Successive Convexification
for 6-DoF Powered Descent Guidance
with Compound State-Triggered Constraints

Michael Szmuk[*], Taylor P. Reynolds[*], Behçet Açıkmeşe[†], Mehran Mesbahi[‡],

University of Washington, Seattle, WA 98195-2400, USA

and

John M. Carson III[§]

NASA Johnson Space Center, Houston, TX 77058, USA

This paper introduces a continuous formulation for *compound* state-triggered constraints, which are generalizations of the recently introduced state-triggered constraints. State-triggered constraints are different from ordinary constraints found in optimal control in that they use a state-dependent *trigger condition* to enable or disable a *constraint condition*, and can be expressed as continuous functions that are readily handled by successive convexification. Compound state-triggered constraints go a step further, giving designers the ability to compose trigger and constraint conditions using Boolean *and* and *or* operations. Simulations of the 6-degree-of-freedom (DoF) powered descent guidance problem obtained using successive convexification are presented to illustrate the utility of state-triggered and compound state-triggered constraints. The examples employ a velocity-triggered angle of attack constraint to alleviate aerodynamic loads, and a collision avoidance constraint to avoid large geological formations. In particular, the velocity-triggered angle of attack constraint demonstrates the ability of state-triggered constraints to introduce new constraint phases to the solution without resorting to combinatorial techniques.

Nomenclature

DoF	Degrees-of-Freedom	m_{ig}	Mass of vehicle at t_{ig}
DCM	Direction cosine matrix	$g_{\mathcal{I}}$	Gravity vector
STC	State-Triggered Constraint	m	Vehicle mass
cSTC	Continuous State-Triggered Constraint	$r_{\mathcal{I}}$	Inertial position of the vehicle
e_i	Unit vector pointing along i^{th}-axis	$v_{\mathcal{I}}$	Inertial velocity of the vehicle
$\mathcal{I}$	Subscript used to denote the inertial frame	$q_{\mathcal{B}\leftarrow\mathcal{I}}$	Unit quaternion relating $\mathcal{F}_{\mathcal{I}}$ to $\mathcal{F}_{\mathcal{B}}$
$\mathcal{B}$	Subscript used to denote the body frame	$\omega_{\mathcal{B}}$	Angular velocity vector
$\mathcal{F}_{\mathcal{I}}$	The inertial UEN frame	$T_{\mathcal{B}}$	Commanded thrust vector
$\mathcal{F}_{\mathcal{B}}$	The body frame	$C_{\mathcal{B}\leftarrow\mathcal{I}}$	DCM rotating from $\mathcal{F}_{\mathcal{I}}$ to $\mathcal{F}_{\mathcal{B}}$
t	Time	$C_{\mathcal{I}\leftarrow\mathcal{B}}$	DCM rotating from $\mathcal{F}_{\mathcal{B}}$ to $\mathcal{F}_{\mathcal{I}}$
t_{in}	Initial time epoch	γ_{gs}	Glide-slope cone constraint angle
t_{ig}	Ignition time epoch	θ_{max}	Maximum allowable tilt angle
t_f	Final time epoch	ω_{max}	Maximum allowable angular rate
t_c	Coast time	δ_{max}	Maximum allowable gimbal angle
$t_{c,max}$	Max allowable coast time	T_{min}	Minimum allowable thrust magnitude
t_b	Burn time	T_{max}	Maximum allowable thrust magnitude
m_{dry}	Dry mass of the vehicle		

[*]PhD Student, AIAA Student Member, {mszmuk,tpr6}@uw.edu
[†]Associate Professor, AIAA Associate Fellow, behcet@uw.edu
[‡]Professor, AIAA Member, mesbahi@uw.edu
[§]SPLICE Project Manager, AIAA Associate Fellow

I. Introduction

Real-time optimal powered descent guidance algorithms that enable precision landing are key to maximizing the probability of landing success. Such algorithms allow the vehicle to cope with a more adverse set of model uncertainties, environmental disturbances, and unexpected obstacles as the vehicle approaches the landing site. This increase in robustness can be used to attempt landings in more challenging and scientifically interesting areas, or instead can be reallocated by exchanging excessive propellant mass for an increased payload mass fraction. In recent years, optimal powered descent guidance technology has played a key role in the robust recovery of commercial vertical-takeoff-vertical-landing reusable launch vehicles [1], and is at the forefront of the drive to reduce launch costs for the foreseeable future.

Solving the powered descent guidance problem quickly and reliably is necessary because the vehicle has a limited amount of propellant, and must react quickly to deviations and conditions observed close to the ground. Doing so is challenging due to the nonlinear nature of the dynamics, the non-convex nature of the state and control constraints, and the free-final-time nature of the problem.

Work on powered descent guidance began during the Apollo program, with [2, 3, 4] approaching the problem using optimal control theory and calculus of variations. However, these methods were not incorporated into the Apollo flight code since the polynomial-based guidance methods in use at the time were deemed sufficiently optimal, and were far simpler to design and implement [5]. After the Apollo program, research continued in search of analytical solutions to the 3-degree-of-freedom (DoF) landing problem [6, 7, 8, 9].

In the early 2000s, interest in the powered descent landing problem renewed, this time with a focus on robotic missions to Mars. A number of works using direct methods were published, with [10, 11] using numerical simulations to demonstrate theoretical results on the 3-DoF problem, and with [12, 13, 14] culminating in the use of Pontryagin's maximum principle to losslessly convexify the 3-DoF problem. These works were later generalized in [15, 16, 17, 18, 19, 20, 21], and were demonstrated in a sequence of flight experiments in the early 2010s [22, 23, 24, 25].

In 2015, a dual-quaternion-based approach was proposed in which a 6-DoF line-of-sight constraint was convexified [26, 27]. This method was inherently equipped to handle 6-DoF motion, but relied on piecewise-affine approximations to deal with the nonlinear dynamics. As such, the solution degraded in accuracy with coarser temporal discretizations.

While the convexification techniques listed above represent significant advancements in powered descent guidance, they remain applicable to a relatively limited class of optimal control problems. On the other hand, sequential convex programming (SCP) and successive convexification techniques offer a framework for solving more general non-convex optimal control problems, at the expense of more computational complexity and weakened optimality and convergence guarantees [28, 29, 30, 31, 32, 33, 34, 35]. These methods work by converting the non-convex problem into a sequence of convex subproblems that locally approximate the non-convexities to first-order, and do not carry the approximation to second order as in Sequential Quadratic Programming (SQP) methodologies [36].

The work presented herein uses the successive convexification and discretization framework developed in our earlier work [37, 38, 39]. We also draw heavily from our recent work on a free-final-time 6-DoF powered descent guidance problem with state-triggered constraints [40]. State-triggered constraints represent a novel formulation that, to the best of our knowledge, has not been introduced in the optimal control literature. It bears a close resemblance to complementarity constraints [41, 42], which use continuous variables to model discrete events. Complementarity constraints can be more efficient than mixed-integer approaches (see Section 2.3 in [43]), which suffer from poor complexity and are not well suited for solving the powered descent guidance problem in real-time [44]. However, unlike complementarity constraints that make mutually exclusive decisions between two variables, state-triggered constraints are capable of one-way *if-then* implications. The primary contribution of this paper is the introduction of a continuous formulation for *compound* state-triggered constraints, which generalizes our existing state-triggered constraint formulation.

In subsequent sections, we use the following notation. $\mathbb{R}_+$ and $\mathbb{R}_{++}$ respectively denote non-negative and strictly positive reals, $\mathbb{S}_{++}^n$ denotes an $n \times n$ symmetric positive-definite matrix, $\mathcal{S}^3$ denotes the unit 3-sphere, and the superscript c denotes the complement of a set. The symbols $\cdot$ and $\times$ respectively represent the vector dot and cross products, $a \perp b$ denotes the orthogonality constraint $a \cdot b = 0$, and e_i denotes the unit vector with 1 located at the ith element. The subscripts $\mathcal{I}$ and $\mathcal{B}$ respectively denote quantities expressed in inertial frame ($\mathcal{F}_{\mathcal{I}}$) and body frame ($\mathcal{F}_{\mathcal{B}}$) coordinates. The remainder of this paper is organized as follows. In §II, we discuss state-triggered constraints and the main contribution of this paper; in §III, we give an overview of successive convexification; in §IV, we present simulation results; and in §V, we conclude the paper.

American Institute of Aeronautics and Astronautics

II. Continuous State-Triggered Constraints

This section is organized as follows. In §II.A, we review the state-triggered constraint (STC) formulation introduced in Section II.F of [40]. Specifically, §II.A.1 formally defines STCs, §II.A.2 outlines the continuous formulation of STCs, §II.A.3 provides an example application for STCs, and §II.A.4 discusses alternative formulations. In §II.B, we follow the same outline as in §II.A to introduce an extension of state-triggered constraints called *compound state-triggered constraints*.

II.A. State-Triggered Constraints

To better understand state-triggered constraints, let us first consider the most common type of constraints encountered in the optimal control literature: *temporally-scheduled constraints*. Temporally-scheduled constraints are enforced over predefined time intervals that are determined during problem formulation, and that are not altered while the problem is being solved. For example, such constraints are often be enforced over the entire time horizon of an optimal control problem.

Unlike a temporally-schedule constraint, a state-triggered constraint is enforced only when a state-dependent condition is satisfied, and functions as an *if*-statement conditioned on the solution variable it constrains. An optimal control problem containing a state-triggered constraint determines its solution variable with a simultaneous understanding of how the constraint affects the optimization, and of how the optimization enables or disables the constraint.

II.A.1. Formal Definition

A state-triggered constraint is comprised of a *trigger condition* and a *constraint condition* that are arranged such that satisfaction of the former implies satisfaction of the latter. More formally, the logical implication of a state-triggered constraint is given by

$$g(z) < 0 \implies c(z) = 0, \tag{1}$$

where $z \in \mathbb{R}^{n_z}$ represents the solution variable of the parent optimization problem; $g(\cdot) : \mathbb{R}^{n_z} \to \mathbb{R}$ and $c(\cdot) : \mathbb{R}^{n_z} \to \mathbb{R}$ are piecewise continuously differentiable functions that are termed the *trigger function* and *constraint function*, respectively; and $g(z) < 0$ and $c(z) = 0$ are the aforementioned trigger and constraint conditions.

