Collision-Inclusive Trajectory Optimization for Free-Flying Spacecraft

Mark Mote *, Magnus Egerstedt †, and Eric Feron ‡
Georgia Institute of Technology, Atlanta, GA, 30332, United States

Andrew Bylard §, and Marco Pavone ¶
Stanford University, Stanford, CA, 94305, United States

In systems where collisions can be tolerated, permitting and optimizing collisions in vehicle trajectories can enable a richer set of possible behaviors, allowing both better performance and determination of safest courses of action in scenarios where collision is inevitable. This paper develops an approach for optimal trajectory planning on a three degree-of-freedom free-flying spacecraft having tolerance to collisions. First, we use experimental data to formulate a physically realistic collision model for the spacecraft. We show that this model is linear over the expected operational range, enabling a piecewise affine representation of the full hybrid-vehicle dynamics. Next, we incorporate this dynamics model along with vehicle constraints into a mixed integer program. Experimental comparisons of trajectories with and without collision-avoidance requirements demonstrate the capability of the collision-tolerant 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

0_n	=	$n \times n$ zeros matrix
a_j, b_j	=	parameters defining the half-plane for the j^{th} wall
$\mathcal{A}$	=	net obstacle avoidance region
$\mathcal{F}_I$	=	inertial reference frame defined for the testbed
$\mathcal{F}_w^j$	=	inertial reference frame defined for the j^{th} wall
$\hat{i},\ \hat{j},\ \hat{k}$	=	basis vectors for inertial frame
I_n	=	$n \times n$ identity matrix
J	=	cost function

*Ph.D. Candidate, School of Aerospace Engineering, Georgia Institute of Technology
†Steve W. Chaddick School Chair and Professor in the School of Electrical and Computer Engineering, Georgia Institute of Technology
‡Dutton/Ducoffe Professor of Aerospace Software Engineering, School of Aerospace Engineering, Georgia Institute of Technology
§Ph.D. Candidate, Aeronautics and Astronautics Department, Stanford University.
¶Associate Professor, Aeronautics and Astronautics Department, Stanford University.

m	=	mass of spacecraft [kg]
M	=	large scalar defined according to the *Big-M* method
O	=	origin of inertial frame
$\mathcal{P}$	=	polygon defining an obstacle avoidance region
R	=	radius of spacecraft [m]
s_T, s_N	=	position tangent and normal to a wall [m]
$\mathcal{S}, \bar{\mathcal{S}}$	=	half-plane and polygon defining surfaces with which collision is permitted
$\hat{t}_j, \hat{a}_j, \hat{k}_j$	=	basis vectors for the local frame $\mathcal{F}_w^j$ defined tangent, normal, and upwards from the j^{th} wall
u	=	spacecraft control vector
v_T, v_N	=	components of velocity tangent and normal to a wall [m/s]
x_i	=	state of the vehicle on the i^{th} iteration, expressed in the inertial testbed frame $\mathcal{F}_I$
z	=	generic vector used to define sets
$\gamma_{i,j}$	=	event variable indicating the vehicles location with respect to the j^{th} wall of $\bar{\mathcal{S}}$
δ	=	effective distance of a collision surface from the origin of the inertial testbed frame $\mathcal{F}_I$ [m]
Δt	=	fixed time period between instants in discrete time horizon [s]
ζ_i	=	event variable used to indicate collisions with $\mathcal{S}$ on the i^{th} iteration
θ	=	orientation of the spacecraft with respect to the inertial frame [rad]
$\kappa_T, \kappa_N, \kappa_\omega$	=	tangential, normal, and angular restitution coefficients
$\lambda_{i,j}$	=	event variable indicating which wall of $\bar{\mathcal{S}}$ the vehicle is closest to on the i^{th} iteration
$\Xi_{i,j}$	=	event variable indicating collision with the j^{th} wall of $\bar{\mathcal{S}}$ on the i^{th} iteration
τ	=	number of steps in discrete time horizon
ω	=	angular velocity of spacecraft [rad/s]

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, either because they are 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] are 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 **(i)** providing a microgravity research platform, **(ii)** performing mobile camera tasks, and **(iii)** 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 common approach to mobility for microgravity robots is propulsive free-flying, where vehicles expend propellant to actuate their movement. However, propellant is often expensive to acquire or in limited supply. As a 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 [9], where the vehicle uses grasping contact in the surrounding environment to traverse between locations. A proposed faster and simpler alternative is the hopping modality [10, 11]. In this case, the vehicle uses a robotic arm to propel itself between some fixed handrails. While this strategy is attractive in the sense that it is completely propellantless, it is also much more restrictive than the propulsive free-flying strategy as 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). We show that this approach offers a strategy that is both less restrictive than the propellant-free approaches, and 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. Since 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 it is able to withstand low speed impact.

There is a rich body of work related to impulsive contact in robotics, spanning applications such as running [12], jumping [13], batting [14], air hockey [15], etc. In addition, the problem appears in the aerospace context, within landing [16], docking [17], grasping [18], and bouncing on planetary bodies [19]. Looking specifically at the case of vehicle collisions, there has been foundational work in analyzing the stability and robustness of a colliding vehicle [20], designing vehicles that are tolerant to collisions [21], and even extracting localization information from instances of impact [22]. 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 e.g. swimming, parkour, or in nature e.g. animals pushing off of [23] 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 [24], 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 utilize these initial results to develop a collision-tolerant, optimal trajectory-planning formulation for in-plane motion of a free-flying spacecraft. Note that since the allowable set of trajectories tolerant to 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 to the case where collisions are always avoided.

