dynamics needs to be updated. In particular, we let

$$\frac{\|x_{(t|i+1)} - x_{(t|i)}\|}{\Delta t} \geq \|\dot{x}_{\max}\| + D_{\max} + U_{\max}, \quad (44)$$

where \dot{x}_{\max} , D_{\max} , and U_{\max} denote the maximum rate of change for the state, discretization error bounds, and sensor uncertainties, respectively. From the above assumption, we can find an upper bound on the feasible time step Δt . Note that for the simulation example presented in the paper, we have fixed the time step Δt and adjusted the parameters $\|\dot{x}_{\max}\|$ using a Monte Carlo approach with various initial conditions; see section VI.

In the following section, we investigate conditions for the *persistent feasibility* of the PWA-MPC and *uniformly asymptotic stability* of the origin for the resulting closed system. Persistent feasibility refers to the desirable property where the PWA-MPC remains sequentially feasible. First, we examine conditions that can ensure feasibility.

Assumption 1. The PWA system is subject to state and input constraints of the form,

$$x_{(t|i)} \in \mathcal{X} \text{ and } u_{(t|i)} \in \mathcal{U}, \quad \text{for all } i, t, \quad (45)$$

where $\mathcal{X} \in \mathbb{R}^n, \mathcal{U} \in \mathbb{R}^m$ are compact convex sets containing the origin in their interiors.

Definition 1 (Feasible Set). At time t , $x_{(t|i)} \in \mathcal{X}_{(t|i)} \subseteq \mathcal{X}$ is said to be feasible for a given set $\mathcal{X}_{(t+i+1)}$ if there exists $u_{(t|i)} \in \mathcal{U}$ such that

$$x_{(t+i+1)} = A_t x_{(t|i)} + B_t u_{(t|i)}, \quad (46)$$

where $x_{(t+i+1)} \in \mathcal{X}_{(t+i+1)}$. Furthermore, for a given set $\mathcal{X}_{(t+N)}$, the set of feasible initial states $x_{(t|i)} \in \mathcal{X}_{(t|i)}$ is said to be an **N-step feasible set** if there exists a set of controls $[u_{(t|0)}, u_{(t|1)}, \dots, u_{(t|N-1)}] \subseteq \mathcal{U}$ such that $x_{(t+j)} \in \mathcal{X}$ for $j = i + 1, \dots, N - 1$ and $x_{(t+N)} \in \mathcal{X}_f$.

Note that the feasible state set $\mathcal{X}_{(t|i)}$ is time-dependent since it is determined based on the PWA system sampled at time t .

Assumption 2. The terminal set \mathcal{X}_f includes a neighborhood of the origin, i.e., $0 \in \mathcal{X}_f$ and there exists a terminal controller, $u_f \in \mathcal{U}$ such that

$$x_{(t+N)} \in \mathcal{X}_f \implies A_t x_{(t+N)} + B_t u_f \in \mathcal{X}_f, \quad (47)$$

where (A_t, B_t) denotes the PWA system sampled at time t .

Our last assumption on the PWA-MPC setup pertains to the transition between two subsequent PWA systems (A_t, B_t) and (A_{t+1}, B_{t+1}) . The “distance” between these two subsequent systems is used for the stability analysis of the overall PWA-MPC procedure.

Assumption 3. Suppose that the solution of PWA-MPC at time t is $U_t^* = [u_{(t|0)}^*, \dots, u_{(t|N-1)}^*]$, results in the state transition from $x_{(t|0)}$ to the set $\mathcal{X}_{(t+N)}$. Then the implementation of same control sequence for the subsequent PWA system at $t + 1$ leads to the transition from $x_{(t+1|0)}$ to $\mathcal{X}_{(t+N)}$. Note that the set of controls implemented on the subsequent PWA system is over a horizon of length $N - 1$. This assumption essentially bounds the variation on the PWA system at time t and $t + 1$.

Proposition 6. Consider the feasible PWA-MPC (39a) at time t . Under Assumptions 1-3 and $\mathcal{X}_{(t+N)} \subseteq \mathcal{X}_f$, the PWA-MPC is persistently feasible.

Proof. It suffices to show that the feasibility of the optimization problem at time t implies the feasibility of the optimization problem at time $t + 1$; persistent feasibility then follows by recursion. This however follows from Assumptions 2-3, as $x_{(t+1|0)}$ is in the feasible set of \mathcal{X}_f . \square

We now turn our attention to stability properties of the proposed PWA-MPC using a Lyapunov technique. In particular, we show that if the terminal set \mathcal{X}_f is suitably chosen, the PWA-MPC admits a Lyapunov function that is analogous to the notion of cost-to-go in optimal control.

Theorem 2 (Uniform Asymptotic Stability for Discrete-Time Systems [57, 58]). Given the closed-loop system Eq. (43), with $\mathcal{X} \in \mathbb{R}^n$ as the state constraint set containing the origin as its equilibrium, suppose there exists a continuous Lyapunov function $\mathcal{C}(x_{(t|0)}) : \mathbb{R}^n \times \mathbb{Z}_+ \rightarrow \mathbb{R}_+$ for $x_{(t|0)} \in \mathcal{X}$, such that

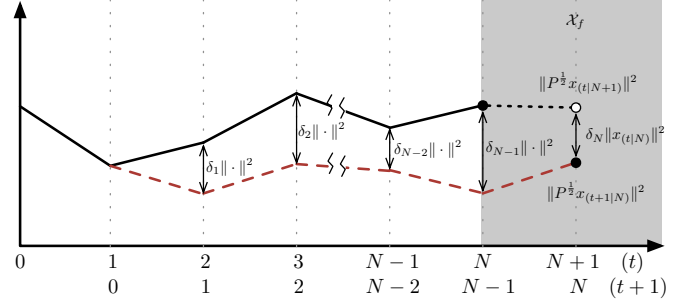


Figure 8. 2-norm square difference between weighted states over finite horizon

1. $\mathcal{C}(0) = 0$ and $\mathcal{C}(x_{(t|0)}) > 0$ when $x_{(t|0)} \neq 0$,
2. $\mathcal{C}(x_{(t+1|0)}) - \mathcal{C}(x_{(t|0)}) \leq -\alpha(x_{(t|0)})$,

where $\alpha(\cdot)$ denotes a positive definite function. Then the origin is uniformly asymptotically stable.