The feasible set corresponding to (1) is shown in the leftmost pane of Figure 1. The figure illustrates that simultaneously enforcing the state-triggered constraints $g(z) < 0 \implies c(z) = 0$ and $-g(z) < 0 \implies c(z) = 0$ yields the feasible set corresponding to the orthogonality constraint $g(z) \perp c(z)$. Further, intersecting these feasible sets with the set $\{ g, c : g \geq 0 , c \geq 0 \}$ recovers the feasible set of the the well-known complementarity constraint $0 \leq g(z) \perp c(z) \geq 0$ [41]. When viewed in the g-c plane, both orthogonality and complementarity constraints effectively represent bi-directional *if-and-only-if*-statements, whereas state-triggered constraints
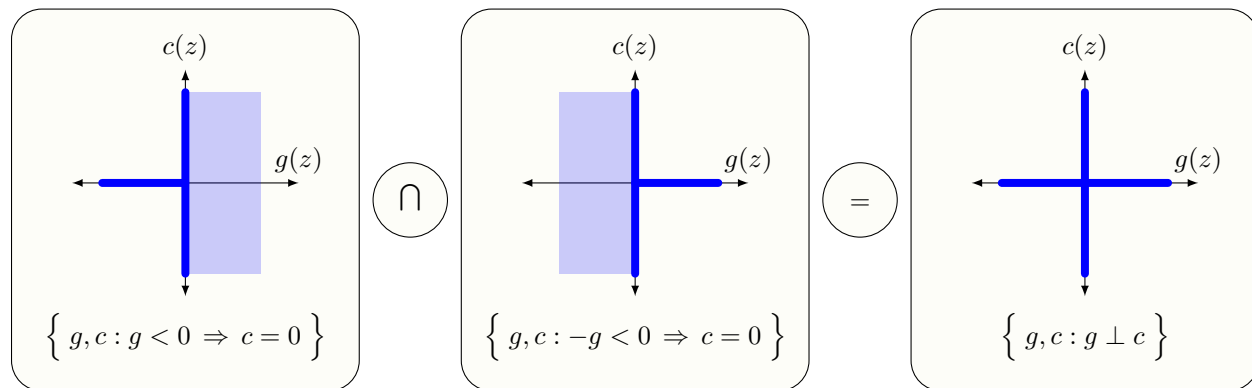

Figure 1: The feasible sets of STCs (left and center panes) and orthogonality constraints (right pane). The feasible sets of the STCs can be intersected in order to recover the feasible set of an orthogonality constraint, and thus capture a broader set of logic in the g-c plane.
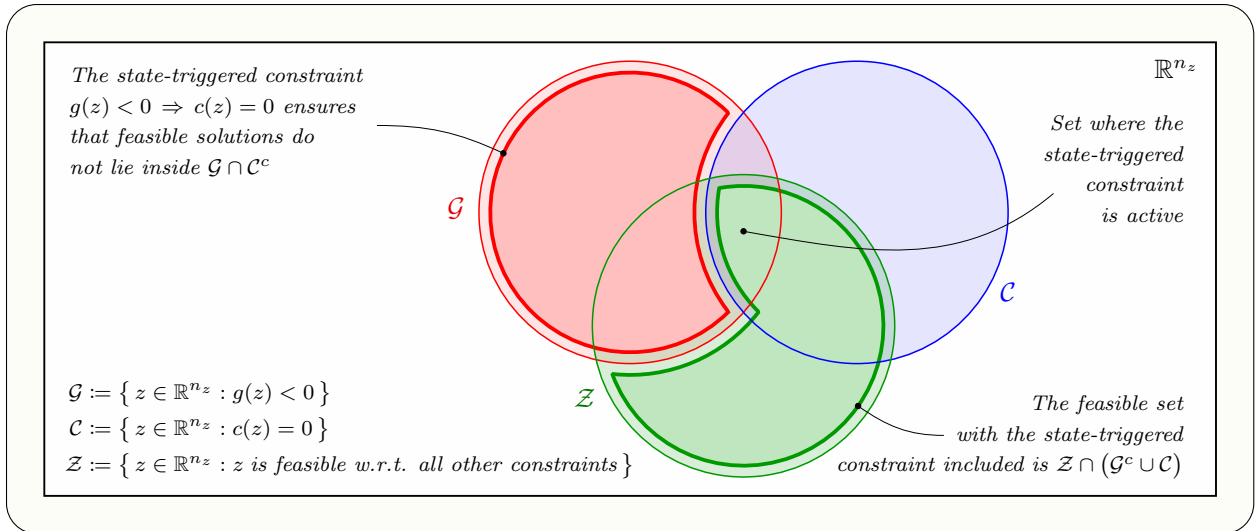
American Institute of Aeronautics and Astronautics

The state-triggered constraint $g(z) < 0 \Rightarrow c(z) = 0$ ensures that feasible solutions do not lie inside $\mathcal{G} \cap \mathcal{C}^c$

$\mathbb{R}^{n_z}$

Set where the state-triggered constraint is active

$\mathcal{G}$

$\mathcal{C}$

$\mathcal{Z}$

$\mathcal{G} := \left\{ z \in \mathbb{R}^{n_z} : g(z) < 0 \right\}$
$\mathcal{C} := \left\{ z \in \mathbb{R}^{n_z} : c(z) = 0 \right\}$
$\mathcal{Z} := \left\{ z \in \mathbb{R}^{n_z} : z \text{ is feasible w.r.t. all other constraints} \right\}$

The feasible set with the state-triggered constraint included is $\mathcal{Z} \cap \left(\mathcal{G}^c \cup \mathcal{C} \right)$

Figure 2: A Venn diagram of the sets of solution variables $z \in \mathbb{R}^{n_z}$ that satisfy the trigger condition ($\mathcal{G}$), constraint condition ($\mathcal{C}$), and all other state and control constraints excluding STCs ($\mathcal{Z}$). State-triggered constraints ensure that feasible solutions satisfy $z \notin \left(\mathcal{G} \cap \mathcal{C}^c \right)$.

emulate *if*-statements. Thus, we argue that the state-triggered constraint formulation represents a more general logical building block for the design of guidance algorithms.

Defining the sets $\mathcal{G}$, $\mathcal{C}$, and $\mathcal{Z}$ as in Figure 2, the statement "$z \in \mathcal{G}$ implies $z \in \mathcal{C}$" is typically taken to mean that $\mathcal{G} \subseteq \mathcal{C}$. However, as seen in Figure 2, $\mathcal{G}$ is not necessarily a subset of $\mathcal{C}$. In fact, if the trigger and constraint functions are defined such that $\mathcal{G} \subseteq \mathcal{C}$, then the corresponding state-triggered constraint is trivially satisfied and serves no practical purpose in the context of the optimization problem. However, if $g(z)$ and $c(z)$ are defined such that $\mathcal{G} \nsubseteq \mathcal{C}$, then the state-triggered constraint *removes* the set $\mathcal{G} \cap \mathcal{C}^c$ from the feasible set, thus ensuring that $\mathcal{G} \subseteq \mathcal{C}$ over the remaining feasible set.

Before proceeding, we note that (1) can equivalently represent a state-triggered *inequality* constraint, provided that z is augmented with a non-negative slack variable and that $g(z)$ and $c(z)$ are modified accordingly [45]. Thus, we proceed with the understanding that the equality in (1) can just as well be replaced with an inequality. Additionally, from De Morgan's Law, satisfaction of (1) implies satisfaction of the contrapositive $c(z) \neq 0 \Rightarrow g(z) \neq 0$. This behavior is exhibited in the simulation examples presented in §IV.

II.A.2. Continuous Formulation

The continuous formulation corresponding to the state-triggered constraint in (1) is given by the following system of equations

$$g(z) + \sigma \geq 0, \tag{2a}$$
$$\sigma \geq 0, \tag{2b}$$
$$\sigma \cdot c(z) = 0, \tag{2c}$$

where $\sigma \in \mathbb{R}_+$ is an auxiliary variable that enables or disables the constraint condition. We refer to (2) as the *original formulation*, and note that it possesses an ambiguity in σ given a $g(z)$ and $c(z)$. This form can be written more compactly and without the ambiguity in σ using the *projected formulation* given by

$$h(z) := -\min\left(g(z), 0\right) \cdot c(z) = 0. \tag{3}$$

Note that the feasible sets of (2) and (3) are identical to that of (1) shown in the leftmost pane of Figure 1. The *continuous state-triggered constraints* (cSTCs) presented above are therefore logically equivalent to the state-triggered constraints detailed in §II.A.1, and can be readily incorporated into a continuous optimization framework. For more details, we refer the reader to Section II.F in [40].
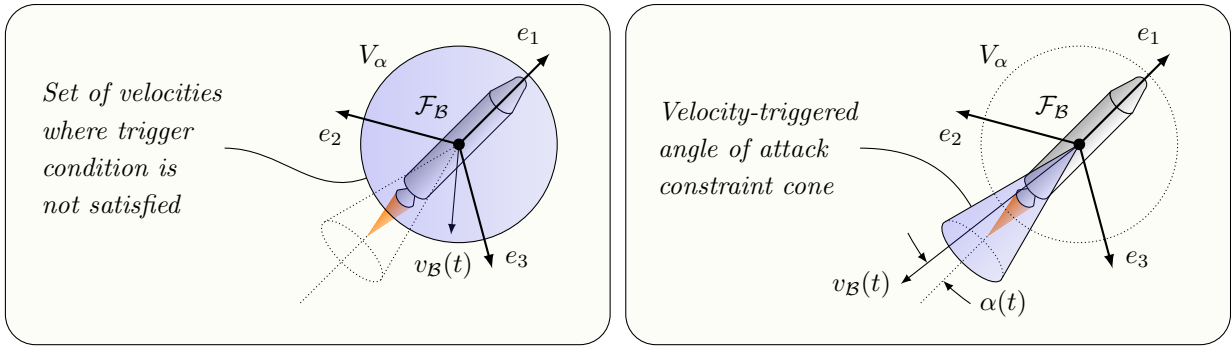
(a) Disabled Constraint Condition   (b) Enabled Constraint Condition

Figure 3: An example powered descent guidance application where a velocity-triggered angle of attack constraint is used to alleviate aerodynamic loads experienced by the vehicle. The set of velocities that do not satisfy the trigger condition (and for which the constraint condition is not enforced) is represented by the blue sphere in (a). When the trigger condition is satisfied, the velocities are constrained to the blue cone shown in (b).