In addition to performance benefits, collisions may be utilized 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 safe 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 cannot be avoided. This problem can be addressed by either resorting to a backup controller when the MPC is not feasible [25] or softening the constraints (i.e. replacing constraints with penalties in the objective function) such that feasibility is preserved [26]. 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. Such capabilities 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 Section II, we review the mathematical preliminaries required to develop the main results. In Section III.A, we introduce the spacecraft, hardware, and environment used in the analysis and experimental case studies. Experimental collision data is obtained in this environment and used to derive a realistic collision model for the spacecraft in Section IV. Section V uses this information to specify an optimal strategy for moving between states. In Section VI an experimental case study is considered to compare the strategy to the collision-free case. It is shown that the proposed method is capable of significantly reducing a chosen objective function. Finally, Section VII 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 [27]. For our purposes, this is leveraged to optimize trajectories for a spacecraft experiencing unique dynamic modes encountered during collision and free flight. By modelling this hybrid behavior, integer variables to encode the choice of whether or not to collide.

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

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

where, $D_c \in \mathbb{R}^{m \times n_c}$, $D_b \in \mathbb{R}^{m \times n_b}$, $g \in \mathbb{R}^m$, $A_c \in \mathbb{R}^{p \times n_c}$, $A_b \in \mathbb{R}^{p \times n_b}$, $h \in \mathbb{R}^p$. Though 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 [28]. Formulations with linear, quadratic, or second order cone objectives are commonly applied in a variety of practical applications [29–31]. Although in theory these problems are difficult to solve [32], solutions can readily be found with good average case performance using off-the-shelf optimization software (e.g. CPLEX [33], Gurobi [34], MOSEK [35]).

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 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 $c^T z < d$ is used to *activate* distinct equality constraints,

$$\begin{cases} a_0^T z = b_0 & \text{if } c^T z < d \\ a_1^T z = b_1 & \text{if } c^T z \geq d, \end{cases} \tag{2}$$

with $z,\ a_i,\ c \in \mathbb{R}^n,\ b_i,\ d \in \mathbb{R}, i = 0, 1$. 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 $a, c \in \mathbb{R}^n$, $b, d \in \mathbb{R}$. We then have the following results [36–38].

Lemma 1 [27] *Given $M \in \mathbb{R}$ such that $\max_{z \in Z}(d - c^T z) < M$, the following are equivalent:*

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

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

Lemma 2 [27] *Given* $M \in \mathbb{R}$ *such that* $\max_{z \in Z}(c^T z - d) < M$*, the following are equivalent:*

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

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

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

$$\textbf{\textit{(i)}}\ [\zeta = 1] \implies [a^T z = b] \qquad \textbf{\textit{(ii)}}\ [a^T z - M(1-\zeta) \leq b] \ \wedge \ [a^T z + M(1-\zeta) \geq b]$$

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

From here we can combine these results to represent Eq. (2),

Theorem 1 [39] (Sec. 16.4) *Given* $M \in \mathbb{R}$ *sufficiently large such that* $\max(\max_{z \in Z}(|c^T z - d|), \max_{z \in Z}(a^T z - b)) < M$ *holds, then the system Eq.* (2) *is equivalent to:*

$$\begin{aligned} &[a_0^T z - M\zeta \leq b_0] \quad \wedge \quad [a_1^T z - M(1-\zeta) \leq b_1] \quad \wedge \quad [c^T z - M\zeta < d] \\ &[a_0^T z + M\zeta \geq b_0] \quad \wedge \quad [a_1^T z + M(1-\zeta) \geq b_1] \quad \wedge \quad [c^T z + M(1-\zeta) \geq d] \end{aligned}$$

In practice, the parameter M should be chosen carefully. While 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 non-strict inequality and adding a small number ε to the side with lesser value.

III. Description of the Free-Flyer Spacecraft and Testbed

A. Hardware and Environment

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 [40], 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

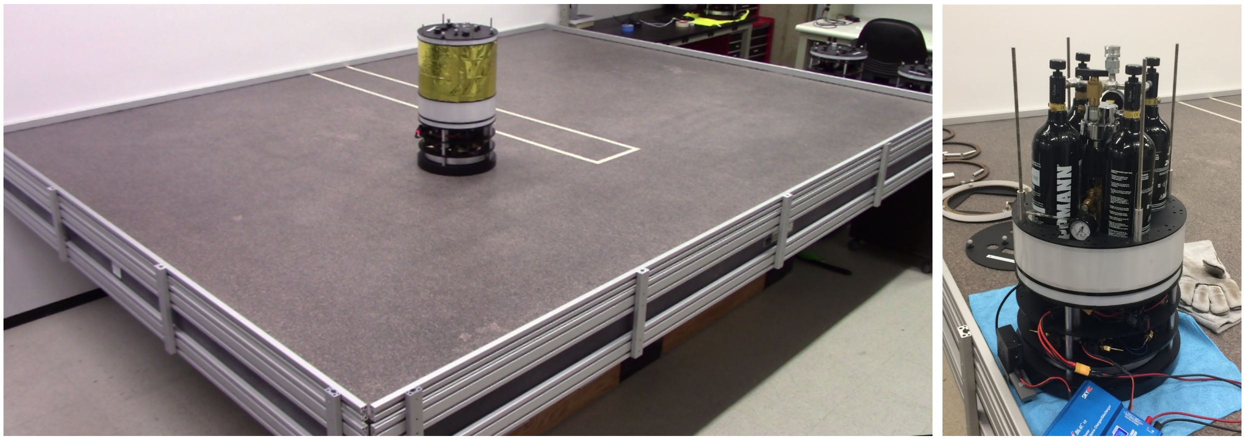

Fig. 1 Free-Flyer spacecraft and testbed. The avoidance region for the experimental scenario in Section VI is outlined in adhesive tape.