Denote by $\mathcal{C}_t^*(x_{(t|0)}, U_t^*)$ the value of the cost function $\mathcal{C}_t(x_{(t|0)}, U_t)$ in Eq. (38a) at time t when the control law U_t^* has been applied. As the PWA system evolves over time, we need to quantify the difference between two subsequent systems at time t and $t + 1$ in terms of a bound on the state error. We first make an assumption on the variation of the PWA system over the time interval Δt such that the change of predicted states is bounded as shown in Figure 8, namely,

$$\|Q^{\frac{1}{2}} x_{(t+1|i)}\|^2 = \|Q^{\frac{1}{2}} x_{(t|i+1)}\|^2 + \delta_i \|x_{(t|i)}\|^2, \quad \text{for all } i, \quad (48)$$

and for the terminal step N ,

$$\|P^{\frac{1}{2}} x_{(t+1|N)}\|^2 = \|P^{\frac{1}{2}} x_{(t|N+1)}\|^2 + \delta_N \|x_{(t|N)}\|^2, \quad (49)$$

where $P^{\frac{1}{2}} x_{(\cdot)}$ and $Q^{\frac{1}{2}} x_{(\cdot)}$ denote the weighted states via the positive definite matrices P and Q , respectively. We note that $\|x_{(t|i)}\|$ is monotonically decreasing as $t \rightarrow \infty$ and thus $\delta_0, \dots, \delta_{N-1}$ and δ_N can be defined as positive constants, representing the 2-norm square differences between the states depicted in Figure 8. Now consider the following relation,

$$\sum_{i=0}^{N-1} q(x_{(t|i)}) = \sum_{i=0}^{N-1} q(x_{(t|i+1)}) + q(x_{(t|0)}) - q(x_{(t|N)}). \quad (50)$$

Having Eq. (48), we can then find an upper bound on the difference between two predicted states at time t and $t + 1$ as,

$$\sum_{i=0}^{N-1} (q(x_{(t+1|i)}) - q(x_{(t|i)})) = \sum_{i=0}^{N-1} \delta_i \|x_{(t|N)}\|^2 - q(x_{(t|0)}) + q(x_{(t|N)}), \quad (51)$$

and

$$p(x_{(t+1|N)}) - p(x_{(t|N+1)}) = \delta_N \|x_{(t|N)}\|^2. \quad (52)$$

The following proposition provides a sufficient condition for the stability of PWA-MPC.

Proposition 7. Consider the PWA system Eq. (39b) subject to constraints Eqs. (39c)-(39e). Suppose that PWA-MPC control Eq. (42) has been applied under Assumption 3. If the set of initial conditions $\mathcal{X}_{(t|0)}$ is N -step feasible and

$$p(x_{(t|N+1)}) + \delta \|x_{(t|N)}\|^2 + q(x_{(t|N)}) - p(x_{(t|N)}) \leq 0, \quad (53)$$

for $\delta = \max\{\delta_0, \dots, \delta_N\}$, then the closed-loop system Eq. (43) converges asymptotically to the origin.

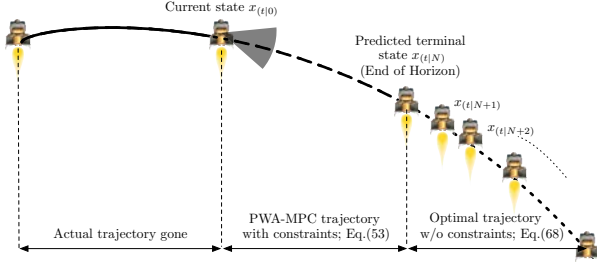


Figure 9. Illustration of PWA-MPC horizon at time instant t

Proof. First, since the PWA-MPC is N -step feasible, we have a feasible $\mathcal{X}_{(t|0)}, \mathcal{X}_{(t|1)}, \dots, \mathcal{X}_{(t|N-1)}$ sets and the PWA-MPC remains persistently feasible. For $x_{(t|0)} \in \mathcal{X}_{(t|0)}$, consider the optimized cost function $\mathcal{C}_t^*(x_{(t|0)}, U_t^*)$ at time t as

$$\mathcal{C}_t^* = p(x_{(t|N)}) + \sum_{i=0}^{N-1} \left(q(x_{(t|i)}) + r(u_{(t|i)}^*) \right), \quad (54a)$$

where $U_t^* = [u_{(t|0)}^*, \dots, u_{(t|N-1)}^*]^T$ denotes an optimal control sequence for minimizing the cost function subject to constraints. After the implementation of the first control $u_{(t|0)}^*$, we obtain

$$x_{(t|1)} = A_t x_{(t|0)} + B_t u_{(t|0)}^*, \quad (54b)$$

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 $x_{(t|1)} \rightarrow x_{(t+1|0)}$. Consider that the PWA-MPC problem at time instance $t+1$. Our aim is to find an upper bound on $\mathcal{C}_{t+1}^*(x_{(t+1|0)}, U_{t+1})$ as we impose the same control sequence U_t^* to the PWA system at $t+1$. Note that due to **Assumption 3**, the set of control U_t^* remains feasible. Moreover, U_{t+1} has the form

$$U_{t+1} = U_t^* = [u_{(t|0)}^*, u_{(t|1)}^*, \dots, u_{(t|N-2)}^*, u_{(t|N-1)}^*]^T. \quad (54c)$$

Note that since two systems are sampled at t and $t+1$, the trajectories generated by U_t do not overlap. Therefore, U_{t+1} is not optimal and $\mathcal{C}_{t+1}^*(x_{(t+1|0)}, U_{t+1})$ is upper bounded by $\mathcal{C}_{t+1}(x_{(t+1|0)}, U_t^*)$ as,

$$\mathcal{C}_{t+1}^*(x_{(t+1|0)}, U_{t+1}) \leq \mathcal{C}_{t+1}(x_{(t+1|0)}, U_t^*) \quad (54d)$$

with

$$\begin{aligned} \mathcal{C}_{t+1}(x_{(t+1|0)}, U_t^*) &= \mathcal{C}_t^*(x_{(t|0)}, U_t^*) - p(x_{(t|N)}) + p(x_{(t+1|N)}) \\ &\quad + \sum_{i=0}^{N-1} \left(q(x_{(t+1|i)}) - q(x_{(t|i)}) \right). \end{aligned} \quad (54e)$$