II.A.3. Example Application

The continuous state-triggered constraint formulation outlined in §II.A.2 allows a richer set of constraints to be incorporated into a continuous optimization framework. For example, consider the problem of aerodynamic load alleviation during an atmospheric powered descent guidance maneuver. Imposing a single angle of attack constraint over the entire length of the trajectory may be overly conservative, and may eliminate large swaths of otherwise feasible solutions. Hence, we state the objective of an aerodynamic load alleviation constraint as follows: limit the angle of attack of the vehicle when the dynamic pressure is high, and leave the angle of attack unconstrained when the dynamic pressure is negligible.

Assuming constant atmospheric density, this constraint may be interpreted as a velocity-triggered angle of attack constraint, and is stated formally as the following state-triggered constraint

$$\|v_{\mathcal{B}}(t)\|_2 > V_\alpha \ \Rightarrow \ -e_1 \cdot v_{\mathcal{B}}(t) \geq \cos \alpha_{max} \|v_{\mathcal{B}}(t)\|_2 , \tag{4}$$

where $V_\alpha \in \mathbb{R}_{++}$ is a speed above which the angle of attack is limited to $\alpha(t) \in [0 \, , \alpha_{max}]$, and $v_{\mathcal{B}}(t) \in \mathbb{R}^3$ is the velocity expressed in $\mathcal{F}_{\mathcal{B}}$ coordinates. Figure 3a shows the set of feasible velocities when the trigger condition is not satisfied and the constraint condition is disabled. Figure 3b shows the set of feasible velocities when the trigger condition is satisfied and the constraint condition is enabled. Using the projected formulation in (3), the STC in (4) is converted into the following cSTC:

$$h_\alpha \big(v_{\mathcal{I}}(t), q_{\mathcal{B}\leftarrow\mathcal{I}}(t)\big) := - \min \Big(g_\alpha \big(v_{\mathcal{I}}(t)\big), 0 \Big) \cdot c_\alpha \big(v_{\mathcal{I}}(t), q_{\mathcal{B}\leftarrow\mathcal{I}}(t)\big) \leq 0, \tag{5a}$$

$$g_\alpha \big(v_{\mathcal{I}}(t)\big) := V_\alpha - \|v_{\mathcal{I}}(t)\|_2 , \tag{5b}$$

$$c_\alpha \big(v_{\mathcal{I}}(t), q_{\mathcal{B}\leftarrow\mathcal{I}}(t)\big) := \cos \alpha_{max} \|v_{\mathcal{I}}(t)\|_2 + e_1 \cdot C_{\mathcal{B}\leftarrow\mathcal{I}}\big(q_{\mathcal{B}\leftarrow\mathcal{I}}(t)\big) v_{\mathcal{I}}(t), \tag{5c}$$

where $v_{\mathcal{I}}(t) \in \mathbb{R}^3$ is the inertial velocity, $q_{\mathcal{B}\leftarrow\mathcal{I}}(t) \in \mathcal{S}^3 \subset \mathbb{R}^4$ is the unit quaternion representing the transformation from $\mathcal{F}_{\mathcal{I}}$ to $\mathcal{F}_{\mathcal{B}}$, and $C_{\mathcal{B}\leftarrow\mathcal{I}}(t) := C_{\mathcal{B}\leftarrow\mathcal{I}}\big(q_{\mathcal{B}\leftarrow\mathcal{I}}(t)\big)$ is the direction cosine matrix associated with $q_{\mathcal{B}\leftarrow\mathcal{I}}(t)$.

II.A.4. Alternative Formulations

We now consider a few alternative formulations that can be used in lieu of the continuous state-triggered constraint formulation presented above. Since mixed-integer methodologies were deemed not conducive to solving the powered descent guidance problem in real-time, we do not consider them further. To present these alternatives, the following discussion assumes a powered descent guidance problem that includes the state-triggered constraint formalized in (4).

Alternative 1 - Naive Implementation: The first implementation we consider is a naive implementation that uses temporally-scheduled constraints. Due to its simplicity, this approach is arguably the most obvious implementation one might pursue when trying to capture the logical implication in (4). In this approach, the optimization problem is first solved without (4), and the resulting solution is used to determine the set of times $\mathcal{T}_g$ over which the trigger condition $g_\alpha(\cdot) < 0$ is satisfied. The problem is then solved a second time, this time with the constraint condition $c_\alpha(\cdot, \cdot) \leq 0$ enforced for all $t \in \mathcal{T}_g$. The second solution is guaranteed to satisfy the constraint condition for all $t \in \mathcal{T}_g$. However, this solution may have time instances $t \in \mathcal{T}_g^- \subseteq \mathcal{T}_g$ for which the trigger condition is *no longer* satisfied, and where the constraint condition is needlessly enforced. This solution may also have time instances $t \in \mathcal{T}_g^+ \subseteq \mathcal{T}_g^c$ where the trigger condition *is* now satisfied, and for which the implication in (4) is violated. Thus, the inclusion of the constraint condition in the second attempt may alter the solution such that the triggering profile assumed at the onset is invalidated. Clearly, the second solution may be used to redefine the set $\mathcal{T}_g$, and the procedure can be repeated. However, since this approach obscures the relationship between the trigger and constraint conditions from the optimization, this behavior may persist indefinitely. Continuous state-triggered constraints avoid this issue by exposing the relationship between the trigger and constraint conditions to the optimization algorithm.

Alternative 2 - Nonlinear Implementation: The second implementation we consider is one where (4) is replaced with $-e_1 \cdot v_\mathcal{B}(t) / \|v_\mathcal{B}(t)\|_2 \geq \cos f_\alpha(t)$, where $f_\alpha(t) \coloneqq f_\alpha\big(\|v_\mathcal{B}(t)\|_2\big)$ is a nonlinear scalar-valued function that relates the maximum allowable angle of attack to the speed of the vehicle. Unlike the formulation used in (5) which switches the constraint condition on and off in a binary fashion, this approach adjusts the constraint condition smoothly according to $f_\alpha(\cdot)$. That is, instead of enforcing an angle of attack limit at high speeds and disabling it at low speeds, this implementation *always* enforces an angle of attack limit that varies smoothly as a function of speed. The ability to specify a smooth transition profile to modulate the constraint condition may be advantageous in certain cases. For example, it may be desirable to leave the angle of attack unconstrained at low speeds, and to restrict it to a maximum of $3°$ and $5°$ at intermediate and high speeds, respectively. Using the STC framework, such a profile would either have to be conservatively approximated, or multiple cSTCs would have to be used. However, the nonlinear implementation has two disadvantages when compared to the cSTC approach. First, it is less intuitive since the relationship between the angle of attack and speed must be embedded in $f_\alpha(t)$. This disadvantage can become particularly accentuated when the geometry of the constraint is multi-dimensional and difficult to visualize. Second, even if a satisfactory function $f_\alpha(t)$ is obtained, it may be difficult to ensure it has sufficiently good numerical properties (e.g. optimization problems with high-order polynomials may be difficult to solve reliably).

Alternative 3 - Multi-Phase Implementation: The third implementation we consider splits the problem into two temporal phases: the first phase includes an angle of attack constraint but no velocity constraint, and the second phase includes a velocity constraint but no angle of attack constraint. The terminal condition of the first phase is equated to the initial condition of the second, and both phases are solved *simultaneously* as one optimal control problem. This multi-phase optimization approach ensures satisfaction of (4) by construction, and is well suited for applications where the quantity and temporal ordering of the phases is known a priori. For example, this approach is ideal for optimizing the ascent trajectories of multi-staged rockets. However, since this approach presupposes the quantity and order of the phases, it may introduce unnecessary conservatism. In contrast, cSTCs are able to rearrange and introduce additional phases to achieve feasibility or improve optimality. An example of this behavior is presented in §IV.

II.B. Compound State-Triggered Constraints

The state-triggered constraint formulation presented in §II.A assumes scalar-valued trigger and constraint functions. This section introduces the primary contribution of this paper, namely a novel continuous formulation for *compound* state-triggered constraints.

Compound state-triggered constraints are defined using *vector-valued* trigger and constraint functions whose elements are composed into compound conditions using Boolean logic operations. The compound formulation is a generalization of the formulation given in (1), and offers a straightforward procedure whereby simple trigger and constraint function elements can be composed into more complicated trigger and constraint conditions. We proceed as in §II.A by defining compound state-triggered constraints, presenting a continuous formulation, providing example applications, and considering alternative formulations.