Table 1 Free-Flyer Spacecraft Parameters

Parameter	Value	Unit
average mass (m)	18.08	kg
radius (R)	0.157	m
max individual thruster output ($u_{\mathrm{T,max}}$)	0.20	N
body inertia about spin axis (I_{b})	0.184	kgm^2
reaction wheel inertia (I_{w})	0.029	kgm^2
max acceleration of reaction wheel	0.628	m/s^2
reaction wheel speed range	[60,340]	RPM

with compressed air. The robots are also equipped with actuators commonly used in spacecraft, namely a reaction wheel for attitude control and 8 cold-gas thrusters primarily for translational control. Due to 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 hours continuously. Further parameters for the free-flyer robot can be found in Table 1, where average mass is reported due to a range based on the state of the tanks. 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 offboard 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. Finally, the granite table used for experiments is 9' × 12', allowing ample room for complex planning scenarios.

B. Vehicle Constraints and Nominal Dynamics

The motion of the spacecraft described in Section III.A is expressed in an inertial frame $\mathcal{F}_I = (O, \hat{\imath}, \hat{\jmath}, \hat{k})$ with right-handed orthogonal basis vectors: $\hat{\imath}$, $\hat{\jmath}$ tangent to the surface of the testbed, and $\hat{k} = \hat{\imath} \times \hat{\jmath}$ pointing upwards from the surface. The position of the vehicle's center of mass O_B with respect to the origin O is $s = s_x\hat{\imath} + s_y\hat{\jmath}$ and the translational velocity $v = v_x\hat{\imath} + v_y\hat{\jmath}$. We may also define a body frame $\mathcal{F}_B = (O_B, \hat{\imath}_B, \hat{\jmath}_B, \hat{k}_B)$, 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 and angular velocity of the vehicle is $\omega = \dot{\theta}\hat{k}$. The *nominal* —i.e. 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, u_y are the translational accelerations due to applied thrust, 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. 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.

$$u_x^2 + u_y^2 \leq u_{\max}^2, \qquad u_{\max} = \min_\theta(u_{\max}(\theta))), \tag{4}$$

With this geometry, we have that $u_{\max} = \frac{2}{m} u_{T,max}$, where $u_{\mathrm{T,max}}$ is the maximum force output of a single thruster, and m the body mass.

IV. Collision Model for Free-Flyer Spacecraft

In order 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 that have been proposed to model *general* collision behavior over a wide range of scenarios [41, 42]. However, since 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 2 shows the effects of 82 individual collisions for the spacecraft and environment described in Section III.A. 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 pre-collision normal velocity (v_N^-) and relative velocity of the point of contact (v_{rel}^-). Furthermore, we observe that effects in the normal direction are uncoupled from the tangential and rotational effects, leading to the following model,

$$\begin{bmatrix} \Delta v_T \\ \Delta v_N \\ \Delta \omega \end{bmatrix} = \begin{bmatrix} 0 & \kappa_T \\ \kappa_N & 0 \\ 0 & \kappa_\omega \end{bmatrix} \begin{bmatrix} v_N^- \\ v_{\text{rel}}^- \end{bmatrix} \tag{5}$$

where,

$$v_{\text{rel}}^- \triangleq v_T^- + R\omega^- \tag{6}$$

and $(\kappa_T,\ \kappa_N,\ \kappa_\omega) = (-0.29, -1.43, -5.0)$. These 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 pre-collision velocities until the point of contact, and post-impact velocities afterward. Let $\Delta t = \Delta t^- + \Delta t^+$ 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 Figure 3a). Then the experimental model of Eq. (5) yields the following position update equations,

$$\begin{aligned} \Delta s_T &= (1 + \kappa_T)\Delta t v_T^- + \kappa_T R \Delta t \omega^- - \kappa_T (v_T^- + R\omega^-)\Delta t^- \\ \Delta s_N &= (1 + \kappa_N)\Delta t v_N^- + \kappa_N (s_N^- - \delta) \\ \Delta \theta &= (1 + \kappa_\omega R)\Delta t \omega^- + \kappa_\omega \Delta t v_T^- - \kappa_\omega (v_T^- + R\omega^-)\Delta t^- \end{aligned} \tag{7}$$

where the time until collision is,

$$\Delta t^- = \frac{(\delta - s_N^-)}{v_N^-} \tag{8}$$

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,

$$e_T \le \frac{\kappa_T}{2} |v_{\text{rel}}^-| \Delta t, \qquad e_N = 0, \qquad e_\theta \le \frac{\kappa_\omega}{2} |v_{\text{rel}}^-| \Delta t, \tag{9}$$

where e_T, e_N, e_θ are the errors in the tangential, normal, and angular directions respectively. Note that errors vanish both as $|v_{\text{rel}}^-|$ decreases, and for finer resolutions Δt. The basic collision geometry is illustrated in Figure 3a.