In view of Eqs. (51)-(52), it now follows that if Eq. (53) is satisfied, we have

$$\mathcal{C}_{t+1}^*(x_{(t+1|0)}, U_{t+1}) - \mathcal{C}_t^*(x_{(t|0)}, U_t) \leq -q(x_{(t|0)}), \quad (54f)$$

where $Q \in \mathbb{S}_+^n$ ensures that $\mathcal{C}_t^*(x_t, U_t)$ monotonically decreases along the trajectory generated for the system Eq. (43) for $x_{(t|0)} \in \mathcal{X} \setminus \{0\}$. Since the cost function satisfies all requirements for being a Lyapunov function, using **Theorem 2**, it now follows that the origin in \mathcal{X} is asymptotically stable. \square

Thus, the stability condition results in finding an appropriate terminal cost P that satisfies the inequality,

$$p(x_{(t|N+1)}) + \delta \|x_{(t|N)}\|^2 + q(x_{(t|N)}) - p(x_{(t|N)}) \leq 0. \quad (53)$$

Since the PWA-MPC is persistently feasible, from **Assumption 2** we have

$$x_{(t|N)} \text{ and } x_{(t|N+1)} \in \mathcal{X}_f, \quad (55)$$

where \mathcal{X}_f is the terminal set for the following closed loop system,

$$\begin{aligned} x_{(t|N+1)} &= A_t x_{(t|N)} + B_t u_f \\ &= (A_t + B_t K_\infty) x_{(t|N)} = A_{cl} x_{(t|N)}, \end{aligned} \quad (56)$$

where $u_f = K_\infty x_{(t|N)}$ is the associated infinite horizon LQ optimal controller computed via,

$$K_\infty = -(B_t P_\infty B_t + R)^{-1} B_t P_\infty A_t, \quad (57)$$

and P_∞ satisfies the discrete time algebraic Riccati equation (ARE),

$$P_\infty = Q + A^T P_\infty A A^T P_\infty B (R + B^T P_\infty B)^{-1} B^T P_\infty A. \quad (58)$$

This terminal control is not applied to the system since $N \geq 1$, but is required to find a feasible P for the PWA-MPC optimization, namely, Eq. (39a). By substituting the closed loop system Eq. (56) into Eq. (53), we obtain

$$x_{(t|N)}^T \left[A_{cl}^T P A_{cl} + Q - P + \delta \mathbf{I}_n \right] x_{(t|N)} \leq 0, \quad (59)$$

where $x_{(t|N)} \in \mathcal{X}_f$. This condition is satisfied by solving the following discrete Lyapunov equation,

$$A_{cl}^T P A_{cl} + Q - P + \delta \mathbf{I}_n = 0, \quad (60)$$

which yields the solution to the desired weighting matrix P . As the system is closed-loop stable, the state enters the terminal region \mathcal{X}_f in finite time. The schematic of PWA-MPC with stability assurance is illustrated in Figure 9 with an application to the constrained optimal landing. This figure depicts three phases of the PWA-MPC algorithm. The first phase shows the actual trajectory the lander spacecraft has travelled. In the second phase, the predicted trajectory is then represented by the dashed line where all constraints are considered. The trajectory in the last phase corresponds to the terminal control where the stability of the origin is guaranteed. Note that state constraints are not considered in the last stage. However, as the last stage proceeds by second phase of the PWA-MPC, one step in the third phase is regenerated to satisfy the constraints since the optimization problem is assumed to be persistently feasible. We summarize the proposed PWA-MPC as follows:

1. Obtain the state measurements of the system and formulate the PWA system matrices A_t and B_t .
2. Given δ , solve the ARE to obtain K_∞ and A_{cl} .
3. Compute a set of optimal controls by minimizing the given cost function \mathcal{C}_t over the horizon N .
4. Implement the first part of the optimal control until new measurements of the state are available.
5. Continue with step 1.

In the next section, we present the application of the above PWA-MPC approach to the precision landing problem in the presence of state constraints.

D. Precision Landing with Constraints via PWA-MPC

In this section, we formulate the precision landing algorithm in the presence of translational and rotational constraints using the design technique described in the previous section. The optimization problem for solving the PWA-MPC is defined as follows:

$$\min_{U_t} \mathcal{C}_t = \sum_{i=0}^{N-1} \left(q(x_{(t|i)}) + r(u_{(t|i)}) \right) + p(x_{(t|N)}) \quad (61a)$$

$$\text{s.t. } x_{(t|i+1)} = A_t x_{(t|i)} + B_t u_{(t|i)}, \quad i = 0, \dots, N-1 \quad (34)$$

$$x_{(t|i+1)}^T M_H' x_{(t|i+1)} + \|2 E_d' x_{(t|i+1)}\| \cos \theta \leq 0, \quad (19a')$$

$$x_{(t|i+1)}^T M_G' x_{(t|i+1)} + \|2 E_d' x_{(t|i+1)}\| \cos \phi \leq 0, \quad (23a')$$

$$x_{(t|i+1)}^T M_A' x_{(t|i+1)} + \cos \psi \leq 0 \quad (26a')$$

$$M_F' u_{(t|i)} + \cos \eta \|E_F' u_{(t|i)}\| \leq 0, \quad (28b')$$

$$-\zeta_x \leq x_{(t|i+1)} \leq \zeta_x \quad (61b)$$

$$-\zeta_u \leq u_{(t|i)} \leq \zeta_u, \quad (61c)$$

$$x_{(t|0)} = x_0(t), \quad (61d)$$

with

$$x_{(t|i)} = \begin{bmatrix} \tilde{\omega}_{(t|i)} \\ \tilde{\mathbf{q}}_{(t|i)} \end{bmatrix} \quad \text{and} \quad u_{(t|i)} = \tilde{\mathbf{F}}_{(t|i)} = \begin{bmatrix} \mathbf{F}_{(t|i)} \\ \mathbf{T}_{(t|i)} \end{bmatrix}, \quad (61e)$$

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. As we have defined a new state variable, the four types of constraints as well as the dynamics have been parameterized using the following matrices:

$$M'_H = \begin{bmatrix} 0 & 0 \\ 0 & M_H \end{bmatrix}, \quad M'_G = \begin{bmatrix} 0 & 0 \\ 0 & -M_G \end{bmatrix}, \quad (61f)$$

$$M'_A = \begin{bmatrix} M_A & 0 \\ 0 & 0 \end{bmatrix}, \quad E'_d = \begin{bmatrix} 0 & 0 \\ 0 & E_d \end{bmatrix}, \quad (61g)$$

$$M'_F = \begin{bmatrix} -\mathbf{z}_b & 0 \end{bmatrix}, \quad E'_F = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}. \quad (61h)$$

We denote as $U_t^* = [u_{(t|0)}^*, \dots, u_{(t|N-1)}^*]$ the sequence of optimal control 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 PWA system model A_{t+1} , B_{t+1} with the new initial condition x_{t+1} . The matrix P in the terminal cost is calculated from Eq. (60) to ensure the overall stability and reachability of the overall system.