II.B.1. Formal Definition

The scalar state-triggered constraint defined in (1) may be generalized to one of the following compound state-triggered constraints

$$\textit{Or}\text{-Trigger with }\textit{Or}\text{-Constraint:} \qquad \bigvee_{i=1}^{n_g}\Big(g_i(z)<0\Big) \quad\Rightarrow\quad \bigvee_{i=1}^{n_c}\Big(c_i(z)=0\Big), \tag{6a}$$

$$\textit{Or}\text{-Trigger with }\textit{And}\text{-Constraint:} \qquad \bigvee_{i=1}^{n_g}\Big(g_i(z)<0\Big) \quad\Rightarrow\quad \bigwedge_{i=1}^{n_c}\Big(c_i(z)=0\Big), \tag{6b}$$

$$\textit{And}\text{-Trigger with }\textit{Or}\text{-Constraint:} \qquad \bigwedge_{i=1}^{n_g}\Big(g_i(z)<0\Big) \quad\Rightarrow\quad \bigvee_{i=1}^{n_c}\Big(c_i(z)=0\Big), \tag{6c}$$

$$\textit{And}\text{-Trigger with }\textit{And}\text{-Constraint:} \qquad \bigwedge_{i=1}^{n_g}\Big(g_i(z)<0\Big) \quad\Rightarrow\quad \bigwedge_{i=1}^{n_c}\Big(c_i(z)=0\Big), \tag{6d}$$

where $g(\cdot):\mathbb{R}^{n_z}\to\mathbb{R}^{n_g}$ and $c(\cdot):\mathbb{R}^{n_z}\to\mathbb{R}^{n_c}$ are redefined as *vector-valued* trigger and constraint functions, and where $g_i(z)$ and $c_i(z)$ represent the ith entries of the vectors $g(z)$ and $c(z)$, respectively. When *at least one* element of the compound trigger condition is satisfied, (6a) implies that *at least one* element of the compound constraint condition is enforced, whereas (6b) implies that *all* elements of the compound constraint condition are enforced. When *all* elements of the compound trigger condition are satisfied, (6c) implies that *at least one* element of the compound constraint condition is enforced, whereas (6d) implies that *all* elements of the compound constraint condition are enforced. Note that the compound generalizations in (6) recover the scalar definition in (1) when $n_g = n_c = 1$.

II.B.2. Continuous Formulation

The projected continuous formulations corresponding to the compound state-triggered constraints defined in (6) are tabulated in Table 1. Note that $\hat{\sigma}_i(z) := -\min\big(g_i(z),0\big) \geq 0$ for all $i \in \{1,\dots,n_g\}$, and that $h_{\wedge\wedge}(\cdot) = 0$ and $h_{\vee\wedge}(\cdot) = 0$ may be implemented as standalone equality constraints *only if* $c_i(z) \geq 0$ holds for all $i \in \{1,\dots,n_c\}$. However, when element-wise non-negativity of $c(z)$ does not hold, $h_{\wedge\wedge}(\cdot)$ and $h_{\vee\wedge}(\cdot)$ must be implemented as n_c distinct equality constraints respectively given by $c_j(z)\cdot\prod_{i=1}^{n_g}\hat{\sigma}_i(z) = 0$ and $c_j(z)\cdot\sum_{i=1}^{n_g}\hat{\sigma}_i(z) = 0$ for all $j \in \{1,\dots,n_c\}$. The standard continuous formulation can be obtained from Table 1 by replacing $\hat{\sigma}_i(z)$ with elements of a non-negative vector-valued $\sigma \in \mathbb{R}_+^{n_g}$ that adheres to (2a) and (2b) for each $i \in \{1,\dots,n_g\}$.

Table 1: Summary of Compound Continuous State-Triggered Constraints

Compound Trigger Condition	Compound Constraint Condition	Continuous Implementation $\hat{\sigma}_i(z) := -\min\big(g_i(z),0\big)$
$\displaystyle\bigwedge_{i=1}^{n_g}\Big(g_i(z)<0\Big)$	$\displaystyle\bigwedge_{i=1}^{n_c}\Big(c_i(z)=0\Big)$	$\displaystyle h_{\wedge\wedge}(z) := \left[\prod_{i=1}^{n_g}\hat{\sigma}_i(z)\right]\cdot\left[\sum_{i=1}^{n_c}c_i(z)\right] = 0,\ c(z)\geq 0$
$\displaystyle\bigwedge_{i=1}^{n_g}\Big(g_i(z)<0\Big)$	$\displaystyle\bigvee_{i=1}^{n_c}\Big(c_i(z)=0\Big)$	$\displaystyle h_{\wedge\vee}(z) := \left[\prod_{i=1}^{n_g}\hat{\sigma}_i(z)\right]\cdot\left[\prod_{i=1}^{n_c}c_i(z)\right] = 0$
$\displaystyle\bigvee_{i=1}^{n_g}\Big(g_i(z)<0\Big)$	$\displaystyle\bigwedge_{i=1}^{n_c}\Big(c_i(z)=0\Big)$	$\displaystyle h_{\vee\wedge}(z) := \left[\sum_{i=1}^{n_g}\hat{\sigma}_i(z)\right]\cdot\left[\sum_{i=1}^{n_c}c_i(z)\right] = 0,\ c(z)\geq 0$
$\displaystyle\bigvee_{i=1}^{n_g}\Big(g_i(z)<0\Big)$	$\displaystyle\bigvee_{i=1}^{n_c}\Big(c_i(z)=0\Big)$	$\displaystyle h_{\vee\vee}(z) := \left[\sum_{i=1}^{n_g}\hat{\sigma}_i(z)\right]\cdot\left[\prod_{i=1}^{n_c}c_i(z)\right] = 0$

American Institute of Aeronautics and Astronautics

(a) Landing in a Crater or Canyon
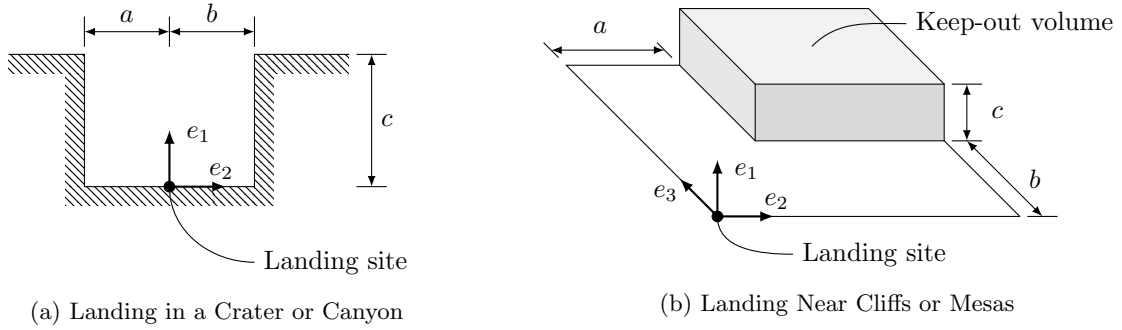
(b) Landing Near Cliffs or Mesas

Figure 4: An example powered descent guidance application where compound state-triggered constraints are used to avoid collisions with geological formations near the landing site.

II.B.3. Example Application

To motivate the use of compound STCs, consider a scenario where the objective is to land in the vicinity of large geologic formations such as the craters, canyons, cliffs, or mesas. Figure 4a illustrates a scenario where the landing site is located at the bottom of a crater or canyon. Using the parameters defined in the figure, the collision avoidance constraint is given by the following *or*-triggered compound STC

$$\big(e_2 \cdot r_{\mathcal{I}}(t) < -a\big) \vee \big(b < e_2 \cdot r_{\mathcal{I}}(t)\big) \;\Rightarrow\; c - e_1 \cdot r_{\mathcal{I}}(t) \leq 0,$$

where $r_{\mathcal{I}}(t) \in \mathbb{R}^3$ is the vehicle's inertial position vector. The corresponding compound cSTC is given by

$$h_\vee\big(r_{\mathcal{I}}(t)\big) := \left[\sum_{i=1}^{2} -\min\Big(g_{\vee,i}\big(r_{\mathcal{I}}(t)\big),0\Big)\right] \cdot c_\vee\big(r_{\mathcal{I}}(t)\big) \leq 0,$$

$$g_{\vee,1}\big(r_{\mathcal{I}}(t)\big) := a + e_2 \cdot r_{\mathcal{I}}(t),$$

$$g_{\vee,2}\big(r_{\mathcal{I}}(t)\big) := b - e_2 \cdot r_{\mathcal{I}}(t),$$

$$c_\vee\big(r_{\mathcal{I}}(t)\big) := c - e_1 \cdot r_{\mathcal{I}}(t).$$

Figure 4b illustrates a similar but more challenging scenario, where a keep-out volume is used to steer the vehicle away from geological formations such as cliffs or mesas. Following the procedure outlined above, we obtain a continuous formulation of the collision avoidance constraints using the following *and*-triggered compound cSTC

$$h_\wedge\big(r_{\mathcal{I}}(t)\big) := \left[\prod_{i=1}^{2} -\min\Big(g_{\wedge,i}\big(r_{\mathcal{I}}(t)\big),0\Big)\right] \cdot c_\wedge\big(r_{\mathcal{I}}(t)\big) \leq 0, \tag{8a}$$

$$g_{\wedge,1}\big(r_{\mathcal{I}}(t)\big) := a - e_2 \cdot r_{\mathcal{I}}(t), \tag{8b}$$

$$g_{\wedge,2}\big(r_{\mathcal{I}}(t)\big) := b - e_3 \cdot r_{\mathcal{I}}(t), \tag{8c}$$

$$c_\wedge\big(r_{\mathcal{I}}(t)\big) := c - e_1 \cdot r_{\mathcal{I}}(t). \tag{8d}$$

II.B.4. Alternative Formulations

The scenario described in §II.B.3 can instead be addressed using the glide-slope cone constraint formulated in [14]. However, enforcing a sufficiently steep constraint over the entire trajectory may be overly conservative, and may prohibitively restrict the set of feasible initial conditions (e.g., Mars' Valles Marineris reaches depths of up to 7 km). Alternatively, enforcing the constraint over only a terminal time interval may allow the trajectory to violate the physical constraints at times prior to said interval. While this approach may work in certain cases, the heuristics it employs may fall short for applications that have a large set of possible initial conditions.

Alternatively, this scenario can be addressed using the multi-phase optimization approach presented in [46], which is similar to *Alternative 3* discussed in §II.A.4. The multi-phase optimization approach is well suited for this example application since the quantity and ordering of the phases is known a priori (provided that the altitude decreases monotonically). Nevertheless, we argue that compound state-triggered constraints offer a simple way to formulate complex keep-out volumes in the scenario described above.