V. Problem Formulation

This section formulates the problem of generating optimal trajectories for a spacecraft in the presence of **(i)** an obstacle avoidance region $\mathcal{A}$, composed of N_{P} convex polygons $\mathcal{P}_k$, with $k \in \{1, \ldots, N_{\text{P}}\}$, and **(ii)** surfaces $\mathcal{S}$, $\bar{\mathcal{S}}$ with which collisions are permissible. For both $\mathcal{P}$ and $\mathcal{S}, \bar{\mathcal{S}}$, we will use the convention that the *interior* of the walls is denoted by the

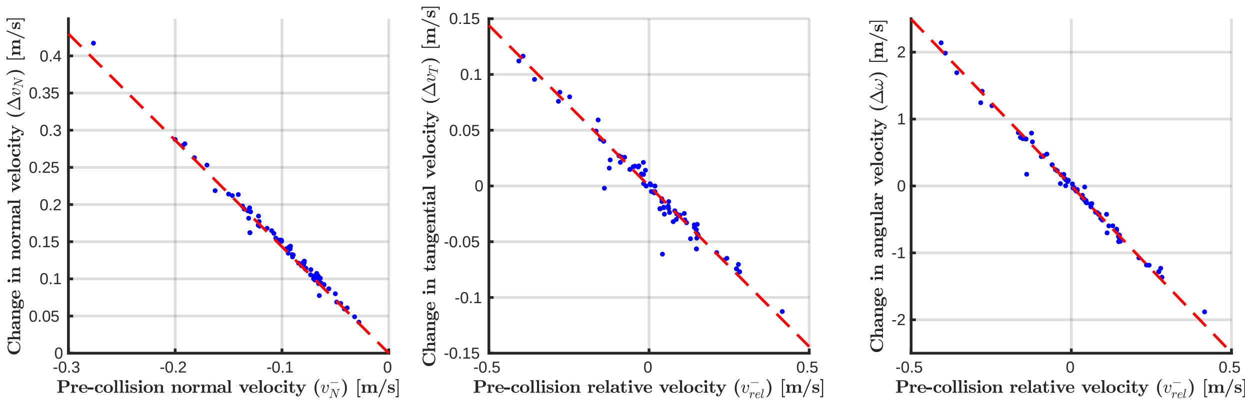

Fig. 2 Observed data from 82 collisions, with linear interpolations taken with respect to the least squares error.

union of *sub*-level 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 a 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 i^{th} time-step is defined as $x_i^T = [s_{x,i}, s_{y,i}, \theta_i, v_{x,i}, v_{y,i}, \omega_i]$, and control vector as $u_i^T = [u_{x,i}, u_{y,i}, u_{\theta,i}]$. For completeness, we expound upon some basic control and obstacle avoidance constraint formulations found in [38].

A. Obstacle Avoidance and Saturation Constraints

The saturation constraint Eq. (4) can be represented by approximating the Euclidean norm with an N_U sided polygon,

$$u_{x,i} \sin\left(\frac{2\pi n}{N_\mathrm{U}}\right) + u_{y,i} \cos\left(\frac{2\pi n}{N_\mathrm{U}}\right) \leq u_{max}, \quad n = 1, \ldots, N_\mathrm{U}, \quad i = 1, \ldots, \tau. \tag{10}$$

While the approximation improves with the number of sizes in the polygon, 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 $\mathcal{P}_k$,

$$\mathcal{A} = \{z \in \mathbb{R}^2 \mid \bigvee_{k=1}^{N_\mathrm{P}} z \in \mathcal{P}_k\}, \quad \text{where,} \quad \mathcal{P}_k = \{z \in \mathbb{R}^2 \mid c_{k,q}^T z < d_{k,q},\ q = 1, \ldots, N_{\mathrm{Q},k}\} \tag{11}$$

where $c_{k,q} \in \mathbb{R}^2$, $d_{k,q} \in \mathbb{R}$ specify the q^{th} side of the k^{th} polygon, which has $N_{\mathrm{Q},k}$ sides. We can construct the avoidance constraint $s \notin \mathcal{A}$ by using defining event variables $\psi_{k,q,i} \in \{0, 1\}$ such that $c_{k,q}^T s_i < d_{k,q} \implies \psi_{k,q,i} = 1$, 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 $\mathcal{P}_k$. This is accomplished with the following constraints,

$$\bigwedge_{q=1}^{N_{\mathrm{Q},k}} c_{k,q}^T s_i + M\psi_{k,q,i} \geq d_{k,q} \quad \wedge \quad \sum_{q=1}^{N_{\mathrm{Q},k}} \psi_{k,q,i} \leq N_{\mathrm{Q},k} - 1, \qquad k = 1, \dots, N_{\mathrm{P}}, \quad i = 1, \dots, \tau. \tag{12}$$

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 $c_{k,q}^T s_i \geq d_{k,q}$.

Example: Rectangular Boundary Let's 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 $s \notin \mathcal{A}$ are,

$$-s_x + M\psi_1 \geq -z_1^{min} \ \wedge \ s_x + M\psi_2 \geq z_1^{max} \ \wedge \ -s_y + M\psi_1 \geq -z_2^{min} \ \wedge \ s_y + M\psi_2 \geq z_2^{max} \ \wedge \ \sum_{q=1}^{4} \psi_q \leq 3. \tag{13}$$

$\Diamond$

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

For notational simplicity, will assume for this case that the basis vectors of $\mathcal{F}_I$ are oriented with the wall $\mathcal{S}$ such that $\hat{j}$ points away from the wall, $\hat{i}$ is tangent, and $\hat{k} = \hat{i} \times \hat{j}$ remains pointed upwards (see Figure 3a). The discrete time equations of motion are given by,

$$x_{i+1} - x_i = \begin{cases} Ax_i + Bu_i & \text{if} \quad \zeta_{i+1} = 0 \\ A_c x_i + b_c & \text{if} \quad \zeta_{i+1} = 1\,, \end{cases} \qquad i = 1, \dots, \tau - 1 \tag{14}$$

where A, B represent the nominal dynamics,

$$A = \begin{bmatrix} 0_3 & I_3\Delta t \\ 0_3 & 0_3 \end{bmatrix}, \qquad B = \begin{bmatrix} 0.5 I_3 \Delta t^2 \\ I_3 \Delta t \end{bmatrix} \tag{15}$$