VI. Numerical Simulations

In this section, we present a simulation example to demonstrate the viability of the proposed PWA-MPC approach for the Mars autonomous landing problem in the presence of constraints. In the scenario considered, the Mars lander is required not to violate four types of rotational and translational constraints. The energy consumption and computational efficiencies of the proposed algorithms are then compared. The simulations are carried out on a Mac Pro with Matlab R2015a software; the convex PWA-MPC problem is solved using the Matlab based interior point optimizer (IPOPT) solver [59] included in the OPTI Toolbox [60]. Although the simulations were not carried out using a custom solver optimized for the real time landing problem, the rapid execution time and the convergence guarantees make the proposed approach a viable candidate for onboard guidance control for Mars landing.

In the simulations, we use the following discretized models for mass $m(t)$ and inertia J updates at every iteration:

$$m(t+1) = m(t) - \frac{\Delta t}{g I_{sp}} \|\tilde{\mathbf{F}}(t)\| \quad (62)$$

and

$$J(t+1) = \begin{bmatrix} J_{11} & 0 & 0 \\ 0 & J_{22} & J_{23} \\ 0 & J_{32} & J_{33} \end{bmatrix}, \quad (63)$$

with

$$\begin{aligned} J_{11} &= 1.75m(t+1), & J_{22} &= 1.31m(t+1) + 0.5m(t+1) \\ J_{23} &= J_{32} = 0.75m(t+1) - 0.5\sqrt{3}m(t+1) \\ J_{33} &= 0.44m(t+1) + 1.5m(t+1), \end{aligned}$$

where g denotes the gravitational acceleration and I_{sp} denotes the specific impulse of the thruster. The lander spacecraft is considered as a solid cylinder with radius 2 m and height 3 m, and the body frame is assumed to be rotated by 30° about the b_z axis from the principal frame. The spacecraft's physical properties and detailed parameters for the four types of rotationally and translationally constraints are specified in Table 1. The Mars lander configuration with the line of sight vector is depicted in Figure 10. The PWA-MPC

Table 1. Lander Spacecraft Parameters

Parameters	Values
Initial Moment of Inertia	$J = \begin{bmatrix} 1347.5 & 0 & 0 \\ 0 & 1395.6 & -83.5 \\ 0 & -83.5 & 1491.8 \end{bmatrix} \text{ kg m}^2$
Initial Mass	$m = 770.07 \text{ kg}$
Gravity	$g_o = [0, 0, -3.7114] \text{ m/s}^2$
Line of sight	$\mathbf{y}_b = [0, \sqrt{2}/2, -\sqrt{2}/2]$, $\theta = 50^\circ$
Glide slope	$\mathbf{z}_o = [0, 0, 1]$, $\phi = 65^\circ$
General attitude	$\psi = 80^\circ$
Thrust Direction	$\eta = 80^\circ$

simulation parameters as well as initial and final conditions for the lander spacecraft are given in Table 2. The desired attitude and position for the precision landing are defined as the identity unit dual quaternions. Based on the initial and desired attitudes and positions, the corresponding initial unit dual quaternion is computed as $\tilde{\mathbf{q}}(0) = [-0.38, 0.49, -0.40, 0.67, 187.31, -93.75, 269.61, 336.95]^T$, where both initial and desired attitudes $\mathbf{q}(0), \mathbf{q}(t_f)$ and positions $\mathbf{r}_o(0), \mathbf{r}_o(t_f)$ are randomly chosen from a set of corresponding dual quaternions that satisfy all constraints. The system state bounds are described in Table 3. In particular, control bounds are reasonably defined from the Monte Carlo simulations in which the controls do not violate V_{max} . The Monte Carlo simulations have been carried out with 100 samples randomly selected from feasible state sets and prediction horizons between 8 s – 44 s. The sampling rate of $\Delta t = 1$ s is determined to satisfy the requirement, Eq.(44). For a representative simulation scenario, the predicting horizon has been selected to 16 seconds. The corresponding PWA-MPC computation times are shown to vary between 0.1993s to 0.6125s in every iteration; on average, the computation time yields 0.394s to generate the next step control $u_{(t+1|i)}$. The optimization stopping criteria for PWA-MPC is chosen such that the error in unit dual quaternions is bounded as

$$\text{error} = \|\tilde{\mathbf{q}}_{(t+1|i)} - \tilde{\mathbf{q}}_{(t|i)}\|_2 \leq 0.01; \quad (64)$$

such a bound corresponds to less than 2 m distance from the desired landing position. Note that in order to address the significant scale difference between the unit quaternion and the position vector (the second part of the dual quaternion), we have performed a non-dimensionalization process in order to avoid a round-off error during the numerical computation. The desired position for landing is at the origin represented as the center of the blue glide slope cone in the following figures.

In Figure 11, the overall trajectory of the Mars lander is depicted along with the representation of the line of sight and glide slope constraints. For reference, we have also plotted a globally fuel-optimal powered descent trajectory with a thrust pointing constraint ($\leq 80^\circ$) examined in [33]. We note however that these two trajectories are not directly comparable since the trajectory examined in [33] is for a point mass model. In fact, for the point mass model, the corresponding trajectory can in general violate the rotational and translational constraints. We also note that distinct from the point mass model, the shape of the overall trajectory is affected by the initial and desired spacecraft attitudes. This is due to the fact that the rotational and translational motions are combined via the unit dual quaternion parameterization and the shortest path between two points on the unit dual quaternion manifold is expressed via a screw motion on SE(3). We also note that by weighting the second part of the dual quaternion pertaining to the lander's translational dynamics, one can dictate a different convergence rate between the rotational and translational motions of the lander.