III. Problem Statement & Convex Formulation

In this section, we outline a problem statement for a non-convex 6-DoF powered descent guidance problem, and provide a high-level description of the successive convexification procedure used to solve it. The content of this section is drawn from Sections II and III of [40], and is provided as context for §IV.

III.A. Non-Convex Problem Statement

We consider the minimum-time 6-DoF powered descent guidance problem given in Problem 1, which is comprised of nonlinear dynamics and non-convex state and control constraints. This problem formulation includes the baseline problem formulation, free-ignition-time modification, and ellipsoidal aerodynamics model detailed in Sections II.A-II.E of [40]. Additionally, we include the velocity-triggered angle of attack constraint discussed in §II.A.3, and the collision avoidance constraint formulated in §II.B.3.

Our problem formulation consists of three time epochs: *initial time* t_{in}, *ignition time* t_{ig}, and *final time* t_f. These epochs are defined such that $t_{in} \le t_{ig} < t_f$. We call the time interval $t \in [t_{in}, t_{ig}]$ the *coast phase*, and the interval $t \in [t_{ig}, t_f]$ the *burn phase*. During the coast phase, the engine is off, and the vehicle state evolves passively along a predetermined trajectory (as is discussed shortly). We denote the *coast time* by $t_c := t_{ig} - t_{in}$, and the maximum allowable coast time by $t_{c,max}$. During the burn phase, the engine is on, and the vehicle actively maneuvers to achieve its landing objective. We denote the *burn time* by $t_b := t_f - t_{ig}$.

The states of the problem are the mass $m(t) \in \mathbb{R}_{++}$, angular velocity $\omega_{\mathcal{B}}(t) \in \mathbb{R}^3$, and the position $r_{\mathcal{I}}(t)$, velocity $v_{\mathcal{I}}(t)$, and unit quaternion $q_{\mathcal{B} \leftarrow \mathcal{I}}(t)$ introduced in §II.A.3 and §II.B.3. The control of the problem is the thrust vector $T_{\mathcal{B}}(t) \in \mathbb{R}^3$. These are summarized as the state vector

$$x(t) := \begin{bmatrix} m(t) \ r_{\mathcal{I}}^T(t) \ v_{\mathcal{I}}^T(t) \ q_{\mathcal{B} \leftarrow \mathcal{I}}^T(t) \ \omega_{\mathcal{B}}^T(t) \end{bmatrix}^T \in \mathbb{R}^{14},$$

and the control vector $u(t) := T_{\mathcal{B}}(t) \in \mathbb{R}^3$. Since this is a free-final-time minimum-time problem with a free-ignition-time modification, the cost function of the problem is given by $J(z) := t_b$, and the solution variable is given by

$$z(t) := \begin{bmatrix} t_c \ t_b \ x^T(t) \ u^T(t) \end{bmatrix}^T \in \mathbb{R}^{19}.$$

At the ignition time epoch, the following boundary conditions are enforced

$$m(t_{ig}) = m_{ig}, \quad r_{\mathcal{I}}(t_{ig}) = p_{r,ig}(t_c), \quad v_{\mathcal{I}}(t_{ig}) = p_{v,ig}(t_c), \quad \omega_{\mathcal{B}}(t_{ig}) = 0,$$

where, due the free-ignition-time modification, the ignition time position and velocity are constrained to lie on curves described by the polynomials $p_{r,ig} : \mathbb{R} \to \mathbb{R}^3$ and $p_{v,ig} : \mathbb{R} \to \mathbb{R}^3$. Although our problem formulation considers aerodynamic effects, we assume that $p_{r,ig}(\cdot)$ and $p_{v,ig}(\cdot)$ describe an aerodynamics-free free fall trajectory given by

$$p_{r,ig}(t_c) := r_{\mathcal{I},in} + v_{\mathcal{I},in} t_c + \frac{1}{2} g_{\mathcal{I}} t_c^2, \quad p_{v,ig}(t_c) := v_{\mathcal{I},in} + g_{\mathcal{I}} t_c,$$

where $r_{\mathcal{I},in}$ and $v_{\mathcal{I},in}$ are prescribed initial-time position and velocity vectors, and $g_{\mathcal{I}} \in \mathbb{R}^3$ is the (constant) gravity vector. We note, however, that $p_{r,ig}(\cdot)$ and $p_{v,ig}(\cdot)$ can be formulated to reflect higher order effects such as aerodynamics. At the final time epoch, the following boundary conditions are enforced

$$r_{\mathcal{I}}(t_f) = 0, \quad v_{\mathcal{I}}(t_f) = 0, \quad q_{\mathcal{B} \leftarrow \mathcal{I}}(t_f) = q_{id}, \quad \omega_{\mathcal{B}}(t_f) = 0,$$

where $q_{id} := [1 \ 0 \ 0 \ 0]^T$ denotes the identity quaternion.

The mass state of the vehicle evolves according to the affine (in thrust) mass depletion dynamics used in [37], while the translational and attitude states evolve according to 6-DoF rigid body dynamics. The former depends on the mass depletion coefficients $\alpha_{\dot{m}} \in \mathbb{R}_{++}$ and $\beta_{\dot{m}} \in \mathbb{R}_+$ that are functions of the specific impulse, nozzle exit area, and atmospheric back pressure. The latter is affected by the commanded thrust $T_{\mathcal{B}}(t)$ and the aerodynamic force $A_{\mathcal{B}}(t) \in \mathbb{R}^3$, which apply torque to the vehicle with moment arms $r_{T,\mathcal{B}} \in \mathbb{R}^3$ and $r_{cp,\mathcal{B}} \in \mathbb{R}^3$, respectively. The aerodynamic force is defined by the ellipsoidal aerodynamic model introduced in Section II.E of [40]. Additionally, we use $\Omega(\cdot)$ to denote the 4×4 skew-symmetric matrix associated with the quaternion kinematics, and $J_{\mathcal{B}} \in \mathbb{S}_{++}^3$ to denote the vehicle's body-fixed constant inertia.

The problem formulation includes state inequality constraints that keep the mass above a dry mass $m_{dry} \in \mathbb{R}_{++}$, the position inside a glide slope cone with a half-angle of $\gamma_{gs} \in [0°, 90°)$, the tilt angle less

Problem 1. *Non-Convex Optimal Control Problem*

Cost Function:

$$\underset{t_c,\, t_b,\, T_{\mathcal{B}}(t)}{\text{minimize}} \; t_b$$

$$\text{s.t.} \quad t_c \in [0\,,t_{c,max}]$$

Boundary Conditions:

$$
\begin{aligned}
m(t_{ig}) &= m_{ig} & q_{\mathcal{B}\leftarrow\mathcal{I}}(t_f) &= q_{id} \\
r_{\mathcal{I}}(t_{ig}) &= p_{r,ig}(t_c) & r_{\mathcal{I}}(t_f) &= 0 \\
v_{\mathcal{I}}(t_{ig}) &= p_{v,ig}(t_c) & v_{\mathcal{I}}(t_f) &= 0 \\
\omega_{\mathcal{B}}(t_{ig}) &= 0 & \omega_{\mathcal{B}}(t_f) &= 0
\end{aligned}
$$

Dynamics:

$$
\begin{aligned}
\dot{m}(t) &= -\alpha_{\dot{m}} \left\| T_{\mathcal{B}}(t) \right\|_2 - \beta_{\dot{m}} \\
\dot{r}_{\mathcal{I}}(t) &= v_{\mathcal{I}}(t) \\
\dot{v}_{\mathcal{I}}(t) &= \frac{1}{m(t)} C_{\mathcal{I}\leftarrow\mathcal{B}}(t)\big(T_{\mathcal{B}}(t) + A_{\mathcal{B}}(t)\big) + g_{\mathcal{I}} \\
\dot{q}_{\mathcal{B}\leftarrow\mathcal{I}}(t) &= \frac{1}{2}\Omega\big(\omega_{\mathcal{B}}(t)\big) q_{\mathcal{B}\leftarrow\mathcal{I}}(t) \\
J_{\mathcal{B}}\dot{\omega}_{\mathcal{B}}(t) &= r_{T,\mathcal{B}} \times T_{\mathcal{B}}(t) + r_{cp,\mathcal{B}} \times A_{\mathcal{B}}(t) - \omega_{\mathcal{B}}(t) \times J_{\mathcal{B}}\omega_{\mathcal{B}}(t)
\end{aligned}
$$

State Constraints:

$$
\begin{aligned}
m_{dry} &\le m(t) \\
\tan\gamma_{gs} \left\| H_\gamma r_{\mathcal{I}}(t) \right\|_2 &\le e_1 \cdot r_{\mathcal{I}}(t) \\
\cos\theta_{max} &\le 1 - 2 \left\| H_\theta q_{\mathcal{B}\leftarrow\mathcal{I}}(t) \right\|_2 \\
\left\| \omega_{\mathcal{B}}(t) \right\|_2 &\le \omega_{max}
\end{aligned}
$$

Control Constraints:

$$
\begin{aligned}
0 < T_{min} &\le \left\| T_{\mathcal{B}}(t) \right\|_2 \le T_{max} \\
\cos\delta_{max} \left\| T_{\mathcal{B}}(t) \right\|_2 &\le e_3 \cdot T_{\mathcal{B}}(t)
\end{aligned}
$$

State-Triggered Constraints:

$$h_\alpha\big(v_{\mathcal{I}}(t), q_{\mathcal{B}\leftarrow\mathcal{I}}(t)\big) \le 0 \qquad\qquad\qquad \text{See (5)}$$
$$h_\wedge\big(r_{\mathcal{I}}(t)\big) \le 0 \qquad\qquad\qquad \text{See (8)}$$

than $\theta_{max} \in (0°\,,90°]$, and the angular velocity magnitude less than $\omega_{max} \in \mathbb{R}_{++}$. The matrices in the glide slope and tilt constraints are defined as $H_\gamma := [e_2 \; e_3]^T \in \mathbb{R}^{2\times3}$ and $H_\theta := [e_3 \; e_4]^T \in \mathbb{R}^{2\times4}$, respectively. The formulation also includes two control constraints. The first constraint limits the thrust magnitude to a minimum value of T_{min} and a maximum value of T_{max}, while the second ensures that the gimbal angle does not exceed $\delta_{max} \in (0°\,,90°)$. We refer the reader to [40] for more details.

