JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 43, No. 7, July 2020

Collision-Inclusive Trajectory Optimization for Free-Flying Spacecraft

Mark Mote,[*] Magnus Egerstedt,[†] and Eric Feron[‡]
Georgia Institute of Technology, Atlanta, Georgia 30332
and
Andrew Bylard[§] and Marco Pavone[¶]
Stanford University, Stanford, California 94305

https://doi.org/10.2514/1.G004788

Collisions may be harnessed as a way to improve the overall safety and navigational effectiveness of some spacecraft. However, leveraging this capability in autonomous platforms requires the ability to plan trajectories comprising impulsive contact. This paper addresses this problem through the development of a collision-inclusive approach to optimal trajectory planning for a three-degree-of-freedom free-flying spacecraft. First, experimental data are used to formulate a physically realistic collision model for the spacecraft. It is shown that this model is linear over the expected operational range, enabling a piecewise affine representation of the full hybrid vehicle dynamics. Next, the dynamics model and vehicle constraints are incorporated into a mixed integer program. Experimental comparisons of trajectories with and without collision-avoidance requirements demonstrate the capability of the collision-inclusive strategy to achieve significant performance improvements in realistic scenarios. A simulated case study illustrates the potential for this approach to find damage-mitigating paths in online implementations.

Nomenclature

$\mathcal{A}$ = net obstacle avoidance region, composed of polygons, $\mathcal{P}$

$\boldsymbol{a}_j$, b_j = parameters defining the half-plane representing the jth wall

$\mathcal{F}_I$ = inertial reference frame defined for the testbed

$\mathcal{F}_w^j$ = inertial reference frame defined for the jth wall

$\boldsymbol{I}_n$ = $n \times n$ identity matrix

$\hat{i}$, $\hat{j}$, $\hat{k}$ = basis vectors for the inertial frame $\mathcal{F}_I$

J = cost function

M = large scalar defined according to the *Big-M* method

m = mass of spacecraft, kg

$\mathcal{O}$ = origin of inertial frame $\mathcal{F}_I$

$\mathcal{P}$ = convex polygon defining an obstacle avoidance region

R = radius of the spacecraft, measured from rotation center to point of contact on outer surface, m

$\mathcal{S}$, $\bar{\mathcal{S}}$ = half-plane and polygon defining surfaces with which collision is permitted

s_T, s_N = position tangent and normal to a wall, m

$\hat{t}_j$, $\hat{a}_j$, $\hat{k}_j$ = basis vectors for the local frame $\mathcal{F}_w^j$, defined tangent, normal, and upward from the jth wall

$\boldsymbol{u}$ = spacecraft control vector, $\mathrm{m/s^2}$, $\mathrm{m/s^2}$, $\mathrm{rad/s^2}$

v_T, v_N = components of velocity tangent and normal to a wall, m/s

$\boldsymbol{x}_i$ = state of the vehicle on the ith iteration, expressed in the inertial frame $\mathcal{F}_I$

$\boldsymbol{z}$ = generic vector used to define sets

$\gamma_{i,j}$ = event variable indicating position with respect to the jth wall of $\bar{\mathcal{S}}$ on the ith iteration

Δt = fixed time period between samples in discrete time horizon, s

δ = effective distance of a collision surface from the origin of the inertial testbed frame $\mathcal{F}_I$, m

ζ_i = event variable used to indicate collisions with $\mathcal{S}$ on the ith iteration

θ = orientation of the spacecraft axes with respect to the inertial frame, measured as angle about $\hat{k}$, rad

κ_T, κ_N, κ_ω = tangential, normal, and angular restitution coefficients

$\lambda_{i,j}$ = event variable indicating position with respect to the half-plane defined by α_j, β_j on the ith iteration

$\Xi_{i,j}$ = event variable indicating collision with the jth wall of $\bar{\mathcal{S}}$ on the ith iteration

τ = number of steps in discrete time horizon

ω = angular velocity of spacecraft with respect to $\mathcal{F}_I$, taken as the time derivative of θ, rad/s

$\boldsymbol{0}_n$ = $n \times n$ zeros matrix

I. Introduction

ASSISTIVE robotic spacecraft have the potential to enable the automation of many tasks that are not well-suited to be directly performed by astronauts, because they are either too dangerous or overly tedious [1]. Within this context, extravehicular platforms have been proposed for missions such as on-orbit monitoring [2,3], component assembly [4], and debris removal [5]. Likewise, intravehicular assistive robotics [6] is being developed to fulfill many housekeeping duties inside the International Space Station (ISS). For instance, the Astrobee robot [7], a successor to NASA's highly successful SPHERES [8] testbed, has stated goals of 1) providing a microgravity research platform, 2) performing mobile camera tasks, and 3) performing mobile sensor tasks for environment monitoring and inventory management. As with most mobile autonomous platforms, safe and efficient navigation is key to the successful integration of these vehicles into mission operations.

A popular approach to mobility for microgravity robots is expending propellant to actuate movement between periods of free-flight, often referred to as propulsive free-flying. One may find key results related to path planning and close proximity operations for this case in [9–12]. However, a common issue facing this approach is that propellant is often expensive to acquire or in limited supply. As a

Received 14 September 2019; revision received 5 February 2020; accepted for publication 5 February 2020; published online 10 March 2020.

[*] Ph.D. Candidate, School of Aerospace Engineering.

[†] Steve W. Chaddick School Chair and Professor, School of Electrical and Computer Engineering.

[‡] Dutton/Ducoffe Professor, Aerospace Software Engineering, School of Aerospace Engineering; currently on leave at King Abdullah University of Science and Technology (KAUST).

[§] Ph.D. Candidate, Aeronautics and Astronautics Department.

[¶] Associate Professor, Aeronautics and Astronautics Department.

consequence of this, fuel efficiency has become one of the primary performance characteristics for spacecraft. The desire to reduce costs has motivated the development of two alternate navigation modalities. The first is zero-g climbing [13], where the vehicle uses grasping contact in the surrounding environment to traverse between locations. A proposed faster and simpler alternative is the hopping modality [14,15]. In this case, the vehicle uses a robotic arm to propel itself between some fixed handrails. Although this strategy is attractive in the sense that it is completely propellantless, it is also more restrictive than the propulsive free-flying strategy in the sense that it requires the precise coordination of a robotic arm and requires handrails to be present over the operational region.

This paper presents a new approach to mobility for assistive spacecraft: supplementation of propulsive free-flying with planned collisional contact (bouncing). To the best of our knowledge, this approach to spacecraft mobility has not been explored in the open literature. We show that this offers a strategy that is both less restrictive than the propellant-free approaches, and often more efficient (with respect to a given cost function) than its collision-free counterpart. In contrast to hopping, where a robotic arm interacts with the environment to provide the energy needed to change the momentum of the spacecraft, bouncing achieves similar maneuvers passively though impulsive contact. For example, a spacecraft needing to redirect itself inside a corridor may do so swiftly with a single well-planned collision, rather than executing the series of maneuvers needed for coordinated hopping. Because the interaction is passive, bouncing poses very little requirements on the vehicle or the surrounding environment itself. Hence the main challenge stems from the task of developing an effective motion planning strategy to leverage this capability. Focusing specifically on the case of small, assistive intravehicular spacecraft, we assume that the vehicle operates in the proximity of fixed surfaces with which it may collide, and that both the vehicle and the surface are able to withstand low-speed impact.

There is a rich body of work related to impulsive contact in robotics, spanning applications such as running [16]; jumping [17]; batting [18]; air hockey [19]; and car following [20]. In addition, the problem appears in the aerospace context, within landing [21], docking [22], grasping [23], and bouncing on planetary bodies [24]. Looking specifically at the case of vehicle collisions, there has been foundational work in analyzing the stability and robustness of a colliding vehicle [25], designing vehicles that are tolerant to collisions [26], and even extracting localization information from instances of impact [27]. Collisions can further be harnessed as a practical means of improving the effectiveness of trajectories. Through dissipation of energy or redirection of momentum, colliding agents are endowed with greater maneuverability. One can observe many examples of this phenomenon in competitive situations, for example, swimming and parkour, and in nature (e.g., animals pushing off of [28] or jumping between objects). However, the use of planned impulsive contact explicitly for performance gains has only recently been considered in the context of robot trajectory planning. In [29], the authors use a mixed integer linear programming (MILP) formulation to derive a time-optimal trajectory incorporating planned collisions for a point mass. In this paper, we use these initial results to develop a collision-inclusive, optimal trajectory-planning formulation for in-plane motion of a free-flying spacecraft. Note that because the overall set of trajectories allowing collisions encompasses all collision-free trajectories as well, the optimal performance with respect to any objective function must either remain the same or improve when compared with the case where collisions are always avoided.

In addition to performance benefits, collisions may be used to improve the safety of a vehicle in the presence of observed changes in the surrounding environment. Intuitively, in situations where collisions cannot be avoided, a safest plan of action incorporating the collision may be found. Looking specifically at the case of online model predictive control (MPC), hard collision-avoidance constraints may render the problem infeasible when collisions are unavoidable. This problem can be addressed by either resorting to a backup controller when the MPC is not feasible [30] or softening the constraints (i.e., replacing constraints with penalties in the objective function) such that feasibility is preserved [31]. We extend this prototypical constraint-softening approach with the addition of an explicit model of the collision dynamics formulated in the constraints. In addition to remaining feasible in the presence of an inevitable collision, this allows the vehicle to plan around the collision, all while minimizing a penalty function that captures the estimated damage cost. This additional safety measure may offer a particularly useful tool for platforms proposing autonomous operation in the presence of humans.

The remainder of the paper is outlined as follows: In Sec. II, we review the mathematical preliminaries required to develop the main results. In Sec. III.A we introduce the assumptions on the spacecraft, and develop basic dynamical constraints. Experimental collision data are obtained and used to derive a realistic collision model for the spacecraft in Sec. III.B. Section IV uses the motion model and vehicle constraints to specify an optimal strategy for moving between states. In Sec. V.A an experimental case study is described, and the performance of the collision-inclusive algorithm is compared with that of the collision-free case. It is shown that the proposed method is capable of significantly reducing a chosen objective function. Finally, Sec. V.B explores potential safety applications with a simulated scenario.

II. Mixed Integer Programming for Control of Hybrid Systems

Mixed integer programming (MIP) denotes an optimization problem that is composed of both real and integer decision variables. This type of problem provides a very general framework for capturing many types of practical control objectives. Specifically, the inclusion of integer variables allows for the expression of discrete decisions. This makes it naturally well suited to optimizing the actions over systems governed by interdependent dynamic modes, logical statements, and operational constraints [32]. For our purposes, this is leveraged to optimize trajectories for a spacecraft experiencing unique dynamic modes encountered during collision and free flight. By modeling this hybrid behavior, integer variables encode the choice of whether or not to collide.

We consider programs where the objective function $J(z)$ is optimized over piecewise affine (PWA) constraints, fitting the form below.

$$\begin{aligned} \min_{z} \quad & J(z) \\ \text{s.t.} \quad & \boldsymbol{D_c z_c} + \boldsymbol{D_b z_b} \le \boldsymbol{g}, \quad \boldsymbol{A_c z_c} + \boldsymbol{A_b z_b} = \boldsymbol{h} \\ & \boldsymbol{z_c} \in \mathbb{R}^{n_c}, \quad \boldsymbol{z_b} \in \{0,1\}^{n_b}, \quad \boldsymbol{z} = [\boldsymbol{z_c}, \boldsymbol{z_b}] \in \mathbb{R}^{n} \end{aligned} \tag{1}$$

where $\boldsymbol{D_c} \in \mathbb{R}^{m \times n_c}$, $\boldsymbol{D_b} \in \mathbb{R}^{m \times n_b}$, $\boldsymbol{g} \in \mathbb{R}^{m}$, $\boldsymbol{A_c} \in \mathbb{R}^{p \times n_c}$, $\boldsymbol{A_b} \in \mathbb{R}^{p \times n_b}$, and $\boldsymbol{h} \in \mathbb{R}^{p}$. Although the problem is not convex in general, one can in principle compute globally optimal solutions whenever J is convex by solving a finite number of convex subproblems [33]. Formulations with linear, quadratic, or second-order cone objectives are commonly applied in a variety of practical applications [12,34–36]. Although in theory these problems are difficult to solve [37], solutions can readily be found with good average case performance using off-the-shelf optimization software (e.g., CPLEX [38], Gurobi [39], MOSEK [40]).