and A_c, b_c represent the collision dynamics,

$$A_c = \begin{bmatrix} 0 & 0 & 0 & (1 + 0.5\kappa_T)\Delta t & 0 & 0.5\kappa_T R \Delta t \\ 0 & \kappa_N & 0 & 0 & (1 + \kappa_N)\Delta t & 0 \\ 0 & 0 & 0 & 0.5\kappa_\omega \Delta t & 0 & (1 + 0.5\kappa_\omega R)\Delta t \\ 0 & 0 & 0 & \kappa_T & 0 & \kappa_T R \\ 0 & 0 & 0 & 0 & \kappa_N & 0 \\ 0 & 0 & 0 & \kappa_\omega & 0 & \kappa_\omega R \end{bmatrix}, \qquad b_c = \begin{bmatrix} 0 \\ -\kappa_N \delta \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{16}$$

The period between steps i and $i - 1$ is Δt.

Let the wall be located at a distance δ' from the origin of the inertial frame $\mathcal{F}_I$, and define $\delta \triangleq \delta' + R$. Then we can define the wall by the set $\mathcal{S} = \{z \in \mathbb{R}^2 \mid z_2 < \delta\}$. 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, i.e.

$$s_{y,i} + v_{y,i}\Delta t + u_{y,i} 0.5\Delta t^2 < \delta \iff \zeta_{i+1} = 1, \qquad i = 1, \dots, \tau - 1 \tag{17}$$

Using Lemmas 1,2, we can express Eq. (17) with the equivalent set of constraints,

$$\begin{cases} s_{y,i} + v_{y,i}\Delta t + u_{y,i} 0.5\Delta t^2 + M\zeta_{i+1} \geq \delta \\ s_{y,i} + v_{y,i}\Delta t + u_{y,i} 0.5\Delta t^2 - M(1 - \zeta_{i+1}) < \delta, \end{cases} \qquad i = 1, \dots, \tau - 1 \tag{18}$$

and from Theorem 1, we see that Eq. (14) is equivalent to,

$$\begin{cases} x_{i+1} - x_i - Ax_i - Bu_i + MI_6\zeta_{i+1} \geq 0 \\ x_{i+1} - x_i - Ax_i - Bu_i - MI_6\zeta_{i+1} \leq 0 \\ x_{i+1} - x_i - A_c x_i - b_c + MI_6(1 - \zeta_{i+1}) \geq 0 \\ x_{i+1} - x_i - A_c x_i - b_c - MI_6(1 - \zeta_{i+1}) \leq 0, \end{cases} \qquad i = 1, \dots, \tau - 1. \tag{19}$$

Section VIII.B 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. While 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: $\zeta_i = 1 \implies v_{\mathrm{rel},i} = 0$. Equivalently, from Lemma 3,

$$v_{x,i} + R\omega_i - M(1 - \zeta_i) \leq 0 \qquad \wedge \qquad v_{x,i} + R\omega_i + M(1 - \zeta_i) \geq 0, \qquad i = 2, \dots, \tau. \tag{20}$$

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,

$$\bar{\mathcal{S}} = \{z \in \mathbb{R}^2 \mid a_j^T z < b_j,\ j = 1, \dots, N_{\bar{\mathcal{S}}}\} \tag{21}$$

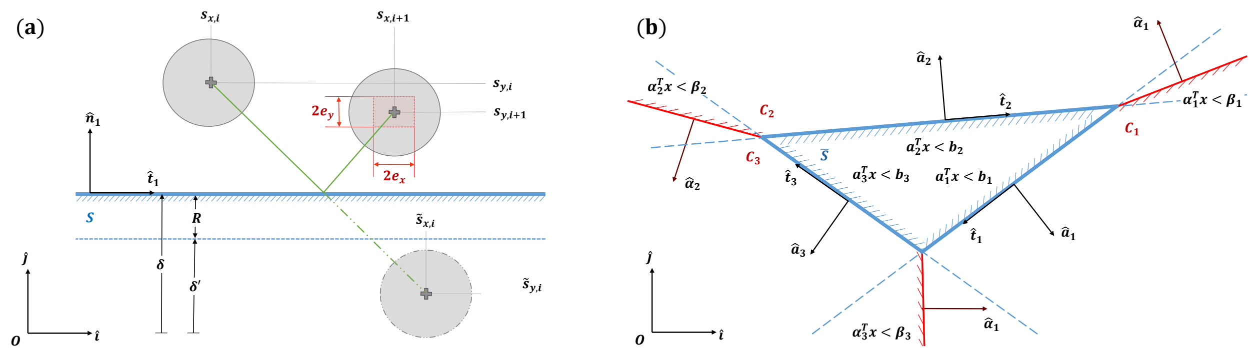

Fig. 3 (a) Collision geometry for flat wall $\mathcal{S}$; (b) Example triangular collision polygon $\bar{\mathcal{S}}$

where $N_{\bar{\mathcal{S}}}$ is the number of sides in the polygon and the indices j label the walls in a counterclockwise order (see Fig. 3). 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 j^{th} wall of $\bar{\mathcal{S}}$ on iteration i if: **(i)** the position at timestep $i-1$ is closest to the boundary of the j^{th} wall, and **(ii)** 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 $\tilde{s}_i \triangleq A_s x_{i-1} + B_s u_{i-1} \in \bar{\mathcal{S}}$, where A_s, B_s are the first two rows of A, B, corresponding to the position update. Likewise we can represent the first condition as $s_{i-1} \in C_j$ where C_j is the region exterior to the polygon, closest to wall j. This can be defined as,

$$C_j = \{z \in \mathbb{R}^2 \,|\, \alpha_j^T z < \beta_j\,,\ \alpha_{\sigma(j)}^T z \geq \beta_{\sigma(j)}\,,\ a_j^T z \geq b_j\,\} \tag{22}$$

where $\sigma(j)$ is the j^{th} element of $\sigma \triangleq (N_{\bar{\mathcal{S}}}, 1, 2, \ldots, N_{\bar{\mathcal{S}}} - 1)$, and α_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 j^{th} edge. An example configuration is shown in Figure 3b.