Figure 12 depicts the time histories for the position and attitude in unit quaternions of the Mars lander observed with respect to a ground fixed frame while Figure 13 depicts the time histories for translational and angular velocities in the body-fixed frame. The translational forces and torques acting along the body-fixed axes are represented in Figure (14). Note that the translational forces never vanish since there exists a gravitational force acting downwards and forces along \mathbf{z}_b to

Table 2. Presented PWA-MPC simulation parameters

Parameters	Values
Sampling period	$\Delta t = 1$ s
Stopping criteria	error ≤ 0.01
Predicting horizons	28 s (28 steps)
Initial Attitude	$\mathbf{q}(0) = [-0.3841, 0.4913, -0.4009, 0.6710]$
Initial Position (fixed frame)	$\mathbf{r}_o(0) = [0.700, -0.400, 0.520]$ km
Initial Velocity (body frame)	$\mathbf{v}_b(0) = [-91.55, 38.34, 12.16]$ m/s
Initial Ang. Vel.	$\boldsymbol{\omega}_b(0) = [0, 0, 0]$ rad/s
Mars Gravity	3.711 m/s ²

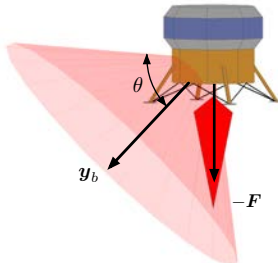
Table 3. System state bounds

Parameters	Values
Max. Distance	$\ \mathbf{r}_b\ \leq 1$ km
Thrust	$ F_x , F_y , F_z \leq 4000$ N
Torque	$ M_x , M_y , M_z \leq 2$ Nm
Velocity	$ v_x , v_y , v_z \leq 95$ m/s
Angular Velocity	$ \omega_x , \omega_y , \omega_z \leq 0.04$ rad/s

Table 4. PWA-MPC Performance Comparison

Prediction Horizon	No. of Iterations	Calculation Time	Terminal Mass	Energy Use (Sum)	
				Trans. Motion	Rot. Motion
8 s	390	0.08 s	82.6 kg	4390 kN	499 Nm
12 s	279	0.22 s	148.6 kg	3901 kN	715 Nm
16 s	220	0.39 s	201.2 kg	3500 kN	866 Nm
20 s	186	0.75 s	238.1 kg	3208 kN	975 Nm
24 s	166	1.29 s	262.2 kg	3010 kN	1040 Nm
28 s	154	1.59 s	278.8 kg	2889 kN	1059 Nm
32 s	146	2.21 s	291.3 kg	2823 kN	1074 Nm
36 s	142	3.19 s	302.2 kg	2793 kN	1081 Nm
40 s	139	4.51 s	309.8 kg	2775 kN	1085 Nm
44 s	140	6.79 s	307.1 kg	2777 kN	1084 Nm
≥ 150 s	125	5.61 s	346.7 kg	2738 kN	N/A

compensate such a gravitational force to maintain the desired altitude during the later part of the simulation. Figure 15 shows the deviation angle over time from the line of sight vector \mathbf{y}_b to the desired landing location whereas Figure 16 represents the trace of the landing location on an azimuthal projection. As we expect from the configuration of the Mars lander shown in Figure 10, then \mathbf{z}_b is close to the boundary of the line of sight, the final vertical descent arises around 40° . The glide slope angle constraint is defined as 65° and Figure 15 represents the glide slope angle over time. In Figure 17, the time histories of the general attitude angle and net thrust direction angle are presented. The initial general attitude angle is close to 80° since the powered descent phase initiates right after the parachute phase. As a result, the trajectories of the net thrust direction exhibit large deviations to compensate the gravitational acceleration and a large horizontal velocity. This can also be observed in Figure 11.

**Figure 10. Mars lander configuration with the line of sight vector \mathbf{y}_b**

We also perform a series of PWA-MPC simulations as we increase the prediction horizons from 8 s to 44 s. The results are summarized in Table 4. Although we uniformly penalize the torque and force in the cost function of PWA-MPC, the longer prediction horizons tend to achieve better energy consumption as well as faster convergence. As shown in Table 4, the number of iterations to achieve the same stopping criteria is dramatically reduced from 390 to 139 as the prediction horizon increases. This is due to the fact that a longer horizon allows the algorithm to synthesize a trajectory that is less restricted by constraint violations at each iteration. In the meantime, the simulations show that beyond 44 s prediction horizon, the number of iterations does not improve as the proposed PWA model fails to approximate the nonlinear rotational dynamics. For reference, the globally optimal solution with no rotational dynamics addressed in [33] shows the number of iterations to be 125 in order to achieve the same stopping criteria with same initial conditions; see Table 4. Figure 18 depicts the time histories of the mass reduction as the prediction horizon increases. As expected, the trajectory synthesized via a longer prediction horizon uses less propellant and hence a better payload capability.

The penalty on computation time as the prediction horizon increases is predictable and can be dramatically improved by utilizing an IPOPT solver specialized for autonomous landing. We however note that as shown in the simulation the longer prediction does not always lead to a improved performance as a fast maneuver can potentially have an adverse effect on the quality of the approximation of the PWA model.

VII. Conclusions

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 four translational and rotational constraints. Subsequently, we derived a piece-wise affine model for the rotational and translational control synthesis based on model predictive control. The resulting PWA-MPC approach was shown to provide a unifying framework through which various classes of constraints can addressed for powered descent guidance, in addition to a wide array of problems for autonomous spacecraft operations.

Acknowledgements

The authors thank the Guest Editor Panagiotis Tsiotras and the reviewers for their constructive suggestions and corrections on the original version of this manuscript.

References

- [1] Steinfeldt, B. A., Grant, M. J., Matz, D. A., Braun, R. D., and Barton, G. H., "Guidance, Navigation, and Control System Performance Trades for Mars Pinpoint Landing," *Journal of Spacecraft and Rockets*, 2010, pp. 188–198. doi:10.2514/1.45779.
- [2] Wolf, A. A., Graves, C., Powell, R., and Johnson, W., "Systems for Pinpoint Landing at Mars," *Proceedings of the AAS/AIAA Space Flight Mechanics Meeting*, 2004.
- [3] Rush, B., Bhaskaran, S., and Synnott, S., "Improving Mars Approach Navigation Using Optical Data," *Proceedings of the AAS/AIAA Astrodynamics Specialist Conference*, 2001.
- [4] Braun, R. D. and Manning, R. M., "Mars Exploration Entry, Descent, and Landing Challenges," *Journal of Spacecraft and Rockets*, 2007, pp. 310–323. doi:10.2514/1.25116.
- [5] Carson, J. M., Bailey, E. S., Trawny, N., Johnson, A. E., Roback, V. E., Amzajerdian, F., and Werner, R. A., "Operations Concept, Hardware Implementation and Ground-Test Verification of a Hazard Detection System for Autonomous and Safe Precision Lunar Landing," 2013.
- [6] Carson, J. M., Johnson, A. E., Hines, G. D., Johnson, W., Anderson, F. S., Lawrence, S., Lee, D. E., Huertas, A., Amzajerdian, F., Olansen, J. B., Devolites, J., Harris, W. J., Trawny, N., Condon, G. L., and Nguyen, L., "GN&C Subsystem Concept for Safe Precision Landing of the Proposed Lunar MARE Robotic Science Mission," *AIAA Guidance, Navigation, and Control Conference*, 2016. doi:10.2514/6.2016-0100.