MIP allows for the representation of hybrid systems by associating integer variables with the current mode of the system. Specifically, integer variables (also known as event variables) allow for the direct expression of first-order logic over the constraints. These variables may be assigned a unique value based on the location of the state vector, and in turn be used to relax a different set of constraints over the continuous variables. To demonstrate this, let us consider the following case, where an inequality condition $\boldsymbol{c}^T \boldsymbol{z} < d$ is used to *activate* distinct equality constraints:

$$\begin{cases} \boldsymbol{a}_{\boldsymbol{0}}^T \boldsymbol{z} = b_0 & \text{if } \boldsymbol{c}^T \boldsymbol{z} < d \\ \boldsymbol{a}_{\boldsymbol{1}}^T \boldsymbol{z} = b_1 & \text{if } \boldsymbol{c}^T \boldsymbol{z} \ge d \end{cases} \tag{2}$$

with $\boldsymbol{z}, \boldsymbol{a}_{\boldsymbol{i}}, \boldsymbol{c} \in \mathbb{R}^n, b_i, d \in \mathbb{R}, i = 0, 1$. The main tools at our disposal for representing hybrid systems as programs in the form of Eq. (1) come from the lemmas below, which define relationships between implications and inequalities of real and binary decision variables.

Let $\zeta \in \{0, 1\}$, $z \in Z \subset \mathbb{R}^n$, and parameters $\boldsymbol{a}, \boldsymbol{c} \in \mathbb{R}^n$ and $b, d \in \mathbb{R}$. We then have the following results [41–44].

Lemma 1 [32]: Given $M \in \mathbb{R}$ such that $\max_{z \in Z}(d - \boldsymbol{c}^T z) < M$, the following are equivalent:

$$(i)\ [\boldsymbol{c}^T z < d] \Rightarrow [\zeta = 1] \qquad (ii)\ \boldsymbol{c}^T z + M\zeta \geq d$$

Proof: If $\max_{z \in Z}(d - \boldsymbol{c}^T z) < M$ holds, then the statement (ii) is true for all $z \in Z$ when $\zeta = 1$. Given $\zeta = 0$, (ii) is true when $\boldsymbol{c}^T z < d$ holds, and is false otherwise. Thus, the truth values for (ii) are identical to the implication (i) for all assignments. □

Lemma 2 [32]: Given $M \in \mathbb{R}$ such that $\max_{z \in Z}(\boldsymbol{c}^T z - d) < M$, the following are equivalent:

$$(i)\ [\zeta = 1] \Rightarrow [\boldsymbol{c}^T z < d] \qquad (ii)\ \boldsymbol{c}^T z - M(1 - \zeta) < d$$

Proof: If $\max_{z \in Z}(\boldsymbol{c}^T z - d) < M$ holds, then (ii) is true for all $\zeta = 0$. Given $\zeta = 1$, (ii) is equivalent to $\boldsymbol{c}^T z < d$. □

The proofs for the above lemmas may also be made apparent via the form of a truth table (see Appendix C). Note that we can apply these together to form an equivalency. Likewise, application to inequalities of opposing sense (in conjunction) extends the result to the case of equality constraints.

Lemma 3 ([45] Sec. 16.4): Given $M \in \mathbb{R}$ such that $\max_{z \in Z}(\boldsymbol{a}^T z - b) < M$, the following are equivalent:

$$(i)\,[\zeta = 1] \Rightarrow [\boldsymbol{a}^T z = b] \quad (ii)\,[\boldsymbol{a}^T z - M(1 - \zeta) \leq b] \wedge [\boldsymbol{a}^T z + M(1 - \zeta) \geq b]$$

Proof: If $\max_{z \in Z}(\boldsymbol{a}^T z - b) < M$ holds, then (ii) is trivially satisfied for $\zeta = 0$. Given $\zeta = 1$, (ii) is equivalent to $\boldsymbol{a}^T z = b$. Equivalence then follows from the truth table. □

From here we can combine these results to yield a conjunction of mixed integer inequalities that is equivalent to Eq. (2) over some specified range on z.

Theorem 1 ([45] Sec. 16.4): Given $M \in \mathbb{R}$ sufficiently large such that $\max(\max_{z \in Z}(|\boldsymbol{c}^T z - d|), \max_{z \in Z}(\boldsymbol{a}^T z - b)) < M$ holds, then the system Eq. (2) is equivalent to

$$[\boldsymbol{a}_0^T z - M\zeta \leq b_0] \wedge [\boldsymbol{a}_1^T z - M(1 - \zeta) \leq b_1] \wedge [\boldsymbol{c}^T z - M\zeta < d]$$
$$[\boldsymbol{a}_0^T z + M\zeta \geq b_0] \wedge [\boldsymbol{a}_1^T z + M(1 - \zeta) \geq b_1] \wedge [\boldsymbol{c}^T z + M(1 - \zeta) \geq d]$$

In practice, the parameter M should be chosen carefully. Whereas values that are too low may not satisfy the above conditions, excessively large values will decrease computational efficiency and may introduce numerical error. For notational simplicity, the sequel uses the same parameter M in all instances of this method. Note that from a computational viewpoint it is often better to avoid strict inequalities in implementation. This may be accomplished by using a nonstrict inequality and adding a small number ε to the side with lesser value.

III. Vehicle Description and Constraints

The system of interest consists of a single free-flying spacecraft in the presence of one or more flat surfaces. The spacecraft is subject to input constraints and is restricted to moving within a plane. The colliding surface is assumed to be fixed and orthogonal to this plane. The collision model is developed empirically for the specific use case of operation on the robot and testbed at Stanford's Space Robotics Facility (see Fig. 1). A detailed description of this environment and the experimental setup is provided in Sec. V.A.1.

A. Free Flight Dynamics

The motion of the spacecraft is expressed in an inertial frame $\mathcal{F}_I = (\mathcal{O}, \hat{\imath}, \hat{\jmath}, \hat{k})$ with right-handed orthogonal basis vectors $\hat{\imath}, \hat{\jmath}$ lying in the plane of motion and $\hat{k} = \hat{\imath} \times \hat{\jmath}$. For the testbed shown in Fig. 1, the origin $\mathcal{O}$ is taken as the lower left corner, $\hat{\imath}, \hat{\jmath}$ lie along the testbed boundary, and $\hat{k}$ points upward. The position of the vehicle's center of mass $\mathcal{O}_B$ with respect to $\mathcal{O}$ is $\boldsymbol{s} = s_x \hat{\imath} + s_y \hat{\jmath}$ and the translational velocity is $\boldsymbol{v} = v_x \hat{\imath} + v_y \hat{\jmath}$. We may also define a body frame $\mathcal{F}_B = (\mathcal{O}_B, \hat{\imath}_B, \hat{\jmath}_B, \hat{k}_B)$, with basis vectors $\hat{\imath}_B$ and $\hat{\jmath}_B$ aligned with the orientation of the thrusters, and $\hat{k}_B = \hat{k}$. The orientation of $\mathcal{F}_B$ with respect to $\mathcal{F}_I$ is θ, with positive increments in the angle θ corresponding to counterclockwise rotations of the spacecraft, as seen from above. This geometry is illustrated in Fig. 2a. The angular velocity of the vehicle is $\omega = \dot{\theta}\,\hat{k}$. The *nominal*—that is, collision-free—spacecraft dynamics[**] are then

$$\ddot{s}_x = u_x, \qquad \ddot{s}_y = u_y, \qquad \ddot{\theta} = u_\theta \tag{3}$$

where u_x and u_y are the translational accelerations due to applied thrust, and u_θ is the rotational acceleration from an applied moment, which is generated by changes in the reaction wheel speed from a lower level controller. We assume that the thrust inputs are balanced, such that the applied moment comes entirely from the reaction wheel. For the configuration shown in Fig. 2a, the relationship between translational acceleration and the thrust output from the eight individual thrusters is

$$\begin{bmatrix} u_x \\ u_y \end{bmatrix} = \frac{1}{m} \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} 0 & -1 & 1 & 0 & 0 & 1 & -1 & 0 \\ -1 & 0 & 0 & -1 & 1 & 0 & 0 & 1 \end{bmatrix} \boldsymbol{u}_T \tag{4}$$

where m is the mass of the spacecraft and $\boldsymbol{u}_T \in [0, u_{T,\max}]^8$ is the vector of individual thruster output forces. Here $u_{T,\max}$ represents the maximum force output of a single thruster.

Note that the thruster arrangements on the spacecraft are such that the maximum accelerations achievable in the $\hat{\imath}$, $\hat{\jmath}$ directions are functions of the body orientation θ. We can simplify this with the use of a conservative inner approximation on the maximum acceleration from thrust $u_{\max}(\theta)$, which generates a condition that is uniform (not dependent on orientation) in the inertial frame.

$$u_x^2 + u_y^2 \leq u_{\max}^2, \qquad u_{\max} = \min_\theta(u_{\max}(\theta))) \tag{5}$$

With the present geometry, we have that $u_{\max} = (2/m)u_{T,\max}$.

B. Collision Model

To develop a framework for optimizing trajectories that allow collisions, we must first develop a model for the collision effects on the spacecraft. Collisions are generally difficult to understand and model conceptually, as a first principles analysis requires the consideration of many interacting physical phenomena relating to the geometric, material, and inertial properties of each body involved, many of which are in themselves difficult to model accurately. Many approaches have been proposed to model *general* collision behavior over a wide range of scenarios [46,47]. However, because we use a specific pair of objects over a relatively limited range of conditions, we are able to develop an algebraic collision model empirically, by directly considering the relationship between the velocities immediately before and after the instant of contact with no thrust commanded. Figure 3 shows the effects of 82 individual collisions for the free-flyer spacecraft and testbed shown in Fig. 1. Within the tested range, the data suggest that the changes in rotational velocity ($\Delta\omega$), translational velocity normal to the wall (Δv_N), and tangent to the wall (Δv_T) all follow a linear relationship with the precollision normal velocity (v_N^-) and relative velocity of the point of contact (v_{rel}^-). Furthermore, we observe that, for this set of parameters, effects in the normal direction are uncoupled from the tangential and rotational effects, leading to the following model:

$$\begin{bmatrix} \Delta v_T \\ \Delta v_N \\ \Delta\omega \end{bmatrix} = \begin{bmatrix} 0 & \kappa_T \\ \kappa_N & 0 \\ 0 & \kappa_\omega \end{bmatrix} \begin{bmatrix} v_N^- \\ v_{\text{rel}}^- \end{bmatrix} \tag{6}$$

[**] The modeling and trajectory planning phases will assume a disturbance-free and deterministic spacecraft model. Noise, parametric uncertainty, and errors from approximation are to be addressed through online regulation.

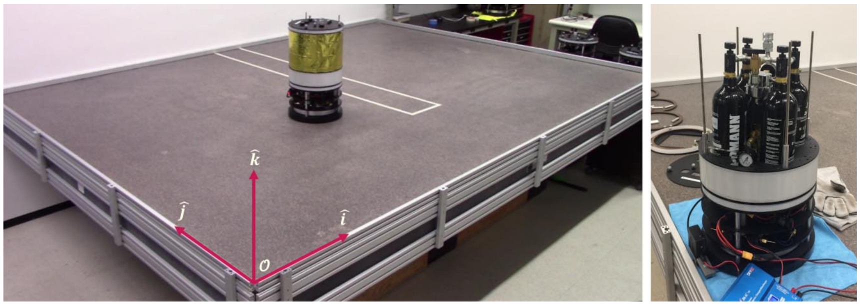

Fig. 1 Free-flyer spacecraft and testbed with the avoidance region for the experiment in Sec. V.A outlined in adhesive tape.