III.B. Convex Formulation

Successive convexification is an iterative solution methodology wherein the original non-convex problem is converted into a sequence of manageable convex approximations, or subproblems. These convex subproblems can be solved using fast and reliable interior point method algorithms [47, 22], thereby bolstering the real-time capabilities of the successive convexification methodology. While the field of solving convex optimization problems is quite mature, methods that convert a non-convex optimal control problem into a numerically tractable parameter optimization problem are more varied.

Section III in [40] outlines a procedure whereby the original free-final-time nonlinear continuous-time discrete-time optimal control problem (i.e. Problem 1) is converted into a fixed-final-time linear-time-varying optimal control problem. This procedure consists of three analytical steps that *normalize*, *linearize*, and *discretize* the problem. The result of these three steps is a numerically tractable second-order cone programming
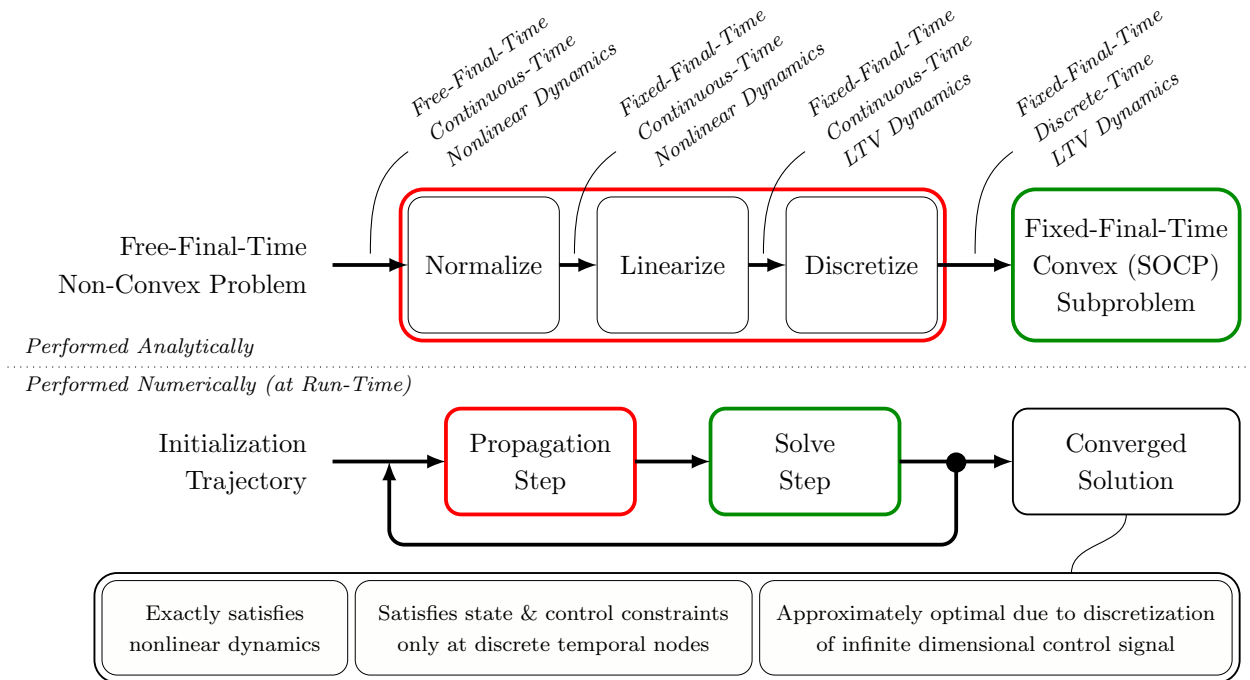
Figure 5: High-level description of the successive convexification algorithm. The top half of the figure shows the three analytical steps that comprise the *propagation step*. The propagation step generates a fixed-final-time second-order cone programming subproblem that is solved during the *solve step*. The propagation and solve steps are executed numerically at run-time. The converged solution exhibits the three properties detailed in the text boxes at the bottom.

subproblem that locally approximates Problem 1. Due to the linearization step, this local approximation is *guaranteed* to be convex. This procedure is illustrated in Figure 5.

In practice, successive convexification works as follows. The iterative process is initialized with a user-specified reference trajectory. During the first iteration, the three analytical steps mentioned above are executed as part of the *propagation step*. The propagation step computes a set of matrices that characterize the approximated discrete-time linear-time-varying dynamics and linearized state and control constraints associated with the convex subproblem. The resulting second-order cone programming subproblem is subsequently solved in the *solve step*, and its solution is used as the reference trajectory for the subsequent successive convexification iteration.

Since the subproblem is an approximation of the non-convex elements of Problem 1 obtained through linearization, one might expect the process described above to generate approximate solutions. While this is true in some sense, we emphasize that a converged solution of the process shown in Figure 5 has the following three properties: (i) the solution exactly satisfies the nonlinear dynamics, (ii) the solution satisfies the state and control constraints at a finite number of temporal nodes defined during the discretization step, and (iii) the solution is approximately (locally) optimal. The third property is due to the first-order-hold interpolation used in [40] to represent the infinite dimensional control signal as a finite-dimensional linearly interpolated signal. The reader is referred to Section III of [40] for more details on the propagation step and on the successive convexification algorithm.

Convergence guarantees for a successive convexification algorithm were presented in [31, 33]. The algorithm presented in these works employed a virtual control term, hard trust regions, and an exact penalty method, and was guaranteed to converge globally to a (not necessarily feasible) stationary point. The result stated that a converged solution with zero virtual control was a local optimum of the original problem. In contrast, the algorithm used herein and in [40] uses soft trust regions that are augmented to the cost, and does not yet have convergence results. However, it is similar in structure to the algorithm presented in [31, 33], and has been observed to work well in practice. Moreover, the results obtained in [40] provide preliminary evidence of the real-time capabilities of our methodology.

American Institute of Aeronautics and Astronautics

IV. Numerical Results

In this section we present simulation results that demonstrate the utility of state-triggered and compound state-triggered constraints. Section IV.A presents an example that uses the velocity-triggered angle of attack constraint discussed in §II.A.3 to alleviate aerodynamic loading, while §IV.B presents an example that uses the collision avoidance compound state-triggered constraint discussed in §II.B.3. Tables 2 and 3 provide the problem parameters used in the simulations. These values correspond to the parameters in Problem 1 when they can be matched. Otherwise, these values reference parameters defined in Problem 1, Problem 2, and Algorithm 1 in [40]. The results presented here were obtained using the spherical aerodynamics model detailed Section II.E of [40], with the quantity $\rho S_A C_A = 0.2$. The maximum coast times and the initial positions and velocities are specified individually in §IV.A and §IV.B. The remaining boundary conditions are given in Problem 1 in §III.A. All quantities are expressed in non-dimensionalized mass (U_M), length (U_L), and time (U_T) units.

Table 2: Traj. Parameters

Param.	Value	Units
$g_{\mathcal{I}}$	$-e_1$	U_L/U_T^2
α_{max}	10.0	°
V_α	2.5	U_L/U_T
m_{ig}	4.0	U_M
m_{dry}	2.0	U_M
θ_{max}	90.0	°
ω_{max}	90.0	$°/U_T$
γ_{gs}	20.0	°
δ_{max}	20.0	°
T_{min}	1.0	$U_M \cdot U_L/U_T^2$
T_{max}	8.0	$U_M \cdot U_L/U_T^2$

Table 3: Vehicle & Algorithm Parameters

Param.	Value	Units
$r_{cp,\mathcal{B}}$	$0_{3\times1}$	U_L
$r_{T,\mathcal{B}}$	$-0.01 \cdot e_1$	U_L
$\alpha_{\dot{m}}$	0.05	U_T/U_L
$\beta_{\dot{m}}$	0.02	U_M/U_T
$J_{\mathcal{B}}$	$0.01 \cdot \mathrm{diag}\left([0.1\ 1\ 1]\right)$	$U_M \cdot U_L^2$
K	30	-
w_ν	1e+3	-
W_{tr}	$\mathrm{diag}\left([\text{3e-6} \cdot 1_{2\times1}^T\ \text{1e-3} \cdot 1_{17\times1}^T]\right)$	-
ϵ_{vc}	1e-4	-
ϵ_{tr}	1e-1	-
$\bar{s}$	10.0	-

IV.A. Example 1: State-Triggered Constraints

This example illustrates the velocity-triggered angle of attack constraint described in §II.A.3. Figure 6 shows a powered descent guidance trajectory that begins on the free fall trajectory shown in black. In this case, the vehicle *is not* subjected to a state-triggered constraint. The initial position is given by $r_{\mathcal{I},in} = [14\ 16\ 0]^T\ U_L$, and the initial velocity by $v_{\mathcal{I},in} = [0\ -3.57\ 1.79]^T\ U_L/U_T$. The vehicle is permitted to coast down this path for a maximum duration of $t_{c,max} = 2.0$, at which point the coast phase ends, the engine ignites, and the burn phase commences. The vehicle can be seen maneuvering away from the original free fall trajectory using a bang-bang style thrust profile, landing at the landing site located at the origin. Figures 6a and 6b show the same trajectory from different angles. Specifically, the horizontal projection shown at the bottom of the figures clearly shows that the vehicle starts moving away from the free fall trajectory as soon as the burn phase beings.