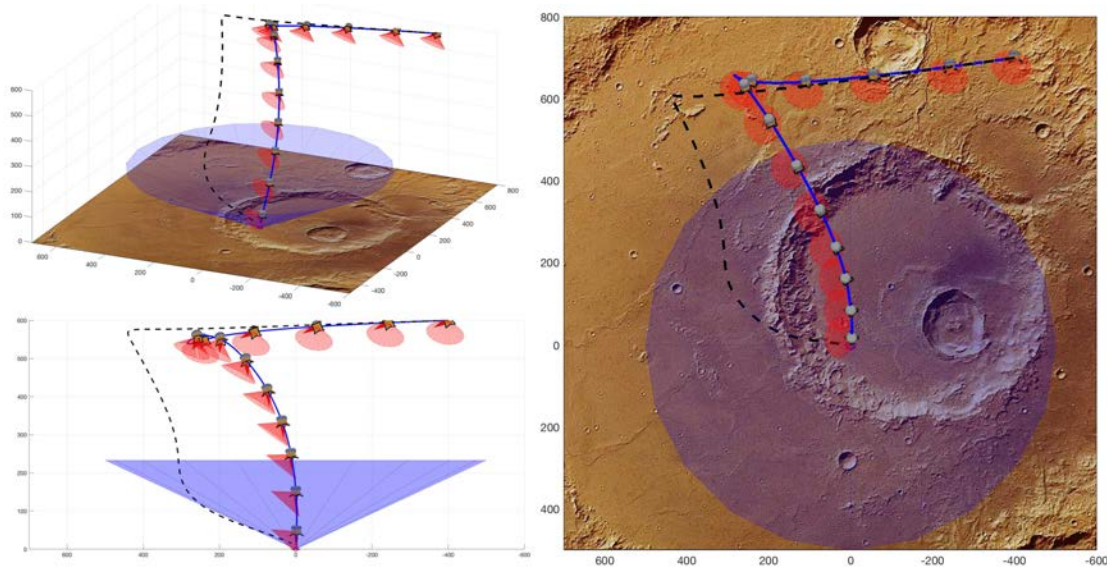


Figure 11. Overall trajectory of Mars lander's pin-point landing along with the representation of the line of sight cone and glide slope.

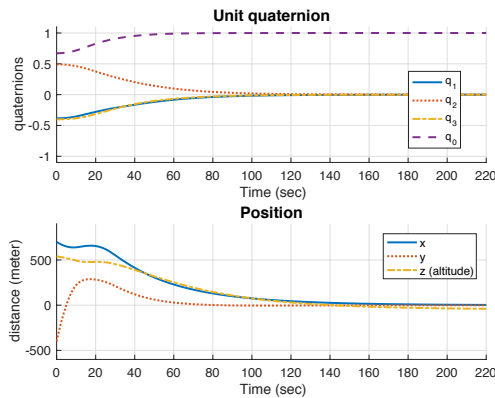


Figure 12. Time histories for the position and attitude of Mars lander observed in the ground fixed frame.

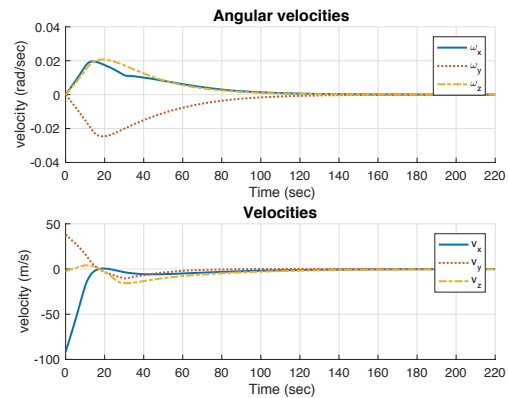


Figure 13. Time histories for the translational and angular velocities in the body-fixed frame.

- [7] Luna, M. E., Huertas, A., Trawny, N., Villalpando, C. Y., Martin, K., Wilson, W., and Restrepo, C., "Simulations of the Hazard Detection System for Approach Trajectories of the Morpheus Lunar Lander," *AIAA Guidance, Navigation, and Control Conference*, 2015. doi:[10.2514/6.2015-0327](#).
- [8] Epp, C., Robertson, E., and Carson, J. M., "Real-Time Hazard Detection and Avoidance Demonstration for a Planetary Lander," *AIAA SPACE 2014 Conference and Exposition*, 2014. doi:[10.2514/6.2014-4312](#).
- [9] Sostaric, R. R., "Powered Descent Trajectory Guidance and Some Considerations for Human Lunar Landing," *Proceedings of the 30th Annual AAS Guidance and Control Conference*, 2007.
- [10] Clifford, W., "Preliminary Sketch of Bi-quaternions," *Proceedings of the London Mathematical Society*, 1873, pp. 381–395.
- [11] Yang, A. T., *Application of Quaternion Algebra and Dual Numbers to the Analysis of Spatial Mechanisms*, Ph.D. thesis, 1963.
- [12] Perez, A. and McCarthy, J. M., "Dual Quaternion Synthesis of Constrained Robotic Systems," *Journal of Mechanical Design*, 2004, pp. 425–435.
- [13] Pham, H. L., Perdureau, V., Adorno, B. V., and Fraisse, P., "Position and Orientation Control of Robot Manipulators Using Dual Quaternion Feedback," *Proceedings of the IEEE/RSJ International Conference of Intelligence Robots System*, 2010.
- [14] Kavan, L., Collins, S., O'Sullivan, C., and Zara, J., "Dual quaternions for Rigid Transformation Blending," Tech. Rep. TCD-CS-2006-46, 2006.
- [15] Schilling, M., "Universally Manipulable Body Models - Dual Quaternion Representations in Layered and Dynamic MMCs," *Autonomous Robots*, 2011, pp. 399–425.
- [16] Filipe, N., Kontitsis, M., and Tsiotras, P., "Extended Kalman Filter for Spacecraft Pose Estimation Using Dual Quaternions," *Journal of Guidance, Control, and Dynamics*, 2015, pp. 1625–1641. doi:[10.2514/1.G000977](#).
- [17] Wang, J. and Sun, Z., "6DOF Robust Adaptive Terminal Sliding Mode Control for Spacecraft Formation Flying," *Acta Astronautica*, 2012, pp. 76–87.
- [18] Wang, J. and Yu, C., "Unit Dual Quaternion-based Feedback Linearization Tracking Problem for Attitude and Position Dynamics," *Systems & Control Letters*, 2013, pp. 225–233.
- [19] Lee, U. and Mesbahi, M., "Optimal Power Descent Guidance with 6-DoF Line of Sight Constraints via Unit Dual Quaternions," *AIAA Guidance, Navigation, and Control Conference*, 2015. doi:[10.2514/6.2015-0319](#).
- [20] Filipe, N. and Tsiotras, P., "Rigid Body Motion Tracking Without Linear and Angular Velocity Feedback Using Dual Quaternions," *IEEE European Control Conference*, 2013, pp. 329–334.
- [21] Filipe, N. and Tsiotras, P., "Adaptive Position and Attitude-Tracking Controller for Satellite Proximity Operations Using Dual Quaternions," *Journal of Guidance, Control, and Dynamics*, 2014, pp. 566–577. doi:[10.2514/1.G000054](#).
- [22] Zu, Y., Lee, U., and Dai, R., "Distributed Motion Estimation of Space Objects Using Dual Quaternions," *AIAA/AAS Astrodynamics Specialist Conference*, 2014. doi:[10.2514/6.2014-4296](#).