Our goal is to define event variables $\Xi_{i,j} \in \{0, 1\}$ such that,

$$\Xi_{i,j} = 1 \iff \tilde{s}_i \in \bar{\mathcal{S}} \wedge s_{i-1} \in C_j, \qquad i = 2, \ldots, \tau, \quad j = 1, \ldots, N_{\bar{\mathcal{S}}}. \tag{23}$$

These variables are used to activate the collision dynamics for the wall involved in the collision. We begin by introducing constraints,

$$a_j^T \tilde{s}_i + M\gamma_{i,j} \geq b_j \quad \wedge \quad a_j^T \tilde{s}_i - M(1 - \gamma_{i,j}) < b_j, \qquad i = 2, \ldots, \tau, \quad j = 1, \ldots, N_{\bar{\mathcal{S}}} \tag{24}$$

which fixes $\gamma_{i,j} = 1$ when the constraint $a_j^T \tilde{s}_i < b_j$ is satisfied. 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),

$$\alpha_j^T s_i + M\lambda_{i,j} \geq \beta_j \quad \wedge \quad \alpha_j^T s_i - M(1 - \lambda_{i,j}) < \beta_j \qquad i = 2 \ldots \tau, \quad j = 1, \ldots, N_{\bar{\mathcal{S}}} \tag{25}$$

which expresses $\lambda_{i,j} = 1 \iff \alpha_j^T \tilde{s}_i < \beta_j$ for the appropriate values of i, j. The equivalencies in Eq. (23) can then be enforced by the constraints,

$$\sum_{p=1}^{N_{\bar{S}}} \gamma_{i,p} + \lambda_{i-1,j} - \lambda_{i-1,\phi(j)} - \gamma_{i-1,j} + M(1 - \Xi_{i,j}) \geq N_{\bar{S}} + 1 \;\wedge\; \sum_{p=1}^{N_{\bar{S}}} \gamma_{i,p} + \lambda_{i-1,j} - \lambda_{i-1,\phi(j)} - \gamma_{i-1,j} - M\Xi_{i,j} \leq N_{\bar{S}} \quad (26)$$

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

The dynamics for this system are then,

$$x_{i+1} - x_i = \begin{cases} Ax_i + Bu_i & \text{if} \quad \sum_{j=1}^{N_{\bar{S}}} \Xi_{i+1,j} \leq 0 \\ A_c^j x_i + b_c^j & \text{if} \quad \Xi_{i+1,j} = 1, \quad j = 1, \ldots, N_{\bar{S}}, \end{cases} \qquad i = 1, \ldots, \tau - 1 \quad (27)$$

where A_c^j, b_c^j are the collision dynamics for the j^{th} wall. To represent these dynamics, let us first define a local frame for the j^{th} wall $\mathcal{F}_w^j = (O, \hat{t}_j, \hat{a}_j, \hat{k})$ with $\hat{t}_j$, $\hat{a}_j$ pointing tangent and normal to wall j, and $\hat{k}_j = \hat{t}_j \times \hat{a}_j$ upwards. Each local frame is a rotation of $\mathcal{F}_I = (O, \hat{i}, \hat{j}, \hat{k})$ about $\hat{k}$ by an angle ϕ_j. Let $L_3(\phi_j) \in \mathbb{R}^{3\times3}$ 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,

$$A_c^j = \Lambda_j A_c \Lambda_j^T \qquad b_c^j = \Lambda_j [0,\ -\kappa_N \frac{b_j}{\|a_j\|_2},\ 0,\ 0,\ 0,\ 0]^T \qquad \text{where,} \quad \Lambda_j = \begin{bmatrix} L_3(\phi_j) & 0_3 \\ 0_3 & L_3(\phi_j) \end{bmatrix} \quad (28)$$

The dynamics in Eq. (27) are then represented by the following MIP constraints,

$$\begin{aligned} x_{i+1} - x_i - Ax_i - Bu_i - M\Big(\sum_{j=1}^{N_{\bar{S}}} \Xi_{i+1,j}\Big) \leq 0 \quad \wedge \quad x_{i+1} - x_i - Ax_i - Bu_i + M\Big(\sum_{j=1}^{N_{\bar{S}}} \Xi_{i+1,j}\Big) \geq 0 \\ x_{i+1} - x_i - A_c^j x_i - b_c^j - M\Xi_{i+1,j} \leq 0 \quad \wedge \quad x_{i+1} - x_i - A_c^j x_i - b_c^j + M\Xi_{i+1,j} \geq 0, \quad j = 1, \ldots, N_{\bar{S}} \end{aligned} \quad (29)$$