Figure 7 shows the same scenario but with the velocity-triggered angle of attack constraint enabled. In this case, the vehicle can be seen to travel along the free fall trajectory for a period of time before making a sudden turn towards the landing pad. This behavior is due to the state-triggered constraint, which forces the vehicle to apply thrust to slow down prior to pulling a high angle of attack. This behavior is shown more clearly in Figure 8 which compares the speed and angle of attack time histories of the two cases.

In Figure 8a, the angle of attack clearly violates the prescribed limit of $10°$ when the speed is above the trigger velocity of $2.5\ U_L/U_T$. However, in Figure 8b, the behavior is markedly different. First of all, the angle of attack begins below the specified angle of attack limit. The vehicle is seen shedding speed between times $0\ U_T$ and $1\ U_T$ in an attempt to deactivate the trigger condition. Once it does so, the constraint condition is no longer enforced, and the angle of attack is allowed to increase to close to $90°$. Enforcement of the contrapositive can be seen between times $2\ U_T$ and $4\ U_T$, where since the angle of attack is so large, the

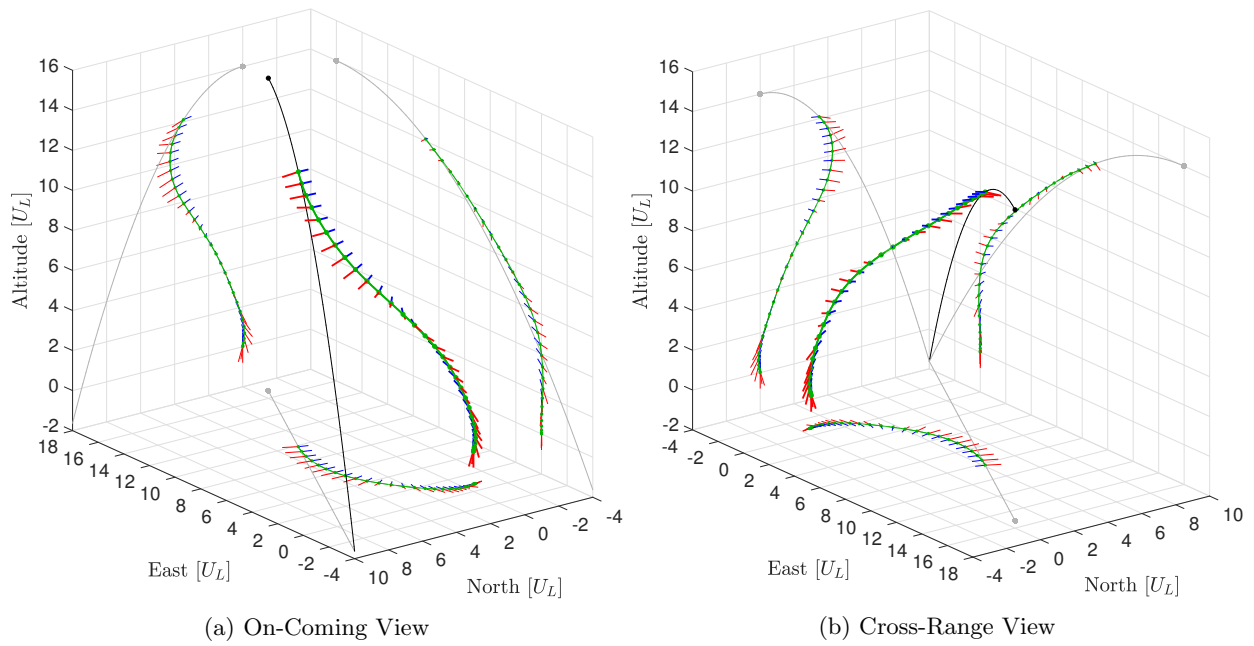(a) On-Coming View

(b) Cross-Range View

Figure 6: A powered descent guidance scenario with a free-ignition-time free fall trajectory (shown in black) and with no state-triggered constraint. The trajectory is shown in green, the thrust vectors in red, and the long axis of the vehicle in blue.
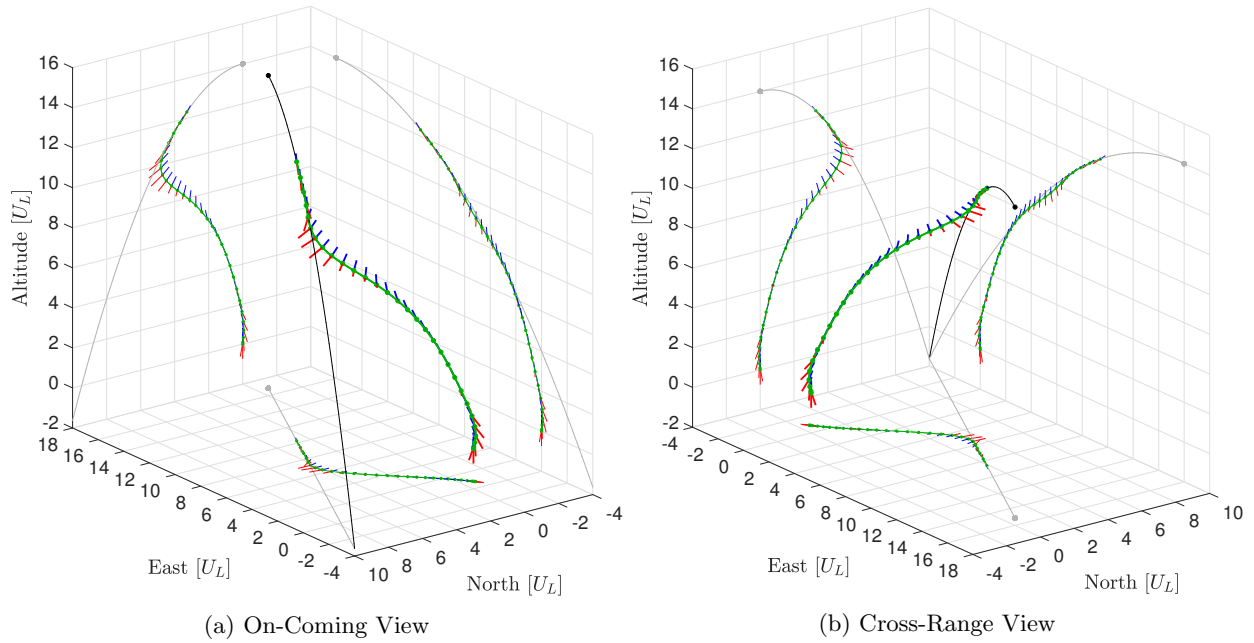

(a) On-Coming View

(b) Cross-Range View

Figure 7: A powered descent guidance scenario with free-ignition-time *and* a velocity-triggered angle of attack constraint.

vehicle is not allowed to break the prescribed speed limit. Between times $4\ U_T$ and $6\ U_T$, the velocity climbs above the trigger value because the angle of attack is below its threshold. This behavior demonstrates that the state triggered constraint can introduce a new constraint phase into the solution, where the multi-phase optimization approach described in Alternative 3 of §II.A.4 would have failed to do so. Moreover, the peak seen between times $5\ U_T$ and $6\ U_T$ in the top plot of Figure 8b is a result of the vehicle exploiting a low angle of attack to gain speed and reduce its minimum time objective. Lastly, as one would expect, the case in Figure 8a managed to find a more optimal burn time when compared to the more constrained case in Figure 8b.