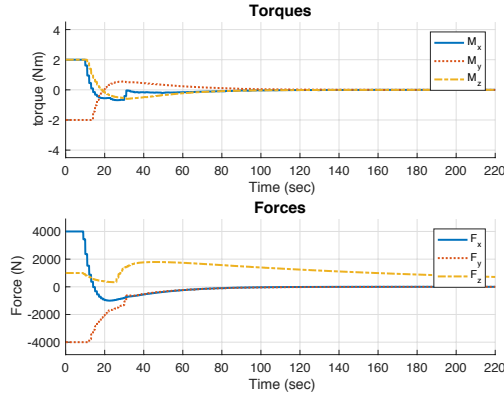


Figure 14. Time histories for the translational forces and torques expressed in the body-fixed frame.

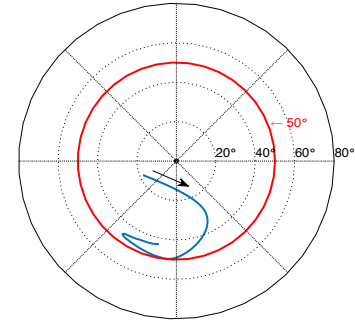


Figure 16. The trace of the target around the line of sight vector (center) on an azimuthal projection.

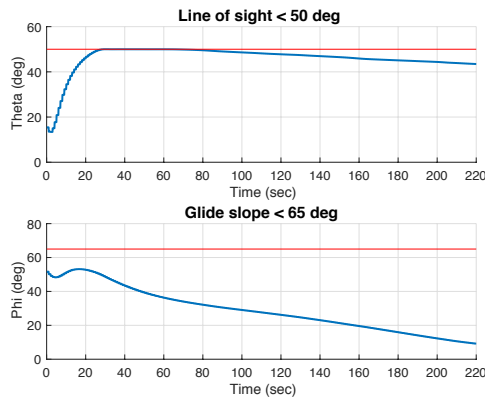


Figure 15. Trace of deviation angle over time from the line of sight vector and the approaching angle over time

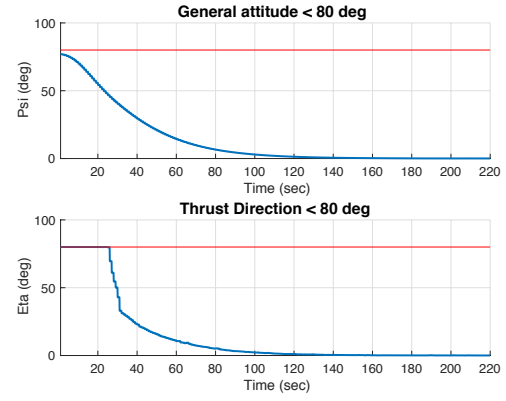


Figure 17. Time histories of the general attitude angle and net thrust direction.