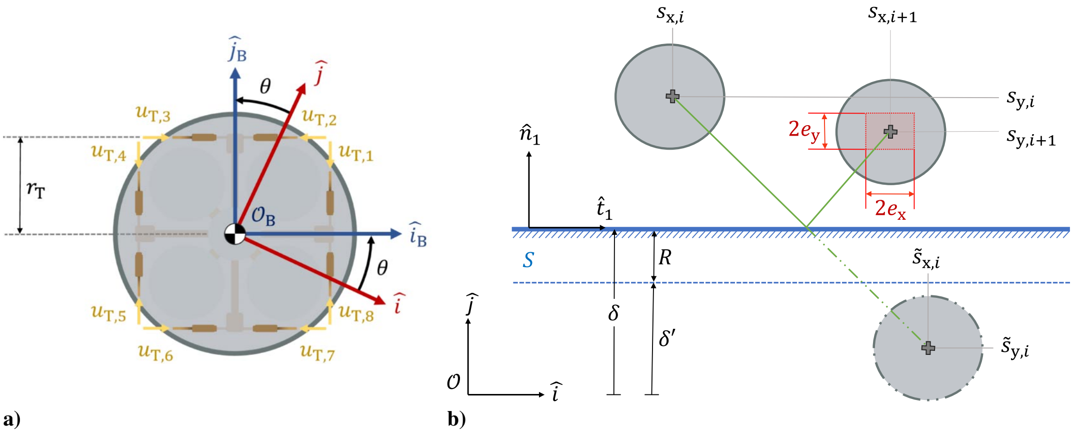

Fig. 2 Geometry and conventions for a) free-flyer spacecraft and b) collision with a flat wall $\mathcal{S}$.

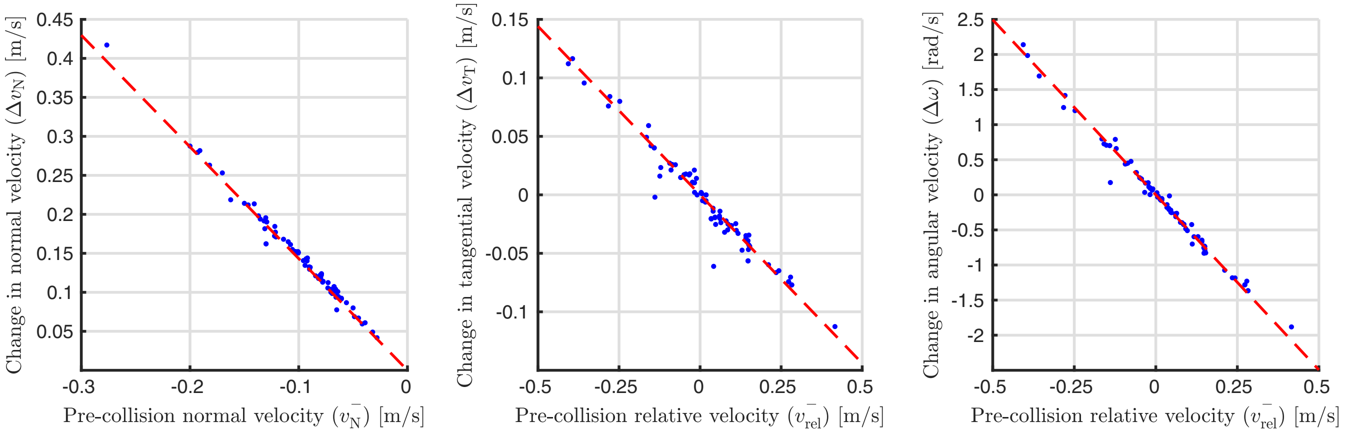

Fig. 3 Observed data from 82 collisions, with linear interpolations taken with respect to the least-squares error.

where

$$v_{\mathrm{rel}}^{-} \triangleq v_T^{-} + R\omega^{-} \tag{7}$$

$(\kappa_T, \kappa_N, \kappa_\omega) = (-0.29, -1.43, -5.0)$, and R is the radius of the spacecraft, measured from the outer rim to the rotation center. The coefficients are obtained via a least-squares regression on the model error.

If we assume that the collision occurs instantaneously, the positions after the collision can be obtained by integrating the equations of motion with precollision velocities until the point of contact, and postimpact velocities afterward. Let $\Delta t \triangleq \Delta t^{-} + \Delta t^{+}$ be the period between the state measurements, and δ be the effective location of the wall along the orthogonal axis (inflated by R, as seen in Fig. 2b). Then the experimental model of Eq. (6) yields the following position update equations:

$$\begin{aligned}\Delta s_T &= (1+\kappa_T)\Delta t v_T^{-} + \kappa_T R \Delta t \omega^{-} - \kappa_T (v_T^{-} + R\omega^{-})\Delta t^{-}\\ \Delta s_N &= (1+\kappa_N)\Delta t v_N^{-} + \kappa_N (s_N^{-} - \delta)\\ \Delta\theta &= (1+\kappa_\omega R)\Delta t \omega^{-} + \kappa_\omega \Delta t v_T^{-} - \kappa_\omega (v_T^{-} + R\omega^{-})\Delta t^{-}\end{aligned} \tag{8}$$

where the time until collision is

$$\Delta t^{-} = \frac{(\delta - s_N^{-})}{v_N^{-}} \tag{9}$$

Note that the term Δt^{-} introduces a nonlinearity in the tangential and rotational update laws. Making the approximation that collision occurs midway through the interval $\Delta t^{-} = 0.5\Delta t$ allows us to obtain a linear form of these equations. The bounds on error from this assumption can be calculated from the maximum difference between the exact and approximated equations, which yields

$$e_T \le \frac{\kappa_T}{2}|v_{\mathrm{rel}}^{-}|\Delta t, \qquad e_N = 0, \qquad e_\theta \le \frac{\kappa_\omega}{2}|v_{\mathrm{rel}}^{-}|\Delta t \tag{10}$$

where e_T, e_N, e_θ are the errors in the tangential, normal, and angular directions, respectively. Note that errors vanish both as $|v_{\mathrm{rel}}^{-}|$ decreases and for finer resolutions Δt. The collision geometry of the free-flyer system is illustrated in Fig. 2b.

IV. Problem Formulation

This section formulates the problem of generating optimal trajectories for a spacecraft in the presence of 1) an obstacle avoidance region $\mathcal{A}$, composed of N_P convex polygons $\mathcal{P}_k$, with $k \in \{1, \ldots, N_P\}$, and 2) surfaces $\mathcal{S}$, $\bar{\mathcal{S}}$ (representing half-planes and convex polygons, respectively) with which collisions are permissible. For both and $\mathcal{S}$, $\bar{\mathcal{S}}$, we will use the convention that the *interior* of the walls is denoted by the union of *sublevel* surfaces of some defined planes in $\mathbb{R}^3$, and the exterior is the complement of the interior. It is shown that the combined dynamics of the spacecraft, saturation constraints, and obstacle avoidance conditions are all amenable to approximation with piecewise affine constraints. In light of this, we choose to pose the trajectory optimization problem as an MIP. We consider the discrete time approximations of the models developed in previous sections over a horizon of $i = 1, \ldots, \tau$. The state of the vehicle at the ith time step is defined as $\boldsymbol{x}_i^T = [s_{x,i}, s_{y,i}, \theta_i, v_{x,i}, v_{y,i}, \omega_i]$, and control vector as $\boldsymbol{u}_i^T = [u_{x,i}, u_{y,i}, u_{\theta,i}]$. For completeness, we expound upon some basic control and obstacle avoidance constraint formulations found in [34,43,44].

A. Obstacle Avoidance and Saturation Constraints

The saturation constraint Eq. (5) can be represented by approximating the Euclidean norm with an N_U sided polygon:

$$u_{x,i}\sin\left(\frac{2\pi n}{N_U}\right) + u_{y,i}\cos\left(\frac{2\pi n}{N_U}\right) \le u_{\max}, \quad n = 1,\ldots,N_U, \quad i = 1,\ldots,\tau \tag{11}$$

Whereas the approximation improves with the number of sides N_U, the added constraints may increase the amount of time required to calculate the solution. The aggregate obstacle avoidance region $\mathcal{A}$ can be constructed from a set of N_P convex polygons $\mathcal{P}_k$:

$$\mathcal{A} \triangleq \left\{ z \in \mathbb{R}^2 \middle| \bigvee_{k=1}^{N_P} z \in \mathcal{P}_k \right\},$$
$$\text{where, } \mathcal{P}_k \triangleq \{ z \in \mathbb{R}^2 | \boldsymbol{c}_{k,q}^T z < d_{k,q}, q = 1, \ldots, N_{Q,k} \} \tag{12}$$

where $\boldsymbol{c}_{k,q} \in \mathbb{R}^2$ and $d_{k,q} \in \mathbb{R}$ specify the qth side of the kth polygon, which has $N_{Q,k}$ sides. We can construct the avoidance constraint $\boldsymbol{s} \notin \mathcal{A}$ by defining event variables $\psi_{k,q,i} \in \{0, 1\}$ such that $\boldsymbol{c}_{k,q}^T \boldsymbol{s}_i < d_{k,q} \Rightarrow \psi_{k,q,i} = 1$, and ensuring that the position of the vehicle lies in the positive end (exterior) of at least one of half-spaces defining the walls of each polygon. This is accomplished with the following constraints:

$$\bigwedge_{q=1}^{N_{Q,k}} \boldsymbol{c}_{k,q}^T \boldsymbol{s}_i + M\psi_{k,q,i} \ge d_{k,q} \;\wedge\; \sum_{q=1}^{N_{Q,k}} \psi_{k,q,i} \le N_{Q,k} - 1,$$
$$k = 1,\ldots,N_P, \qquad i = 1,\ldots,\tau \tag{13}$$

Note that each conjunct is an application of Lemma 1, and the summations enforce the condition that there is at least one side q in each polygon such that $\boldsymbol{c}_{k,q}^T \boldsymbol{s}_i \ge d_{k,q}$.

Example: Rectangular Boundary Let us consider the simplified case of a rectangle $\mathcal{A} = \mathcal{P}_1 = \{ z \in \mathbb{R}^2 | z_1 \in (z_1^{\min}, z_1^{\max}), z_2 \in (z_2^{\min}, z_2^{\max}) \}$. The equivalent MIP constraints for the condition $\boldsymbol{s} \notin \mathcal{A}$ are

$$-s_x + M\psi_1 \ge -z_1^{\min} \wedge s_x + M\psi_2 \ge z_1^{\max} \wedge -s_y + M\psi_1 \ge -z_2^{\min}$$
$$\wedge\; s_y + M\psi_2 \ge z_2^{\max} \wedge \sum_{q=1}^{4} \psi_q \le 3. \tag{14}$$

◇

B. Representing Dynamics in the Presence of a Single Collision Surface

For notational simplicity, we assume for this case that the basis vectors of $\mathcal{F}_I$ are oriented with the wall $\mathcal{S}$ so that $\hat{j}$ points away from the wall, $\hat{i}$ is tangent, and $\hat{k} = \hat{i} \times \hat{j}$ remains pointed upward (see Fig. 2b). The discrete time equations of motion are given by

$$\boldsymbol{x}_{i+1} - \boldsymbol{x}_i = \begin{cases} \boldsymbol{A}\boldsymbol{x}_i + \boldsymbol{B}\boldsymbol{u}_i & \text{if } \zeta_{i+1} = 0 \\ \boldsymbol{A}_c\boldsymbol{x}_i + \boldsymbol{b}_c & \text{if } \zeta_{i+1} = 1, \end{cases} \qquad i = 1,\ldots,\tau - 1 \tag{15}$$

where $\boldsymbol{A}$ and $\boldsymbol{B}$ represent the nominal dynamics:

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{0}_3 & \boldsymbol{I}_3\Delta t \\ \boldsymbol{0}_3 & \boldsymbol{0}_3 \end{bmatrix}, \qquad \boldsymbol{B} = \begin{bmatrix} 0.5\boldsymbol{I}_3\Delta t^2 \\ \boldsymbol{I}_3\Delta t \end{bmatrix} \tag{16}$$

and $\boldsymbol{A}_c$ and $\boldsymbol{b}_c$ represent the collision dynamics:

$$\boldsymbol{A}_c = \begin{bmatrix} 0 & 0 & 0 & (1+0.5\kappa_T)\Delta t & 0 & 0.5\kappa_T R\Delta t \\ 0 & \kappa_N & 0 & 0 & (1+\kappa_N)\Delta t & 0 \\ 0 & 0 & 0 & 0.5\kappa_\omega\Delta t & 0 & (1+0.5\kappa_\omega R)\Delta t \\ 0 & 0 & 0 & \kappa_T & 0 & \kappa_T R \\ 0 & 0 & 0 & 0 & \kappa_N & 0 \\ 0 & 0 & 0 & \kappa_\omega & 0 & \kappa_\omega R \end{bmatrix},$$
$$\boldsymbol{b}_c = \begin{bmatrix} 0 \\ -\kappa_N\delta \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{17}$$

The period between steps i and $i - 1$ is Δt, which is assumed to remain constant over all iterations.