American Institute of Aeronautics and Astronautics

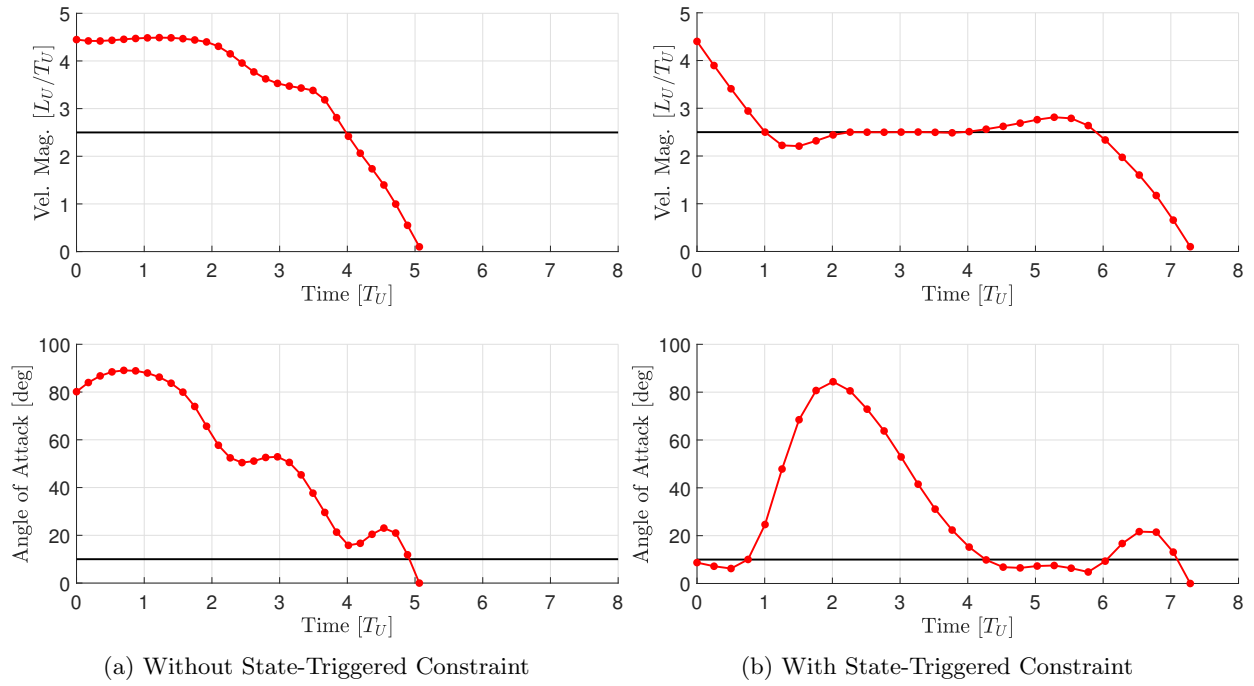(a) Without State-Triggered Constraint

(b) With State-Triggered Constraint

Figure 8: Velocity and angle of attack time histories of the cases shown in Figure 8a and 8b. The black lines represent the trigger velocity V_α and the angle of attack limit α_{max}.
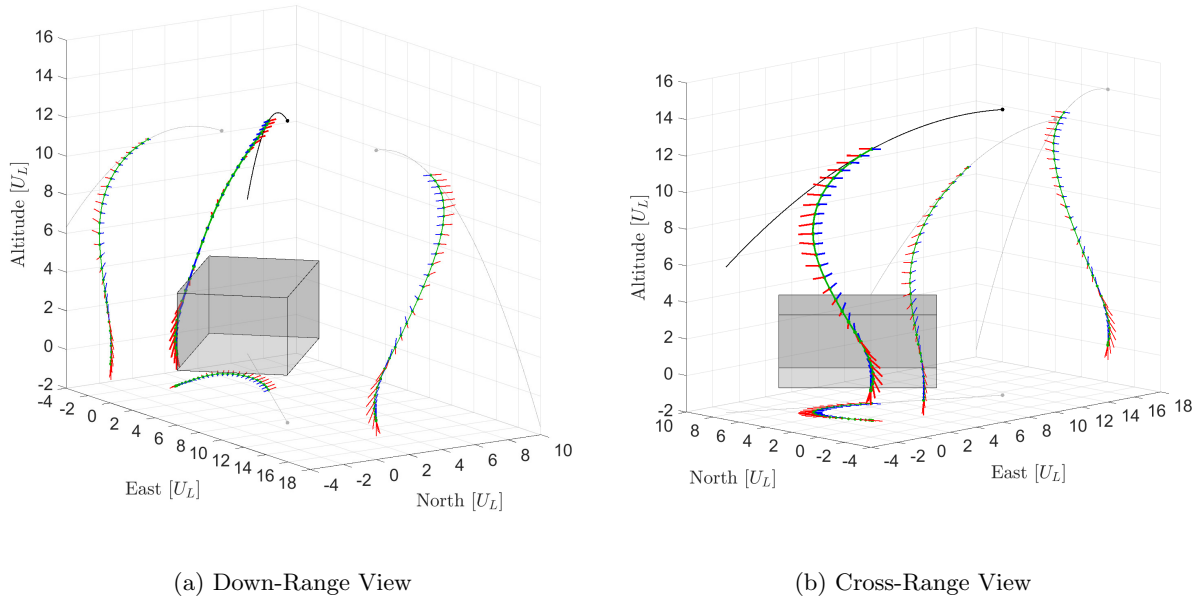

(a) Down-Range View

(b) Cross-Range View

Figure 9: A powered descent guidance scenario with a collision avoidance compound state-triggered constraint. The vehicle can be seen maneuvering clear of the geographical feature.

IV.B. Example 2: Compound State-Triggered Constraints

Figure 9 shows a powered descent guidance trajectory employing a compound state-triggered constraint to avoid geological formations. This case is initialized at an altitude 6 U_L lower than the previous example, with the same initial velocity and coast time. While this simulation result succeeded, additional work will be required to improve convergence properties for these constraints.

American Institute of Aeronautics and Astronautics

V. Conclusion

This paper presents a powered descent guidance problem formulation that contains state-triggered and compound state-triggered constraints. The primary contribution of this paper is a novel continuous formulation for *compound* state-triggered constraints. This formulation is a generalization of the state-triggered constraint formulation presented in [40], and enables the use of trigger and constraint conditions composed using Boolean *and* and *or* operations. Both state-triggered constraint formulations are demonstrated in example simulations using the successive convexification framework. The examples showcase a velocity-triggered angle of attack constraint that is used to alleviate aerodynamic loads, and collision avoidance constraint used to avoid large geological formations near the landing site. Notably, the velocity-triggered angle of attack constraint demonstrates the ability of state-triggered constraints to introduce and rearrange constraint phases in the solution, without resorting to combinatorial techniques. While the results presented here and in [40] show promise for powered descent guidance applications, additional work is required to improve the convergence behavior resulting from the inclusion of state-triggered constraints.

VI. Acknowledgements

References

[1] Blackmore, L., "Autonomous Precision Landing of Space Rockets," <u>NAE Winter Bridge on Frontiers of Engineering</u>, Vol. 4, No. 46, 2016.

[2] Meditch, J. S., "On the Problem of Optimal Thrust Programming For a Lunar Soft Landing," <u>IEEE Transactions on Automatic Control</u>, Vol. 9, No. 4, 1964, pp. 477–484.

[3] Lawden, D., "Optimal Trajectories for Space Navigation," <u>Optimal Trajectories for Space Navigation</u>, Butterworths, London, 1963.

[4] Marec, J.-P., <u>Optimal Space Trajectories</u>, Elsevier Scientific Publishing Company, Amsterdam, 1979.

[5] Klumpp, A. R., "Apollo Lunar Descent Guidance," <u>Automatica</u>, Vol. 10, 1974, pp. 133–146.

[6] Kornhauser, A. L. and Lion, P. M., "Optimal Deterministic Guidance for Bounded-Thrust Spacecrafts," <u>Celestial Mechanics</u>, Vol. 5, No. 3, 1972, pp. 261–281.

[7] Breakwell, J. V. and Dixon, J. F., "Minimum-Fuel Rocket Trajectories Involving Intermediate-Thrust Arcs," <u>Journal of Optimization Theory and Applications</u>, Vol. 17, No. 5, 1975, pp. 465—-479.

[8] Azimov, D. M., "Analytic solutions for intermediate-thrust arcs of rocket trajectories in a newtonian field," <u>Journal of Applied Mathematics and Mechanics</u>, Vol. 60, No. 3, 1996, pp. 421–427.

[9] D'Souza, C., "An Optimal Guidance Law for Planetary Landing," <u>Proceedings of the AIAA Guidance, Navigation and Control Conference</u>, 1997.

[10] Topcu, U., Casoliva, J., and Mease, K. D., "Fuel Efficient Powered Descent Guidance for Mars Landing," <u>Guidance, Navigation ,and Control Conference and Exhibit</u>, No. August, San Francisco, CA, 2005.

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

[12] Acikmese, A. B. and Ploen, S., "A Powered Descent Guidance Algorithm for Mars Pinpoint Landing," <u>AIAA Guidance, Navigation, and Control Conference and Exhibit</u>, Guidance, Navigation, and Control and Co-located Conferences, American Institute of Aeronautics and Astronautics, August 2005.

[13] Ploen, S., Açıkmeşe, B., and Wolf, A., "A Comparison of Powered Descent Guidance Laws for Mars Pinpoint Landing," <u>AIAA GNC Conference and Exhibit, Keystone, CO</u>, 2006.

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

[15] Açıkmeşe, B. and Blackmore, L., "Lossless convexification for a class of optimal control problems with nonconvex control constraints," <u>Automatica</u>, Vol. 47, No. 2, 2011, pp. 341–347.

[16] Harris, M. W. and Açıkmeşe, B., "Lossless Convexification for a Class of Optimal Control Problems with First Order State Constraints," <u>American Control Conference</u>, 06/2013 2013.

[17] Blackmore, L., Açıkmeşe, B., and Carson, J. M., "Lossless convexication of control constraints for a class of nonlinear optimal control problems," <u>Systems and Control Letters</u>, Vol. 61, 2012, pp. 863–871.

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

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

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

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

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

[23] Scharf, D., Açıkmeşe, B., Dueri, D., Casoliva, J., and Benito, J., "Implementation and Experimental Demonstration of Onboard Powered Descent Guidance," to be submitted to AIAA Journal of Guidance, Control, and Dynamics, 2015.

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

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

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

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

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

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

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

[31] Mao, Y., Szmuk, M., and Açıkmeşe, B., "Successive Convexification of Non-Convex Optimal Control Problems and its Convergence Properties," 2016 IEEE 55th Conference on Decision and Control (CDC), Dec 2016, pp. 3636–3641.

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

[33] Mao, Y., Szmuk, M., and Açıkmeşe, B., "Successive Convexification: A Superlinearly Convergent Algorithm for Non-convex Optimal Control Problems," ArXiv e-prints, April 2018, arXiv:1804.06539.

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

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

[36] Boggs, P. T. and Tolle, J. W., "Sequential Quadratic Programming," Acta Numerica, Vol. 4, 1995, pp. 1–51.

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

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

[39] Szmuk, M. and Açıkmeşe, B. A., "Successive Convexification for 6-DoF Mars Rocket Powered Landing with Free-Final-Time," AIAA Guidance, Navigation, and Control Conference, Kissimmee, FL, 2018, p. 0617.

[40] Szmuk, M., Reynolds, T. P., and Açıkmeşe, B., "Successive Convexification for Real-Time 6-DoF Powered Descent Guidance with State-Triggered Constraints," arXiv e-prints, November 2018, arXiv:1811.10803.

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

[42] Heemels, W., Schumacher, J. M., and Weiland, S., "Linear Complementarity Systems," SIAM Journal of Applied Math, Vol. 60, No. 4, 2000, pp. 1234–1269.

[43] Biegler, L., "Recent Advances in Chemical Process Optimization," Chemie Ingenieur Technik, Vol. 86, No. 7, 2014, pp. 943–952.

[44] Richards, A. and How, J., "Mixed-integer Programming for Control," Proceedings of the American Control Conference, Portland, OR, 2005, pp. 2676–2683.

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

[46] Bhasin, N. S., Fuel-Optimal Spacecraft Guidance for Landing in Planetary Pits, Ph.D. thesis, Carnegie Mellon University Pittsburgh, 2016.

[47] Domahidi, A., Chu, E., and Boyd, S., "ECOS: An SOCP solver for embedded systems," European Control Conference (ECC), 2013, pp. 3071–3076.