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. While the proposed methodology does not assume a particular form for the objective, vehicle efficiency is commonly of critical importance to real-world missions. We review here two common approximations for penalizing actuation using quadratic and linear functions. A simple option is to use the power limiting cost function [43, 44],

$$J_1 = \sum_{i=1}^{\tau} (u_{x,i}^2 + u_{y,i}^2) \quad (30)$$

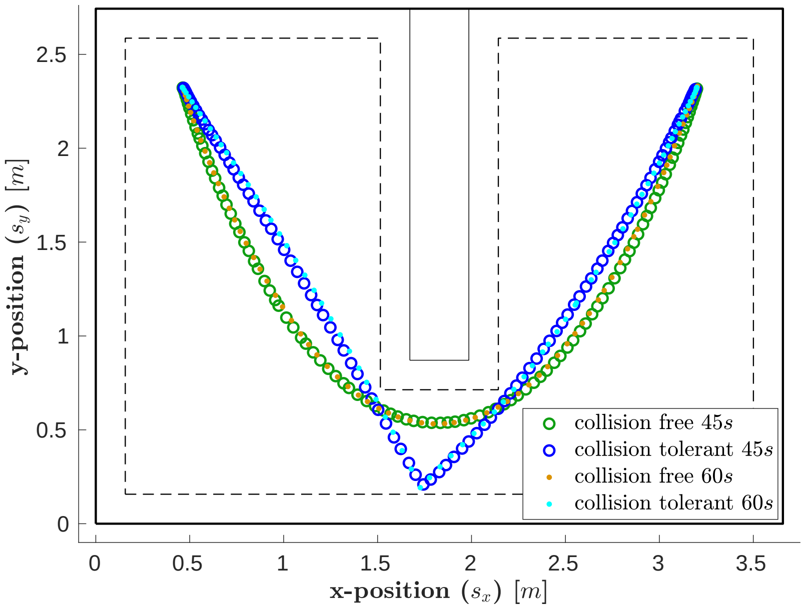

Fig. 4 Paths taken by collision free (green) and collision tolerant (blue) planners in 60 second experimental scenario. Video at https://youtu.be/4kOOn6TPuDI.

which forms a Mixed Integer Quadratic Program (MIQP). With the introduction of additional constraints, is also possible to use a PWA approximation of the Euclidean norm of commanded translational acceleration [29]. The resulting cost function is linear,

$$J_2 = \sum_{i=1}^{\tau} G_i, \quad s.t. \quad \bigwedge_{n=1}^{N_J} \quad G_i \geq u_{\mathrm{x},\,i} \sin\left(\frac{2\pi n}{N_J}\right) + u_{\mathrm{y},\,i} \cos\left(\frac{2\pi n}{N_J}\right), \quad i = 1, \ldots, \tau. \tag{31}$$

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

VI. Experimental Performance Comparison

Consider the spacecraft and testbed described in Sections III-V 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 Figure 4. We now compare the performance of vehicles navigating from rest at initial position $s_1 = [0.46, 2.32]^T$ to rest at final position $s_\tau = [3.20, 2.32]^T$, while remaining in the boundary of the testbed, and avoiding a central rectangular region $\mathcal{P} = \{z \in \mathbb{R}^2 \,|\, z_1 \in [1.50, 2.15],\ z_2 \in [0.70, 2.75]\}$. We minimize the cost J_1 introduced in Eq. (30). 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. (20)). 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.

Table 2 Experimental Cost Values

Specification	PWM Cost [s]	Percent Reduction	J_1 Cost [m^2/s^4]	Percent Reduction
Collision Free, 45s	6.50e+1	N/A	1.82e-2	N/A
Collision Tolerant, 45s	4.44e+1	31.7	1.01e-2	44.3
Collision Free, 60s	4.38e+1	N/A	3.17e-3	N/A
Collision Tolerant, 60s	3.33e+1	23.8	2.44e-3	22.8

The trajectory is generated with Gurobi optimization software using the formulation in Section V with the parameters listed above. 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,

$$u(t) = u^*(t) + K_{\mathrm{lqr}}(x^*(t) - x(t)) \tag{32}$$

where $K_{\mathrm{lqr}} \in \mathbb{R}^{3\times3}$ is the LQR gain matrix, $u^* \in \mathbb{R}^3$, $x^* \in \mathbb{R}^6$ 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 u is then mapped to PWM signals on the thrusters $u_{\mathrm{pwm}} \in [0, 1]^8$. 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 seconds. The respective trajectories taken are shown in Figure 4, and the efficiency measures are plotted against time in Figure 5. 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. We see that the collision tolerant approach is capable of demonstrating significant improvements in overall efficiency for a given time horizon. As a result of this, we see that for a given allocation of fuel, this approach allows the vehicle to traverse to its final location in less time. 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 tolerant spacecraft is able 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 trade-off when using this approach in that a spike in thrust is seen to occur directly after collision. This might be attributed to a number of factors which could potentially lead to increased model error on the collision iteration. For example: sensitivity to modelling 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.