Let the wall be located at a distance δ' from the origin of the inertial frame $\mathcal{F}_I$, and define $\delta \triangleq \delta' + R$, where R is the radius of the spacecraft, measured from the rotation center to the point of contact. Then we can define the wall by the set $\mathcal{S} \triangleq \{ \boldsymbol{z} \in \mathbb{R}^2 | z_2 < \delta \}$. The occurrence of a collision can be associated with an event variable $\zeta_i \in \{0, 1\}$. This triggers the switch between the nominal and collision dynamics; it is activated (equal to one) on an iteration i if the nominal dynamics predict that the vehicle will enter $\mathcal{S}$ on that iteration, that is,

$$s_{y,i} + v_{y,i}\Delta t + u_{y,i}0.5\Delta t^2 < \delta \Leftrightarrow \zeta_{i+1} = 1, \qquad i = 1,\ldots,\tau - 1 \tag{18}$$

Using Lemmas 1 and 2, we can express Eq. (18) with the equivalent set of constraints:

$$\begin{cases} s_{y,i} + v_{y,i}\Delta t + u_{y,i}0.5\Delta t^2 + M\zeta_{i+1} \ge \delta \\ s_{y,i} + v_{y,i}\Delta t + u_{y,i}0.5\Delta t^2 - M(1 - \zeta_{i+1}) < \delta, \end{cases} \qquad i = 1,\ldots,\tau - 1 \tag{19}$$

and from Theorem 1, we see that Eq. (15) is equivalent to

$$\begin{cases} \boldsymbol{x}_{i+1} - \boldsymbol{x}_i - \boldsymbol{A}\boldsymbol{x}_i - \boldsymbol{B}\boldsymbol{u}_i + M\boldsymbol{I}_6\zeta_{i+1} \ge \boldsymbol{0} \\ \boldsymbol{x}_{i+1} - \boldsymbol{x}_i - \boldsymbol{A}\boldsymbol{x}_i - \boldsymbol{B}\boldsymbol{u}_i - M\boldsymbol{I}_6\zeta_{i+1} \le \boldsymbol{0} \\ \boldsymbol{x}_{i+1} - \boldsymbol{x}_i - \boldsymbol{A}_c\boldsymbol{x}_i - \boldsymbol{b}_c + M\boldsymbol{I}_6(1 - \zeta_{i+1}) \ge \boldsymbol{0} \\ \boldsymbol{x}_{i+1} - \boldsymbol{x}_i - \boldsymbol{A}_c\boldsymbol{x}_i - \boldsymbol{b}_c - M\boldsymbol{I}_6(1 - \zeta_{i+1}) \le \boldsymbol{0}, \end{cases} \qquad i = 1,\ldots,\tau - 1 \tag{20}$$

Section B of the Appendix provides an example implementation of this method for the simple case of an idealized bouncing ball.

Note that collisions may have the undesirable effect of imparting an external moment onto the spacecraft. Although this may be useful for translating stored angular momentum into lateral momentum in safety critical scenarios, it could also lead to saturation of the reaction wheels over time. As such, it may be desirable to minimize momentum transfer by constraining the relative velocity of the contact point to zero at the time of collision: $\zeta_i = 1 \Rightarrow v_{\mathrm{rel}_i} = 0$. Equivalently, from Lemma 3,

$$v_{x,i}+R\omega_i-M(1-\zeta_i)\leq 0 \wedge v_{x,i}+R\omega_i+M(1-\zeta_i)\geq 0, \quad i=2,\ldots,\tau \tag{21}$$

Note that meeting this condition preserves the initial tangential and angular velocity over the collision.

C. Representing Dynamics in the Presence of Polygonal Collision Surfaces

At the expense of introducing some complexity, we can generalize collision surfaces $\mathcal{S}$ from half-planes to convex polygons:

$$\bar{\mathcal{S}} \triangleq \{z \in \mathbb{R}^2 | \boldsymbol{a}_j^T z < b_j, j = 1,\ldots, N_{\bar{\mathcal{S}}}\} \tag{22}$$

where $N_{\bar{\mathcal{S}}}$ is the number of sides in the polygon and the indices j label the walls in a counterclockwise order. In contrast to the previous case, the basis vectors $\hat{i}$, $\hat{j}$ are not restricted to a particular orientation. We will assume that the vehicle collides into the jth wall of $\bar{\mathcal{S}}$ on iteration i if 1) the position at time step $i-1$ is closest to the boundary of the jth wall, and 2) the nominal update equation predicts that the vehicle will enter the interior of $\bar{\mathcal{S}}$ on iteration i. It is convenient to represent the second condition as $\tilde{\boldsymbol{s}}_i \triangleq \boldsymbol{A}_s \boldsymbol{x}_{i-1} + \boldsymbol{B}_s \boldsymbol{u}_{i-1} \in \bar{\mathcal{S}}$, where $\boldsymbol{A}_s$, $\boldsymbol{B}_s$ are the first two rows of $\boldsymbol{A}$, $\boldsymbol{B}$, corresponding to the position update under the nominal dynamics. Likewise we can represent the first condition as $\boldsymbol{s}_{i-1} \in \mathcal{C}_j$, where $\mathcal{C}_j$ is the region exterior to the polygon, closest to wall j. This can be defined as

$$\mathcal{C}_j \triangleq \{z \in \mathbb{R}^2 | \boldsymbol{\alpha}_j^T z < \beta_j, \boldsymbol{\alpha}_{\sigma(j)}^T z \geq \beta_{\sigma(j)}, \boldsymbol{a}_j^T z \geq b_j\} \tag{23}$$

where $\sigma(j)$ is the jth element of $\sigma \triangleq (N_{\bar{\mathcal{S}}}, 1, 2, \ldots, N_{\bar{\mathcal{S}}} - 1)$, and $\boldsymbol{\alpha}_j$, β_j define the half-space bisecting wall j and the next wall in the counterclockwise rotation, such that $\alpha_j^T z < \beta_j$ is satisfied for points closer to the jth edge. An example configuration is shown in Fig. 4.

Our goal is to define event variables $\Xi_{i,j} \in \{0, 1\}$ to indicate the occurrence of collisions. Specifically, we want $\Xi_{i,j} = 1$ when a collision occurs with the jth wall on iteration i,

$$\Xi_{i,j}=1 \Leftrightarrow \tilde{\boldsymbol{s}}_i \in \bar{\mathcal{S}} \wedge \boldsymbol{s}_{i-1} \in \mathcal{C}_j, \quad i=2,\ldots,\tau, \quad j=1,\ldots,N_{\bar{\mathcal{S}}} \tag{24}$$

These indicator variables may then be used to activate the collision dynamics for the wall involved in the collision. To indicate the position with respect to the walls $(\boldsymbol{a}_j, b_j)$ defining the polygon $\bar{\mathcal{S}}$, we introduce constraints:

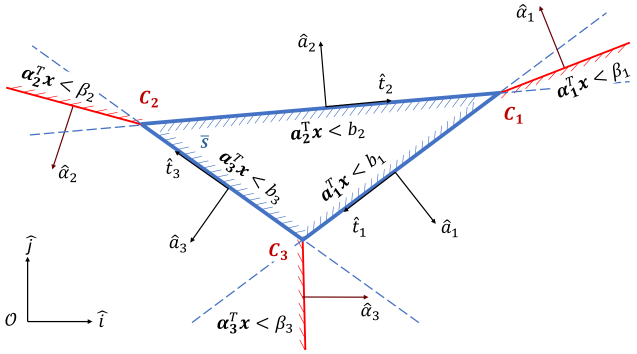
$\hat{\alpha}_1$
$\hat{a}_2$
$\boldsymbol{\alpha}_2^T\boldsymbol{x} < \beta_2$
$\hat{t}_2$
$\boldsymbol{C_2}$
$\boldsymbol{\alpha}_1^T\boldsymbol{x} < \beta_1$
$\boldsymbol{C_1}$
$\boldsymbol{a}_2^T\boldsymbol{x} < b_2$
$\bar{s}$
$\hat{\alpha}_2$
$\hat{t}_3$
$\boldsymbol{\alpha}_3^T\boldsymbol{x} < b_3$
$\boldsymbol{\alpha}_1^T\boldsymbol{x} < b_1$
$\hat{t}_1$
$\hat{a}_1$
$\hat{a}_3$
$\hat{j}$
$\boldsymbol{C_3}$
$\mathcal{O}$
$\hat{i}$
$\boldsymbol{\alpha}_3^T\boldsymbol{x} < \beta_3$
$\hat{\alpha}_3$

Fig. 4 Example geometry for triangular collision polygon $\bar{\mathcal{S}}$.

$$\begin{aligned} &\boldsymbol{a}_j^T \tilde{\boldsymbol{s}}_i + M\gamma_{i,j} \geq b_j \quad \wedge \quad \boldsymbol{a}_j^T \tilde{\boldsymbol{s}}_i - M(1-\gamma_{i,j}) < b_j, \\ &i = 2, \ldots, \tau, \qquad j = 1, \ldots, N_{\bar{\mathcal{S}}} \end{aligned} \tag{25}$$

which fixes $\gamma_{i,j} = 1$ exactly when the constraint $\boldsymbol{a}_j^T \tilde{\boldsymbol{s}}_i < b_j$ is satisfied (i.e., $\gamma_{i,j} = 1 \Leftrightarrow \boldsymbol{a}_j^T \tilde{\boldsymbol{s}}_i < b_j$). We can use constraints of the same form to indicate the position of the vehicle with respect to the half-spaces defined by (α_j, β_j):

$$\begin{aligned} &\alpha_j^T \boldsymbol{s}_i + M\lambda_{i,j} \geq \beta_j \quad \wedge \quad \boldsymbol{\alpha}_j^T \boldsymbol{s}_i - M(1-\lambda_{i,j}) < \beta_j \\ &i = 2, \ldots, \tau, \qquad j = 1, \ldots, N_{\bar{\mathcal{S}}} \end{aligned} \tag{26}$$

which expresses $\lambda_{i,j} = 1 \Leftrightarrow \boldsymbol{\alpha}_j^T \tilde{\boldsymbol{s}}_i < \beta_j$ for the appropriate values of i, j. The equivalencies in Eq. (24) can then be enforced by the constraints

$$\begin{aligned} &\sum_{p=1}^{N_{\bar{\mathcal{S}}}} \gamma_{i,p} + \lambda_{i-1,j} - \lambda_{i-1,\sigma(j)} - \gamma_{i-1,j} + M(1-\Xi_{i,j}) \geq N_{\bar{\mathcal{S}}} + 1 \\ &\wedge \sum_{p=1}^{N_{\bar{\mathcal{S}}}} \gamma_{i,p} + \lambda_{i-1,j} - \lambda_{i-1,\sigma(j)} - \gamma_{i-1,j} - M\Xi_{i,j} \leq N_{\bar{\mathcal{S}}} \end{aligned} \tag{27}$$

which are applied for $i = 2, \ldots, \tau$, $j = 1, \ldots, N_{\bar{\mathcal{S}}}$.