- [23] Wu, Y., Hu, X., Hu, D., and Lian, J., "Strapdown Inertial Navigation System Algorithms Based on Dual Quaternions," *IEEE Transaction on Aerospace and Electronic Systems*, 2005, pp. 110–132.
- [24] Meditch, J. S., "On the Problem of Optimal Thrust Programming for a Lunar Soft Landing," *IEEE Transactions on Automatic Control*, 1964, pp. 477–484.
- [25] Miele, A., *The Calculus of Variations in Applied Aerodynamics and Flight Mechanics in Optimization Techniques*, 1962.
- [26] Klumpp, A. R., "Apollo Lunar Descent Guidance," *Automatica*, 1974, pp. 133–146.
- [27] Guo, J. and Han, C., "Design of Guidance Laws for Lunar Pinpoint Soft Landing," *Proceedings of the AAS/AIAA Astrodynamics Specialist Conference*, 2009.
- [28] Sostaric, R. and Rea, J., "Powered Descent Guidance Methods For The Moon and Mars," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2005. doi:10.2514/6.2005-6287.
- [29] Topcu, U., Casoliva, J., and Mease, K. D., "Minimum-Fuel Powered Descent for Mars Pinpoint Landing," *Journal of Spacecraft and Rockets*, 2007, pp. 324–331. doi:10.2514/1.25023.
- [30] Açıkmeşe, B. and Ploen, S., "A Powered Descent Guidance Algorithm for Mars Pinpoint Landing," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2005. doi:10.2514/6.2005-6288.
- [31] Açıkmeşe, B. and Ploen, S. R., "Convex Programming Approach to Powered Descent Guidance for Mars Landing," *Journal of Guidance, Control, and Dynamics*, 2007, pp. 1353–1366. doi:10.2514/1.27553.
- [32] Açıkmeşe, B., Blackmore, L., Scharf, D. P., and Wolf, A., "Enhancements on the Convex Programming Based Powered Descent Guidance Algorithm for Mars Landing," *Proceedings of the AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, 2008.
- [33] Açıkmeşe, B., Carson, J., and Blackmore, L., "Lossless Convexification of Non-Convex Control Bound and Pointing Constraints of the Soft Landing Optimal Control Problem," *IEEE Transactions on Control System Technology*, 2013, pp. 2104–2113.
- [34] Açıkmeşe, B., Scharf, D., Hadaegh, F., and Murray, E., "A Convex Guidance Algorithm for Formation Reconfiguration," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2006. doi:10.2514/6.2006-6070.
- [35] Morgan, D., Chung, S.-J., and Hadaegh, F. Y., "Model Predictive Control of Swarms of Spacecraft Using Sequential Convex Programming," *Journal of Guidance, Control, and Dynamics*, 2014, pp. 1725–1740. doi:10.2514/1.G000218.
- [36] Schulman, J., Ho, J., Lee, A., Awwal, I., Bradlow, H., and Abbeel, P., "Finding Locally Optimal, Collision-Free Trajectories with Sequential Convex Optimization," *Robotics: Science and Systems*, 2013, pp. 1–10.
- [37] Borrelli, F., Bemporad, A., Fodor, M., and Hrovat, D., "An MPC/Hybrid System Approach to Traction Control," *IEEE Transactions on Control Systems Technology*, 2006, pp. 541–552.
- [38] Keviczky, T. and Balas, G. J., "Flight Test of a Receding Horizon Controller for Autonomous UAV Guidance," *Proceedings of the American Control Conference*, 2005.
- [39] Bartlett, R. A., Biegler, L. T., Backstrom, J., and Gopal, V., "Quadratic Programming Algorithms for Large-scale Model Predictive Control," *Journal of Process Control*, 2002, pp. 775–795.
- [40] Keviczky, T. and Balas, G. J., "Software-Enabled Receding Horizon Control for Autonomous Unmanned Aerial Vehicle Guidance," *Journal of Guidance, Control, and Dynamics*, 2006, pp. 680–694. doi:10.2514/1.15562.
- [41] Mayne, D. Q. and Rakovic, S., "Model Predictive Control of Constrained Piecewise Affine Discrete-time Systems," *International Journal of Robust Nonlinear Control*, 2003, pp. 261–279.

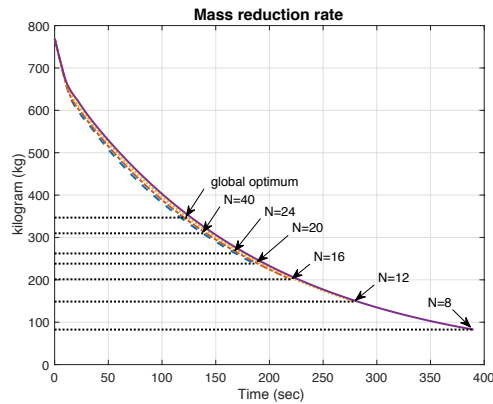


Figure 18. Time histories of the mass reduction rate for the length of prediction horizon.

- [42] Falcone, P., Borrelli, F., Tseng, H. E., Asgari, J., and Hrovat, D., "Linear Time-varying Model Predictive Control and Its Application to Active Steering Systems: Stability Analysis and Experimental Validation," *International Journal of Robust Nonlinear Control*, 2008, pp. 862–875.
- [43] Liu, C., Chen, W. H., and Andrews, J., "Piecewise Constant Model Predictive Control for Autonomous Helicopters," *Robotics and Autonomous Systems*, 2011, pp. 571–579.
- [44] Hedrick, J. and Kang, Y., "Linear Tracking for a Fixed-wing UAV Using Nonlinear Model Predictive Control," *IEEE Transactions on Control Systems Technology*, 2009, pp. 1202–1210.
- [45] Liu, Z., Liu, L., Zhang, J., and Yuan, X., "Model Predictive Control of Piecewise Affine System with Constrained Input and Time Delay," *Abstract and Applied Analysis*, 2013, pp. Article ID 937274.
- [46] Lin, R. and Unbehauen, J. N., "Canonical Piecewise-linear Approximations," *IEEE Transaction on Circuit and Systems I: Fundamental theory and applications*, 1992, pp. 697–699.
- [47] Sontag, E., "Nonlinear Regulation: The Piecewise Linear Approach," *IEEE Transactions on Automatic Control*, 1981, pp. 346–358.
- [48] Imura, J., "Optimal Control of Sampled-data Piecewise Affine Systems," *Automatica*, 2004, pp. 661–669.
- [49] Imura, J. and van der Schaft, A., "Characterization of Well-posedness of Piecewise-linear Systems," *IEEE Transactions on Automatic Control*, 2000, pp. 1600–1619.
- [50] Lee, U. and Mesbahi, M., "Feedback Control for Spacecraft Reorientation under Attitude Constraints via Convex Potentials," *IEEE Transactions on Aerospace and Electronic Systems*, 2014, pp. 2578–2592.
- [51] Joshi, S. M., Kelkar, A. G., and Wen, J. T., "Robust Attitude Stabilization of Spacecraft using Nonlinear Quaternion Feedback," *IEEE Transactions on Automatic Control*, 1995, pp. 1800–1803.
- [52] Liu, S. and Trenkler, G., "Hadamard, Khatri-Rao, Kronecker and other Matrix Products," *International Journal of Information and Systems Sciences*, 2008, pp. 775–795.
- [53] Beutelspacher, A. and Rosenbaum, U., *Projective Geometry: From Foundations to Applications*, 1998.
- [54] Stengel, R. F., *Flight Dynamics*, 2004.
- [55] Boyd, S. and Vandenberghe, L., *Convex Optimization*, 2004.
- [56] Stewart, G. W., *Matrix Algorithms: Volume 1: Basic Decompositions*.
- [57] Kalman, R. and Bertram, J., "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*, 1960, pp. 371–393.
- [58] Kalman, R. and Bertram, J., "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*, 1960, pp. 394–400.
- [59] Wächter, A. and Biegler, L. T., "On the Implementation of a Primal-Dual Interior Point Filter Line Search Algorithm for Large-Scale Nonlinear Programming," *Mathematical Programming*, 2006, pp. 25–57.
- [60] Currie, J. and Wilson, D. I., "OPTI: Lowering the Barrier Between Open Source Optimizers and the Industrial MATLAB User," *Foundations of Computer-Aided Process Operations*, 2012.