The dynamics for this system are then

$$\boldsymbol{x}_{i+1}-\boldsymbol{x}_i=\begin{cases} \boldsymbol{A}\boldsymbol{x}_i+\boldsymbol{B}\boldsymbol{u}_i & \text{if} \sum_{j=1}^{N_{\bar{\mathcal{S}}}} \Xi_{i+1,j} \leq 0 \\ \boldsymbol{A}_c^j \boldsymbol{x}_i + b_c^j & \text{if}\, \Xi_{i+1,j}=1,\, j=1,\ldots,N_{\bar{\mathcal{S}}}, \end{cases} \quad i=1,\ldots,\tau-1 \tag{28}$$

where $\boldsymbol{A}_c^j$, $\boldsymbol{b}_c^j$ are the collision dynamics for the jth wall. To represent these dynamics, let us first define a local frame for the jth wall $\mathcal{F}_w^j = (\mathcal{O}, \hat{t}_j, \hat{a}_j, \hat{k})$ with $\hat{t}_j$, $\hat{a}_j$ pointing tangent and normal to wall j, and $\hat{k}_j = \hat{t}_j \times \hat{a}_j$ upward. Each local frame is a rotation of $\mathcal{F}_I = (\mathcal{O}, \hat{i}, \hat{j}, \hat{k})$ about $\hat{k}$ by an angle ϕ_j. Let $\boldsymbol{L}_3(\phi_j) \in \mathbb{R}^{3\times 3}$ be the rotation matrix converting vectors in $\mathcal{F}_I$ to vectors in $\mathcal{F}_w^j$. Then we can express the collision dynamics for each wall in $\mathcal{F}_I$ by rotating the position and velocity vectors to and from this local frame:

$$\begin{aligned} &\boldsymbol{A}_c^j = \boldsymbol{\Lambda}_j \boldsymbol{A}_c \boldsymbol{\Lambda}_j^T \qquad \boldsymbol{b}_c^j = \boldsymbol{\Lambda}_j \left[0, -\kappa_N \frac{b_j}{\|\boldsymbol{a}_j\|_2}, 0, 0, 0, 0\right]^T \\ &\text{where,} \quad \boldsymbol{\Lambda}_j \triangleq \begin{bmatrix} \boldsymbol{L}_3(\phi_j) & \boldsymbol{0}_3 \\ \boldsymbol{0}_3 & \boldsymbol{L}_3(\phi_j) \end{bmatrix} \end{aligned} \tag{29}$$

The dynamics in Eq. (28) are then represented by the following MIP constraints:

$$\begin{aligned} &\boldsymbol{x}_{i+1}-\boldsymbol{x}_i-\boldsymbol{A}\boldsymbol{x}_i-\boldsymbol{B}\boldsymbol{u}_i-M\left(\sum_{j=1}^{N_{\bar{\mathcal{S}}}}\Xi_{i+1,j}\right)\leq \boldsymbol{0} \\ &\wedge \boldsymbol{x}_{i+1}-\boldsymbol{x}_i-\boldsymbol{A}\boldsymbol{x}_i-\boldsymbol{B}\boldsymbol{u}_i+M\left(\sum_{j=1}^{N_{\bar{\mathcal{S}}}}\Xi_{i+1,j}\right)\geq \boldsymbol{0} \\ &\boldsymbol{x}_{i+1}-\boldsymbol{x}_i-\boldsymbol{A}_c^j\boldsymbol{x}_i-\boldsymbol{b}_c^j-M\Xi_{i+1,j}\leq \boldsymbol{0} \\ &\wedge \boldsymbol{x}_{i+1}-\boldsymbol{x}_i-\boldsymbol{A}_c^j\boldsymbol{x}_i-\boldsymbol{b}_c^j+M\Xi_{i+1,j}\geq \boldsymbol{0}, \quad j=1,\ldots,N_{\bar{\mathcal{S}}} \end{aligned} \tag{30}$$

which are applied at $i = 1, \ldots, \tau - 1$.

D. Example Objective Functions

In practice, the appropriate choice of an objective function depends on the specific needs of the mission. The proposed methodology does not assume a particular form for the objective. However, because vehicle efficiency is commonly of critical importance to real-world missions, we find it useful to review here two common approximations for penalizing actuation using quadratic and linear functions. A simple option is to use the power limiting cost function [48,49],

$$J_1 = \sum_{i=1}^{\tau} (u_{x,i}^2 + u_{y,i}^2) \tag{31}$$

which forms a Mixed Integer Quadratic Program (MIQP). With the introduction of additional constraints, it is also possible to use a PWA approximation of the Euclidean norm of commanded translational acceleration [34]. The resulting cost function is linear:

$$J_2 = \sum_{i=1}^{\tau} G_i, \text{ s.t. } \bigwedge_{n=1}^{N_J} G_i \geq u_{x,i} \sin\left(\frac{2\pi n}{N_J}\right) + u_{y,i} \cos\left(\frac{2\pi n}{N_J}\right),\ i = 1,\ldots,\tau \tag{32}$$

The constraints here approximate the second-order cone constraints $G_i \geq \|[u_{x,i}, u_{y,i}]\|_2$, $i = 1, \ldots, \tau$ with an N_J-sided polygon.

V. Applications and Case Studies

The potential benefits of the proposed approach are demonstrated through two case studies. First, the problem formulation developed in Sec. IV is validated on hardware. The effectiveness of the approach in improving upon a chosen objective function is studied through a comparison between collision-inclusive and collision-free trajectories. Results are tabulated for both ideal and experimental cases. Next, a simulated example provides a qualitative demonstration of how the collision-inclusive planner may be applied in safety-critical applications.

A. Experimental Performance Comparison

1. Description of Hardware and Testbed

Experiments were conducted for this work in the Stanford Space Robotics Facility on the free-flyer spacecraft robot testbed. A set of robots is designed to hover frictionlessly on air bearings, thus emulating microgravity dynamics in the plane of a table. Though previous generations of the free-flyer robot used in this experiment operated on compressed air [50], the current iteration of the free-flyer operates on CO_2, owing to CO_2's ability to be stored in liquid form at room temperature at only 1000 psi, resulting in a much higher fuel density than can be achieved at comparable pressures with compressed air. The robots are also equipped with actuators commonly used in spacecraft, namely, a reaction wheel for attitude control and eight cold-gas thrusters primarily for translational control. Because of high capacity of the CO_2 tanks, the robots can perform aggressive thrust maneuvers for over an hour and can hover without thrust for over 10 h continuously.

The robots use an Odroid XU4 for its primary onboard computation, as well as an mbed Microcontroller for low-level control of various subsystems. Additionally, the free-flyer software stack is implemented in ROS and is connected to an off-board hub computer, where more heavy computation can be run as needed for planning and control. The ROS stack also gives access to real-time data from a motion-capture system, giving position and velocity information at 120 Hz. The granite table used for experiments is $9' \times 12'$—approximately 2.74 m × 3.66 m—allowing ample room for complex planning scenarios. Further parameters for the free-flyer robot can be found in Table 1, where average mass is reported due to variations in the state of the tanks.

Table 1 Free-flyer spacecraft parameters

Parameter	Value	Unit
Average mass, m	18.08	kg
Radius, R	0.157	m
Max individual thruster output, $u_{T,\max}$	0.20	N
Body inertia about spin axis, I_b	0.184	$\mathrm{kg \cdot m^2}$
Reaction wheel inertia, I_w	0.029	$\mathrm{kg \cdot m^2}$
Max acceleration of reaction wheel	0.628	$\mathrm{rad/s^2}$
Reaction wheel speed range	60–340	RPM

2. Performance Comparison

The spacecraft and testbed described in the above section are considered with $\mathcal{S}$ taken as the lower wall of the testbed and the origin of $\mathcal{F}_I$ at the lower left corner, as shown in Figs. 1 and 5.

We now compare the performance of vehicles navigating from rest at initial position $s_1 = [0.41\ \mathrm{m}, 2.29\ \mathrm{m}]^T$ to rest at final position $s_\tau = [3.15\ \mathrm{m}, 2.29\ \mathrm{m}]^T$, while remaining in the boundary of the testbed, and avoiding a central rectangular region $\mathcal{P} = \{z \in \mathbb{R}^2 | z_1 \in [1.45, 2.12], z_2 \in [0.57, 2.74]\}$. We minimize the cost J_1 introduced in Eq. (31). The performances of the vehicle are compared both in terms of this approximation, and a fuel cost measured through pulse width modulation (PWM) signals sent to the thruster. Assuming constant mass flow rate through the thrusters, the latter cost is directly proportional to fuel consumption. The relative velocity of the contact point is constrained to zero in order to minimize angular momentum transfer with the wall [Eq. (21)]. A small penalty on angular velocity is also included to reduce unnecessary spin of the spacecraft, which, due to the limited update rate of the thrust controller—approximately 2 Hz—may diminish the accuracy of acceleration commands.

The trajectory is generated with Gurobi optimization software using the formulation in Sec. IV with the parameters listed above. The thrust saturation constraint Eq. (5) uses $N_U = 20$ sides in the approximation. To ensure that regulation is possible in the presence of disturbances, the MIP limits the maximum acceleration ($u_{\max}$) to 90% of its theoretical value. The ideal state is tracked using a linear quadratic regulator (LQR) as the ancillary control law. The net control at time $t \in \mathbb{R}$ is

$$\boldsymbol{u}(t) = \boldsymbol{u}^*(t) + \boldsymbol{K}_{\mathrm{lqr}}(\boldsymbol{x}^*(t) - \boldsymbol{x}(t)) \tag{33}$$

where $\boldsymbol{K}_{\mathrm{lqr}} = [2.86\boldsymbol{I}_3, 14.43\boldsymbol{I}_3] \in \mathbb{R}^{3\times6}$ is the LQR gain matrix, and $\boldsymbol{u}^* \in \mathbb{R}^3, \boldsymbol{x}^* \in \mathbb{R}^6$ are the ideal control and state at time t, taken from a polynomial interpolation of the control and state solutions returned from the MIP. The input $\boldsymbol{u}$ is then mapped to PWM signals on the thrusters $\boldsymbol{u}_{\mathrm{pwm}} = \boldsymbol{u}_T / u_{T,\max} \in [0, 1]^8$. This mapping is derived by taking the pseudo-inverse of the mapping from individual thruster forces to forces in the body frame. The resulting mixing equation is balanced in the body frame, ensuring that no moment is produced from the thrusters. An inner PID loop regulates the speed of the reaction wheel, which is used to achieve the desired moment.

Experiments are conducted for this scenario with the time horizons fixed to 45 and 60 s. The experimental update rate of the controller varies slightly from the fixed 0.5 s period assumed in the planning phase. As a consequence of this, the 45 s experiments are both completed in 82 steps, and the collision-free and collision-inclusive 60 s experiments are completed in 108 and 109 steps, respectively. The trajectories taken are shown in Fig. 5, and the efficiency measures are plotted against time in Fig. 6. A video comparison of two experiments may be found in [51]. Note that despite having a significant effect on the total cost, the difference in time allocated to reach the goal has virtually no effect on the shape of the planned path. Table 2 shows the total costs for each experiment, along with the corresponding ideal values, and the resulting PWM costs. It is apparent that the collision-inclusive approach is capable of demonstrating significant improvements in overall efficiency for a given time horizon. In particular, we see reductions in the J_1 cost of 44.3 and 22.9% for the 45 and 60 s experiments, respectively (compared with 47.8 and 41.2% for the ideal case), and reductions of 31.7 and 23.8% in the PWM cost for the 45 and 60 s experiments, respectively. The main boost in efficiency occurs midway through the trajectory. As the collision-free vehicle requires increased thrust to reduce its velocity and redirect its momentum, the collision-inclusive approach allows the spacecraft to minimize its thrust at this point, gaining the required momentum transfer directly from an impulsive force at the wall. There appears to be some tradeoff when using this approach in that a spike in thrust is seen to occur directly after collision.

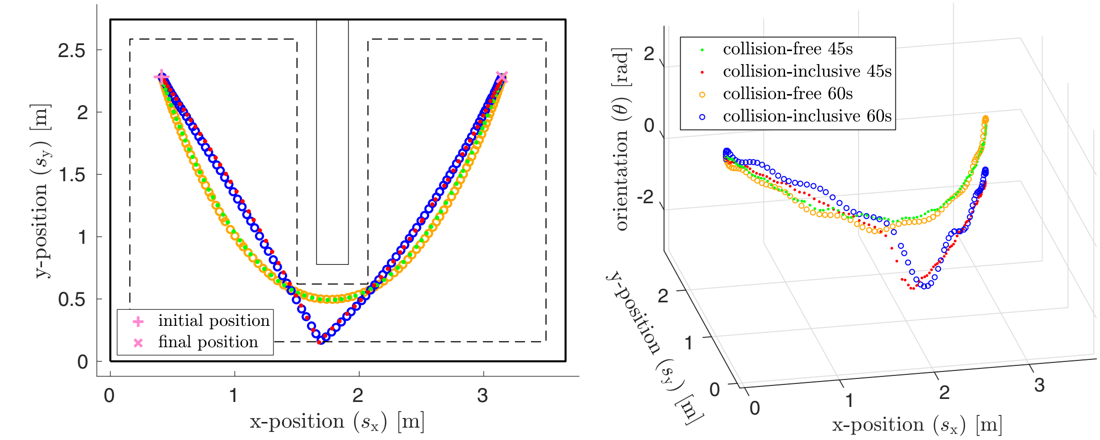

Fig. 5 Comparison of paths taken in experimental scenarios.

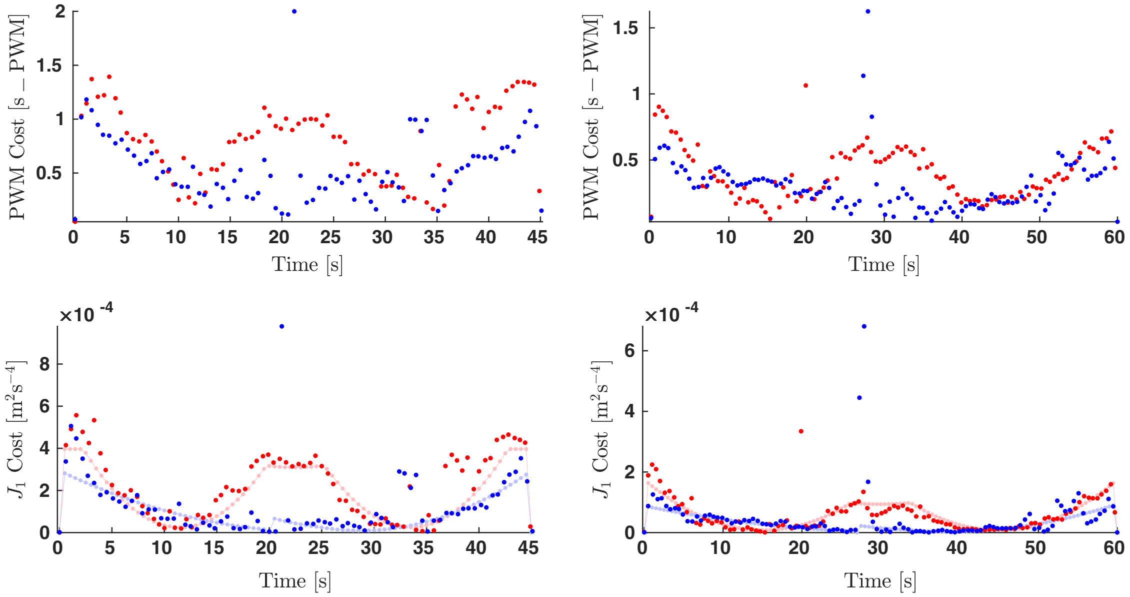

Fig. 6 Costs vs time for collision-free (red) and collision-inclusive (blue) experiments, and corresponding ideal values (transparent) for the J_1 cost.

This might be attributed to a number of factors that could potentially lead to increased model error on the collision iteration. For example, sensitivity to modeling the precise location of the wall (δ) or vehicle radius (R); to precisely matching the commanded tangential and angular velocities at the time of collision; or from the zero thrust approximation made in the update equations.

B. Application to Safety-Critical Systems

If collision avoidance is posed as a hard constraint in the problem formulation, then online MPC becomes vulnerable to being rendered infeasible in situations where collisions can no longer be avoided. This situation might arise from a number of factors, including model error, external perturbations, or movement of objects in the environment. The problem is exacerbated by the tendency of optimal trajectories to lie near the boundary of the infeasible region (see, e.g., Fig. 5). On the other hand, if the constraint to avoid collisions is replaced by a term in the cost function capturing the damage from this event, and the effects of the collision are considered in the constraints, then the planner can not only remain feasible, but also direct the vehicle toward an *optimal mitigating action*. In this section we turn to a simulated scenario to demonstrate the potential of the collision-inclusive planner to bring about enhanced safety in this sense.

Table 2 Experiment cost values

Specification	Experimental J_1 cost, m^2/s^4	Ideal J_1 cost, m^2/s^4	Experimental PWM cost, s
Collision-free, 45 s	0.01822	0.01619	65.04
Collision-inclusive, 45 s	0.01014	0.00845	44.42
Collision-free, 60 s	0.00633	0.00611	43.76
Collision-inclusive, 60 s	0.00488	0.00359	33.33

Consider the spacecraft and parameters as described in previous sections, now given the task of traversing across a larger, more cluttered environment, consisting of a number of walls whose locations are known to it via an internal map. The vehicle also performs online sensing, which it may use to detect unmapped objects in the environment. Collision with the walls is known to cause minor damage to the robot, whereas collision with a newly detected object is considered more damaging, as neither the type of object nor consequences of hitting it are known in advance. We now compare the two strategies for this case in the environment shown in Fig. 7. Here the spacecraft (with parameters from previous sections) is given the goal of reaching the point in the top right corner while avoiding obstacles. The vehicle starts on the green trajectory shown in Fig. 7; however, the vehicle eventually detects the presence of a new obstacle (red box) obstructing the original path and is not able to stop in time to prevent collision. If collision avoidance with the obstacles is posed as a hard constraint, then the MPC is rendered infeasible. Without an update, the vehicle may simply continue on its original course and hit the object at high velocity. A more thoughtful implementation might include a backup controller that brings the vehicle to rest as quickly as possible; however, even this backup strategy will result in inevitable and uncontrolled collisions with both the wall and obstacle [43].

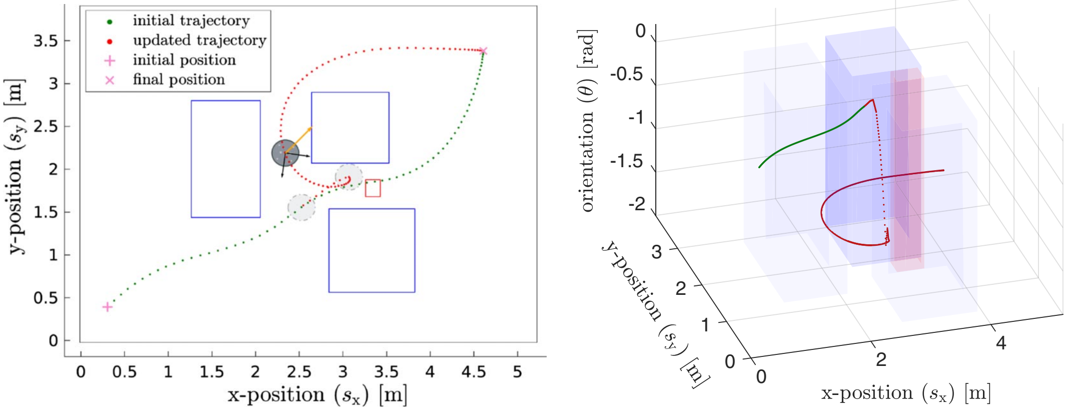

Fig. 7 Original (green) and updated (red) paths taken to evade observed obstacle (left) [52]; composite trajectory (right).

As an alternative strategy, we may incorporate the presence of the various objects in the program through penalties in a multi-objective cost function. The total cost J will be taken as the sum of some nominal cost J_{nom} (e.g., fuel consumption) and a damage cost J_{dam}. Here we will take the damage cost to be a weighted sum of the speeds at the time of impact, defined for the jth wall on the ith iteration as follows:

$$v_{i,j}^{\text{impact}} \triangleq -\Xi_{i,j}[v_{x,i}, v_{y,i}] \frac{\boldsymbol{a}_j}{\|\boldsymbol{a}_j\|_2} \tag{34}$$

where $\Xi_{i,j} = 1$ indicates collision with wall defined by $\boldsymbol{a}_j$ on iteration i. Note that impact speed is a quadratic function of the problem variables. Letting σ_1 be the set of indices for the mapped walls (blue) and σ_2 be the indices for the unmapped walls (red). Then we can express the total cost as $J = J_{\text{nom}} + J_{\text{dam}}$ with

$$J_{\text{dam}} \triangleq \sum_{i=2}^{\tau} \sum_{j \in \sigma_1} K_1 v_{i,j}^{\text{impact}} + \sum_{i=2}^{\tau} \sum_{j \in \sigma_2} K_2 v_{i,j}^{\text{impact}} \tag{35}$$

where K_1, $K_2 \in \mathbb{R}$ weight the collision penalties for each type of object. For this situation, we specify that K_1 is much less than K_2, directing to vehicle to avoid the unknown object as much as possible. The red path in Fig. 7 shows the new trajectory that is calculated using this cost function once the red box is first detected—that is, accounted for in the motion planning. Here the spacecraft is able to leverage the collision dynamics with the blue box to avoid collision with the unknown obstacle altogether. In addition to simply applying thrust to push itself away from the object, the vehicle increases its angular speed before the collision and uses stored angular momentum to push itself away on impact. Additional simulation parameters are listed in Sec. A of the Appendix. A video of the simulation may be found at [52].

VI. Limitations of the Proposed Approach and Future Work

The formulation may be further developed through consideration of more complex environments or vehicle geometries, either of which may require the construction of a more complex collision model. The results of the trajectory planning may be sensitive to the accuracy of the collision model and other parameters. New regulation strategies may be explored to reduce the loss in efficiency in the time samples surrounding the collision, and to make the strategy more robust to parametric uncertainty. The basic framework for optimizing around collisions may be adapted to applications such as docking or landing, or alternate vehicle platforms such as quad-copters. A key focus of future research will be to expand on the applications toward safety though the creation and implementation of damage-minimizing backup controllers. Though the proposed framework is capable of generating the necessary trajectories, the current implementation of the MPC may not be able to generate the aggressive paths quickly enough to be practical for online use. To make this more tractable, a learning approach may be used to approximate the policy of the collision-inclusive backup controller. Once this has been achieved, the backup strategy may be validated in an experimental scenario. As an alternate safety application, the enhanced maneuverability of the collision-inclusive MPC may be leveraged in a controlled set invariance framework such as the one presented in [53,54].

If the proposed approach is to be applied in orbit, and over longer timescales, then one may consider replacing the double-integrator model with Clohessy–Wiltshire–Hill (CWH) dynamics [12,55]. Because this model is linear, the nominal dynamics may be incorporated directly into the form presented in Eqs. (15) and (28). In this case, one would either need to modify the collision update equations to account for the new dynamics, or for the errors introduced if the perturbation is neglected between the time steps surrounding the collision. A general extension of the planner to the six-degree-of-freedom case may potentially be challenging. However, it is important to note that for planar (2-D) and spatial (3-D) operations, the coupling terms between the attitude and translational dynamics disappear during collision if the relative velocity at the point of contact goes to zero. Hence, one could in principle use this approach to plan trajectories for the translational states, and delegate the task of rotating the spacecraft to match the relative velocity of the wall to a separate, lower-level controller. Similarly, one might simplify the spatial case formulation in a way that allows some of the translation–attitude coupling to be leveraged by constraining the robot to orient its rotation axis to be orthogonal to the direction of relative velocity on collision iterations.

VII. Conclusions

This paper both introduces and validates the idea of optimizing spacecraft trajectories comprising planned collisions. The main theoretical contribution consists of formulating the collision-inclusive trajectory planner as a mixed integer programming problem. Experiments comparing the efficiency of collision-free and collision-inclusive trajectories provide a proof-of-concept demonstration of the approach's capability to bring about practical performance enhancements. Moreover, a simulated case study shows the potential for application of the method as an online safety measure. Though modeling of the collision in the constraints and penalizing a metric of damage in the cost, the vehicle is able to find novel solutions to mitigate scenarios where collisions are inevitable. Tradeoffs appear to be present in the added complexity of the problem and its apparent sensitivity to model error over iterations surrounding the collisions. Future work will focus on generalizing the approach to the spatial case, developing more robust regulation strategies, and extending the practicality of online safety applications.

Appendix A: Parameters Used in Simulated Safety Scenario

The vehicle navigates from the initial state $\boldsymbol{x}_1^T = [0.30\,\text{m}, 0.40\,\text{m}, 0.00\,\text{rad}, 0.15\,\text{m/s}, 0.15\,\text{m/s}, 0.00\,\text{rad/s}]$, to the final

Table A1 Location of boxes in safety scenario

$x_{\min}$, m	$y_{\min}$, m	$x_{\max}$, m	$y_{\max}$, m	Index
1.20	1.40	2.00	2.80	1
2.80	0.50	3.80	1.50	2
2.60	2.05	3.50	2.90	3
3.23	1.65	3.43	1.85	4

state $\boldsymbol{x}_\tau^T = [4.60\,\mathrm{m}, 3.40\,\mathrm{m}, 0.00\,\mathrm{rad}, 0.0\,\mathrm{m/s}, 0.00\,\mathrm{m/s}, 0.00\,\mathrm{rad/s}]$. The time resolution of the simulation is 0.06 s. The collision penalty weights are $(K_1, K_2) = (10, 10e6)$. The vehicle is initially commanded to reach its goal location in 25 s, and the time horizon is appended by 10 s until a feasible solution is found. The collision and vehicle parameters are taken as those found for the real vehicle. The locations of the boxes are listed in Table A1. Indices 1–3 correspond to the boxes that were previously mapped out by the object, and box 4 is the box that is discovered by the vehicle. This obstacle is observed at $t = 12.37$ s from the start of the simulation.

$$\boldsymbol{x}_{i+1} - \boldsymbol{x}_i = \begin{cases} \boldsymbol{A}_c \boldsymbol{x}_i & \text{if } s_{y,i} + v_{y,i}\Delta t - 0.5g\Delta t^2 < 0 \\ \boldsymbol{A}\boldsymbol{x}_i + \boldsymbol{b} & \text{if } s_{y,i} + v_{y,i}\Delta t - 0.5g\Delta t^2 \geq 0 \end{cases}$$

where

$$\boldsymbol{A}_c = \begin{bmatrix} \kappa_N & (1+\kappa_N)\Delta t \\ 0 & \kappa_N \end{bmatrix}, \qquad \boldsymbol{A} = \begin{bmatrix} 0 & \Delta t \\ 0 & 0 \end{bmatrix},$$

$$\boldsymbol{b} = \begin{bmatrix} -0.5g\Delta t^2 \\ -g\Delta t \end{bmatrix}, \quad \boldsymbol{x}_i = \begin{bmatrix} s_{y,i} \\ v_{y,i} \end{bmatrix}$$

and Δt is the time step. Note that, because this model is uncontrolled, the objective function plays no role here. The set of constraints used to model these dynamics is given as follows.

First the condition that collision is predicted to occur on the next iteration can be associated with the event variable $\zeta \in \{0, 1\}$, leaving us with $\zeta_i = 1 \Leftrightarrow s_{N,i} + v_{N,i}\Delta t - 0.5g\Delta t^2 < 0$. This is transformed to the constraints

$$\begin{aligned} s_{y,i} + v_{y,i}\Delta t + M\zeta_{i+1} - g0.5\Delta t^2 &\geq 0 && ([s_{y,i} + v_{y,i}\Delta t - g0.5\Delta t^2 \geq 0] \Rightarrow [\zeta_{i+1} = 1]) \\ s_{y,i-1} + v_{y,i-1}\Delta t - M(1 - \zeta_i) - g0.5\Delta t^2 &< 0 && ([s_{y,i} + v_{y,i}\Delta t - g0.5\Delta t^2 < 0][\zeta_{i+1} \Leftarrow 1]) \end{aligned}$$

with $i = 1, \ldots, \tau - 1$. Next, the nominal position and velocity dynamics constraints are

$$\begin{aligned} s_{y,i+1} - s_{y,i} - \Delta t v_{y,i} + M\zeta_{i+1} &\geq 0.5g\Delta t^2 && ([\Delta s_{y,i+1} \geq \Delta t v_{y,i} - 0.5g\Delta t^2] \Rightarrow [\zeta_{i+1} = 0]) \\ s_{y,i+1} - s_{y,i} - \Delta t v_{y,i} - M\zeta_{i+1} &\leq 0.5g\Delta t^2 && ([\Delta s_{y,i+1} \leq \Delta t v_{y,i} - 0.5g\Delta t^2] \Leftarrow [\zeta_{i+1} = 0]) \\ v_{y,i+1} - v_{y,i} + M\zeta_{i+1} &\geq -g\Delta t && ([\Delta v_{y,i+1} \geq -g\Delta t] \Rightarrow [\zeta_{i+1} = 0]) \\ v_{y,i+1} - v_{y,i} - M\zeta_{i+1} &\leq -g\Delta t && ([\Delta v_{y,i+1} \leq -g\Delta t] \Leftarrow [\zeta_{i+1} = 0]) \\ s_{y,i+1} - (1+\kappa_N)s_{y,i} - (1+\kappa_N)\Delta t v_{y,i} - M\zeta_{i+1} &\geq -M && ([\Delta s_{y,i+1} \geq \kappa_N s_{y,i} + (1+\kappa_N)\Delta t v_{y,i}] \Rightarrow [\zeta_{i+1} = 1]) \\ s_{y,i+1} - (1+\kappa_N)s_{y,i} - (1+\kappa_N)\Delta t v_{y,i} + M\zeta_{i+1} &\leq M && ([\Delta s_{y,i+1} \leq \kappa_N s_{y,i} + (1+\kappa_N)\Delta t v_{y,i}] \Leftarrow [\zeta_{i+1} = 1]) \\ v_{y,i+1} - (1+\kappa_N)v_{y,i} - M\zeta_{i+1} &\geq -M && ([\Delta v_{y,i+1} \geq \kappa_N v_{y,i}] \Rightarrow [\zeta_{i+1} = 1]) \\ v_{y,i+1} - (1+\kappa_N)v_{y,i} + M\zeta_{i+1} &\leq M && ([\Delta v_{y,i+1} \leq \kappa_N v_{y,i}] \Leftarrow [\zeta_{i+1} = 1]) \end{aligned}$$

Appendix B: Example: 1-D Modeling of Bouncing Particle with MIP

Here we consider the simple case of a free-floating particle being released from rest at $s_y = 1$ m and pulled down at gravitational acceleration $g = 9.8\ \mathrm{m/s^2}$ until it experiences a collision with the ground at $s_y = 0$. The particle responds to the collision according to the update law $\Delta v_y^+ = \kappa_N v_y^-$, where $\kappa_N = -1.55$ and Δv_y^+, v_y^- indicate the change in velocity, and velocity immediately before collision, respectively. We can model the motion of the particle with

which are applied at $i = 1, \ldots, \tau - 1$. The system is simulated with an MIP using $\Delta t = 0.01$, and $\tau = 150$. The solution is plotted in Fig. B1. Points found with the collision update equation are shown in red, and points found with the nominal update equation are shown in blue.

Appendix C: Truth Tables for Lemmas 1 and 2

As shown in Table C1, Lemmas 1 and 2 may be conveniently expressed in the form of a truth table. The constant M is assumed to

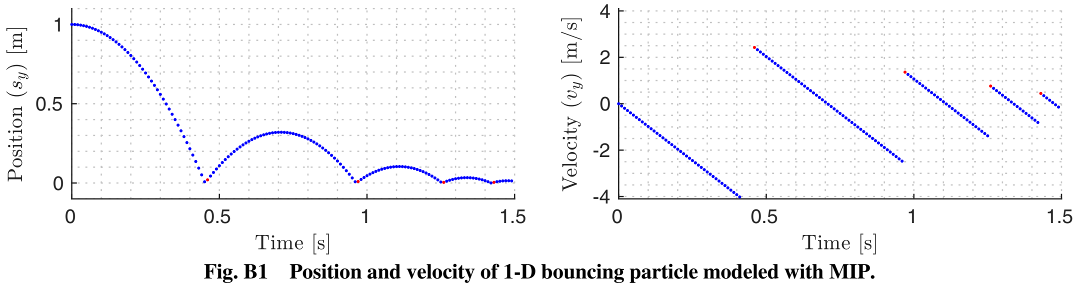

Fig. B1 Position and velocity of 1-D bouncing particle modeled with MIP.

Table C1 Truth table for Lemmas 1 and 2

$\mathbf{c}^T z < d$	$\zeta = 1$	$\mathbf{c}^T z + M\zeta \geq d$	$\mathbf{c}^T z - M(1-\zeta) < d$	$\mathbf{c}^T z < d \Rightarrow \zeta = 1$	$\zeta = 1 \Rightarrow \mathbf{c}^T z < d$
T	T	T	T	T	T
T	F	F	T	F	T
F	T	T	F	T	F
F	F	T	T	T	T

have a value satisfying the appropriate assumptions from these lemmas over the variable domains.

Acknowledgments

This research was supported in part by the National Science Foundation under CPS award 1931815, award 1544332, by an Early Stage Innovations grant from NASA's Space Technology Research Grants Program, by the King Abdulaziz City for Science and Technology (KACST), and by King Abdullah University of Science and Technology (KAUST) baseline fund.

References

[1] Pedersen, L., Kortenkamp, D., Wettergreen, D., Nourbakhsh, I., and Korsmeyer, D., "A Survey of Space Robotics," *7th International Symposium on Artificial, Intelligence, Robotics, and Automation in Space*, Citeseer, 2003.

[2] Fredrickson, S. E., Abbott, L. W., Duran, S., Jochim, J. D., Studak, J. W., Wagenknecht, J. D., and Williams, N. M., "Mini AERCam: Development of a Free-Flying Nanosatellite Inspection Robot," *Space Systems Technology and Operations*, Vol. 5088, International Soc. for Optics and Photonics, Society of Photo-Optical Instrumentation Engineers (SPIE), 2003, pp. 97–112.
https://doi.org/10.1117/12.498108

[3] Williams, T., and Tanygin, S., "On-Orbit Engineering Tests of the AERCam Sprint Robotic Camera Vehicle," *Spaceflight Mechanics*, 1998, pp. 1001–1020.

[4] Boyd, I., Buenconsejo, R., Piskorz, D., Lal, B., Crane, K., and De La Rosa Blanco, E., "On-Orbit Manufacturing and Assembly of Spacecraft," Inst. for Defense Analysis, IDA Paper P-8335, Alexandria, VA, 2017.

[5] Bylard, A., MacPherson, R., Hockman, B., Cutkosky, M. R., and Pavone, M., "Robust Capture and Deorbit of Rocket Body Debris Using Controllable Dry Adhesion," *Aerospace Conference*, IEEE, New York, 2017, pp. 1–9.
https://doi.org/10.1109/aero.2017.7943844

[6] Acquatella, P., "Development of Automation and Robotics in Space Exploration," *Proceedings of the AIAA SPACE 2009 Conference and Exposition*, Citeseer, 2009, pp. 1–7.

[7] Bualat, M., Barlow, J., Fong, T., Provencher, C., and Smith, T., "Astrobee: Developing a Free-flying Robot for the International Space Station," *AIAA SPACE 2015 Conference and Exposition*, AIAA Paper 2015-4643, 2015.
https://doi.org/10.2514/6.2015-4643

[8] Miller, D., Saenz-Otero, A., Wertz, J., Chen, A., Berkowski, G., Brodel, C., Carlson, S., Carpenter, D., Chen, S., and Cheng, S., et al., "SPHERES: A Testbed for Long Duration Satellite Formation Flying in Micro-Gravity Conditions," *Proceedings of the AAS/AIAA Space Flight Mechanics Meeting*, Vol. 105, Advances in the Astronautical Sciences, Clearwater, FL, Jan. 2000, pp. 167–179.

[9] Jewison, C., and Miller, D. W., "Probabilistic Trajectory Optimization Under Uncertain Path Constraints for Close Proximity Operations," *Journal of Guidance, Control, and Dynamics*, Vol. 41, No. 9, 2018, pp. 1843–1858.
https://doi.org/10.2514/1.G003152

[10] Chamitoff, G. E., Saenz-Otero, A., Katz, J. G., Ulrich, S., Morrell, B. J., and Gibbens, P. W., "Real-Time Maneuver Optimization of Space-Based Robots in a Dynamic Environment: Theory and On-Orbit Experiments," *Acta Astronautica*, Vol. 142, Jan. 2018, pp. 170–183.
https://doi.org/10.1016/j.actaastro.2017.10.001

[11] Morrell, B., "Enhancing 3D Autonomous Navigation Through Obstacle Fields: Homogeneous Localisation and Mapping, with Obstacle-Aware Trajectory Optimisation," Univ. of Sydney, 2018.

[12] LeValley, A. S., "A Mixed Integer Programming Framework for the Fuel Optimal Guidance of Complex Spacecraft Rendezvous and Proximity Operation Missions," Air Force Inst. of Technology, AFIT-ENV-MS-19-M-185, Wright-Patterson AFB, OH, 2019.

[13] Rehnmark, F., Bluethmann, W., Mehling, J., Ambrose, R. O., Diftler, M., Chu, M., and Necessary, R., "Robonaut: The 'Short List' of Technology Hurdles," *Computer*, Vol. 38, No. 1, 2005, pp. 28–37.
https://doi.org/10.1109/MC.2005.32

[14] Virgili-Llop, J., and Romano, M., "ASTROBATICS: Demonstrating Propellantless Robotic Maneuvering Onboard the International Space Station," Naval Postgraduate School, Tech. Rept., Monterey, CA, 2018, http://hdl.handle.net/10945/58053.

[15] Alsup, K. P., "Robotic Spacecraft Hopping: Application And Analysis," Naval Postgraduate School, OMB 0704-0188, Monterey, CA, 2018.

[16] Xi, W., and Remy, C. D., "Optimal Gaits and Motions for Legged Robots," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014)*, IEEE, New York, 2014, pp. 3259–3265.
https://doi.org/10.1109/iros.2014.6943015

[17] Giardina, F., and Iida, F., "Efficient and Stable Locomotion for Impulse-Actuated Robots Using Strictly Convex Foot Shapes," *IEEE Transactions on Robotics*, Vol. 34, No. 3, 2018, pp. 674–685.
https://doi.org/10.1109/TRO.2018.2813359

[18] Serra, D., Satici, A. C., Ruggiero, F., Lippiello, V., and Siciliano, B., "An Optimal Trajectory Planner for a Robotic Batting Task: The Table Tennis Example," *International Conference on Informatics in Control, Automation and Robotics (2)*, Vol. 105, Elsevier, July 2018, pp. 121–137.
https://doi.org/10.5220/0005982000900101

[19] Spong, M. W., "Impact Controllability of an Air Hockey Puck," *Systems and Control Letters*, Vol. 42, No. 5, 2001, pp. 333–345.
https://doi.org/10.1016/S0167-6911(00)00105-5

[20] Xin, W., Hourdos, J., Michalopoulos, P., and Davis, G., "The Less-Than-Perfect Driver: A Model of Collision-Inclusive Car-Following Behavior," *Transportation Research Record*, Vol. 2088, No. 1, 2008, pp. 126–137.
https://doi.org/10.3141/2088-14

[21] Kiefer, J., Ward, M., and Costello, M., "Rotorcraft Hard Landing Mitigation Using Robotic Landing Gear," *Journal of Dynamic Systems, Measurement, and Control*, Vol. 138, No. 3, 2016, Paper 031003.
https://doi.org/10.1115/1.4032286

[22] Lee, S. H., Yi, B.-J., Kim, S. H., and Kwak, Y. K., "Analysis on Impact Propagation of Docking Platform for Spacecraft," *IEEE International Conference on Robotics and Automation, Proceedings 2001 ICRA*, Vol. 1, IEEE, New York, 2001, pp. 413–420.
https://doi.org/10.1109/robot.2001.932586

[23] Estrada, M. A., Hockman, B., Bylard, A., Hawkes, E. W., Cutkosky, M. R., and Pavone, M., "Free-flyer Acquisition of Spinning Objects with Gecko-Inspired Adhesives," *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, New York, 2016, pp. 4907–4913.
https://doi.org/10.1109/icra.2016.7487696

[24] Hockman, B. J., Frick, A., Reid, R. G., Nesnas, I. A., and Pavone, M., "Design, Control, and Experimentation of Internally-Actuated Rovers for the Exploration of Low-gravity Planetary Bodies," *Journal of Field Robotics*, Vol. 34, No. 1, 2017, pp. 5–24.
https://doi.org/10.1002/rob.2017.34.issue-1

[25] Posa, M., Tobenkin, M., and Tedrake, R., "Stability Analysis and Control of Rigid-Body Systems with Impacts and Friction," *IEEE Transactions on Automatic Control*, Vol. 61, No. 6, 2016, pp. 1423–1437.
https://doi.org/10.1109/TAC.2015.2459151

[26] Mulgaonkar, Y., Makineni, A., Guerrero-Bonilla, L., and Kumar, V., "Robust Aerial Robot Swarms Without Collision Avoidance," *IEEE Robotics and Automation Letters*, Vol. 3, No. 1, 2018, pp. 596–603.
https://doi.org/10.1109/LRA.2017.2775699

[27] Mayya, S., Pierpaoli, P., Nair, G., and Egerstedt, M., "Localization in Densely Packed Swarms Using Interrobot Collisions as a Sensing Modality," *IEEE Transactions on Robotics*, Vol. 35, No. 1, 2019, pp. 21–34.
https://doi.org/10.1109/TRO.2018.2872285

[28] Connell, S., "Crazy Cat [Video]," 2010, https://www.youtube.com/watch?v=7vak9Vi4e44.

[29] Mote, M., Afman, J. P., and Feron, E., "Robotic Trajectory Planning Through Collisional Interaction," *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, IEEE, New York, 2017, pp. 1144–1149.
https://doi.org/10.1109/cdc.2017.8263810

[30] Seto, D., Krogh, B. H., Sha, L., and Chutinan, A., "Dynamic Control System Upgrade Using the Simplex Architecture," *IEEE Control Systems Magazine*, Vol. 18, No. 4, 1998, pp. 72–80. https://doi.org/10.1109/37.710880

[31] Kerrigan, E. C., and Maciejowski, J. M., "Soft Constraints and Exact Penalty Functions in Model Predictive Control," *Proceedings of UKACC International Conference*, 2000.

[32] Bemporad, A., and Morari, M., "Control of Systems Integrating Logic, Dynamics, and Constraints," *Automatica*, Vol. 35, No. 3, 1999, pp. 407–427. https://doi.org/10.1016/S0005-1098(98)00178-2

[33] Lubin, M., Zadik, I., and Vielma, J. P., *Mixed-integer Convex Representability*, Lecture Notes in Computer Science, edited by F. Eisenbrand, and J. Koenemann, Integer Programming and Combinatorial Optimization, Vol. 10328, Springer, Cham, 2017, pp. 392–404. https://doi.org/10.1007/978-3-319-59250-3_32

[34] Richards, A., Schouwenaars, T., How, J. P., and Feron, E., "Spacecraft Trajectory Planning with Avoidance Constraints Using Mixed-Integer Linear Programming," *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 4, 2002, pp. 755–764. https://doi.org/10.2514/2.4943

[35] Mellinger, D., Kushleyev, A., and Kumar, V., "Mixed-Integer Quadratic Program Trajectory Generation for Heterogeneous Quadrotor Teams," *2012 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, New York, 2012, pp. 477–483. https://doi.org/10.1109/icra.2012.6225009

[36] Lorenzetti, J., Chen, M., Landry, B., and Pavone, M., "Reach-Avoid Games via Mixed-integer Second-order Cone Programming," *2018 IEEE Conference on Decision and Control (CDC)*, IEEE, New York, 2018, pp. 4409–4416. https://doi.org/10.1109/cdc.2018.8619382

[37] Del Pia, A., Dey, S. S., and Molinaro, M., "Mixed-Integer Quadratic Programming Is in NP," *Mathematical Programming*, Vol. 162, Nos. 1–2, 2017, pp. 225–240. https://doi.org/10.1007/s10107-016-1036-0

[38] Cplex, I. I., "V12. 1: User's Manual for CPLEX," *International Business Machines Corporation*, Vol. 46, No. 53, 2009, p. 157.

[39] "Gurobi Optimizer Reference Manual, 2015," Gurobi Optimization, Inc., 2014, http://www.gurobi.com.

[40] Mosek, A., "The MOSEK Optimization Software," Vol. 54, Nos. 1–2, 2010, p. 5, http://www.mosek.com.

[41] Hillier, F. S., Lieberman, G. J., Hillier, F., and Lieberman, G., *Introduction to Operations Research*, McGraw–Hill, New York, 2004, Chap. 12.

[42] Morari, M., "Hybrid System Analysis and Control via Mixed Integer Optimization," *IFAC Proceedings Volumes*, Vol. 34, No. 25, 2001, pp. 1–12. https://doi.org/10.1016/S1474-6670(17)33796-5

[43] Schouwenaars, T., "Safe Trajectory Planning of Autonomous Vehicles," Ph.D. Thesis, Massachusetts Inst. of Technology, 2006.

[44] Richards, A., and How, J., "Mixed-Integer Programming for Control," *Proceedings of the 2005 American Control Conference*, IEEE, New York, 2005, pp. 2676–2683. https://doi.org/10.1109/acc.2005.1470372

[45] Borrelli, F., Bemporad, A., and Morari, M., *Predictive Control for Linear and Hybrid Systems*, 1st ed., Cambridge Univ. Press, New York, 2017, Chap. 2. https://doi.org/10.1017/9781139061759

[46] Chatterjee, A., *Rigid Body Collisions: Some General Considerations, New Collision Laws, and Some Experimental Data*, Cornell Univ., Ithaca, NY, 1997, Chaps. 3–6.

[47] Cataldo, E., and Sampaio, R., "A Brief Review and a New Treatment for Rigid Bodies Collision Models," *Journal of the Brazilian Society of Mechanical Sciences*, Vol. 23, No. 1, 2001, pp. 63–78. https://doi.org/10.1590/S0100-73862001000100006

[48] Guelman, M., and Aleshin, M., "Optimal Bounded Low-Thrust Rendezvous with Fixed Terminal-Approach Direction," *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 2, 2001, pp. 378–385. https://doi.org/10.2514/2.4722

[49] Pardis, C. J., and Carter, T. E., "Optimal Power-Limited Rendezvous with Thrust Saturation," *Journal of Guidance, Control, and Dynamics*, Vol. 18, No. 5, 1995, pp. 1145–1150. https://doi.org/10.2514/3.21517

[50] Schneider, S. A., and Cannon, R. H., "Object Impedance Control for Cooperative Manipulation: Theory and Experimental Results," *IEEE Transactions on Robotics and Automation*, Vol. 8, No. 3, 1992, pp. 383–394. https://doi.org/10.1109/70.143355

[51] Mote, M., Egerstedt, M., Bylard, A., Feron, E., and Pavone, M., "Collision-Inclusive Trajectory Optimization for Spacecraft–Experimental Comparison," Jan. 2020. https://doi.org/10.6084/m9.figshare.11720640.v1

[52] Mote, M., Feron, E., Bylard, A., Egerstedt, M., and Pavone, M., "Online Safety Through Utilization of Collisions," Jan. 2020. https://doi.org/10.6084/m9.figshare.11720664.v1

[53] Gurriet, T., Mote, M., Ames, A. D., and Feron, E., "An Online Approach to Active Set Invariance," *2018 IEEE Conference on Decision and Control (CDC)*, IEEE, New York, 2018, pp. 3592–3599. https://doi.org/10.1109/cdc.2018.8619139

[54] Gurriet, T., Mote, M., Singletary, A., Ames, A. D., and Feron, E., "A Scalable Controlled Set Invariance Framework with Practical Safety Guarantees," *2019 IEEE Conference on Decision and Control (CDC)*, IEEE, New York, 2019.

[55] Starek, J. A., Schmerling, E., Maher, G. D., Barbee, B. W., and Pavone, M., "Fast, Safe, Propellant-Efficient Spacecraft Motion Planning Under Clohessy–Wiltshire–Hill Dynamics," *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 2, 2017, pp. 418–438. https://doi.org/10.2514/1.G001913