VII. Safety Through Collision Tolerance

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

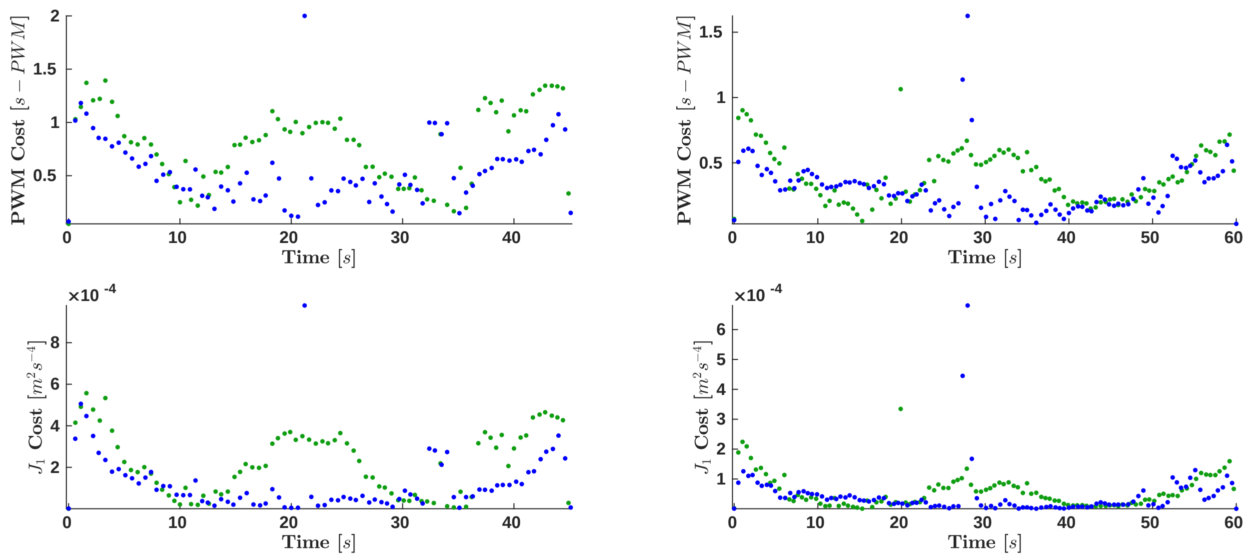

Fig. 5 PWM and J_1 costs vs time for collision free (green) and collision tolerant (blue) approaches in 45 s and 60 s experiments

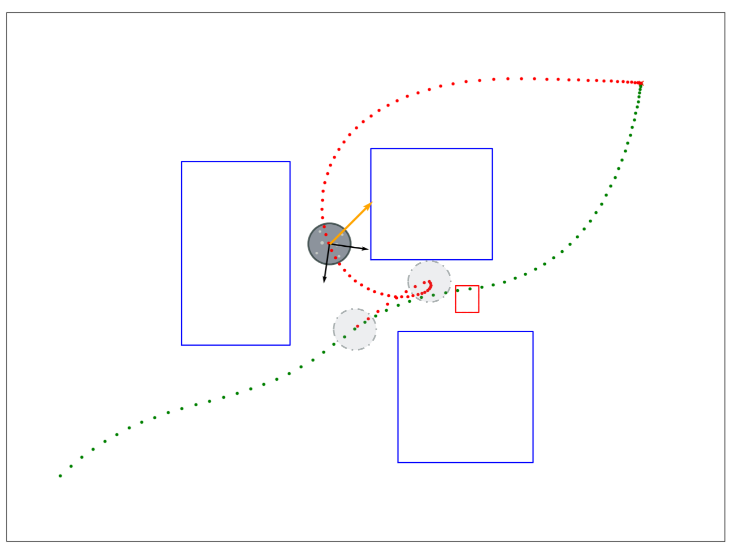

Fig. 6 Original (green) and updated path (red) taken to evade observed obstacle. An outline of the vehicle is shown at the point where the nominal MPC is rendered infeasible, and the point of the planned collision. Video: https://youtu.be/B8VU9IS12WU.

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 for example, Figure 4). 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 direct the vehicle toward an *optimal mitigating action*. In this section we turn to a simulated scenario to demonstrate the potential of the collision tolerant planner to bring about enhanced safety in this sense.

Consider the case where a robot must traverse across a cluttered environment, consisting of a number of walls whose locations are known to the spacecraft 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, while 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 Figure 6. 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 Figure 6, however it discovers that its path is blocked by an obstacle (red box), and is not able to stop in time to prevent collision. For the collision free case, 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 [38].

As an alternate 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 *impact speeds*, defined for the j^{th} wall on the i^{th} iteration as follows,

$$v_{i,j}^{\text{impact}} \triangleq -\Xi_{i,j}[v_{x,i},\ v_{y,i}]\frac{a_j}{||a_j||} \tag{33}$$

where $\Xi_{i,j} = 1$ indicates collision with wall defined by 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 follows,

$$J = J_{\text{nom}} + \sum_{i=2}^{\tau}\sum_{j\in\sigma_1} K_1 v_{i,j}^{\text{impact}} + \sum_{i=2}^{\tau}\sum_{j\in\sigma_2} K_2 v_{i,j}^{\text{impact}} \tag{34}$$

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 the Figure 6 shows the new trajectory that is calculated using this cost function once the red box is discovered. 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 utilizes stored angular momentum to push itself away on impact. Additional simulation parameters are listed in Section VIII.A.

VIII. Conclusion

A framework is proposed for optimizing spacecraft trajectories comprising planned collisions using a mixed integer programming formulation. An empirical collision model is developed and the algorithm is implemented on a three degree-of-freedom frictionless testbed designed to replicate spacecraft motion. Experiments comparing the efficiency of collision free and collision tolerant approaches demonstrate the capability of this approach 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 modelling 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.

Appendix

A. Parameters used in Simulated Safety Scenario

The vehicle navigates from the initial state $x_1^T = [0.30,\ 0.40,\ 0.00,\ 0.15,\ 0.15,\ 0.00\,]$, to the final state $x_\tau^T = [0.30,\ 0.40,\ 0.00,\ 0.15,\ 0.15,\ 0.00\,]$. The time resolution is $\Delta t = 0.06$ s. 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 location of the boxes are listed in Table 3. 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 $t = 21.37$ s from the start of the simulation.

Table 3 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

B. Example: 1-D Modelling 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 $g = 9.8$ m/s 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 $\Delta v_y^+,\ v_y^-$ indicate the change in velocity, and velocity immediately before collision respectively. We can model the motion of the particle with,

$$x_{i+1} - x_i = \begin{cases} A_c x_i & \text{if} \quad s_{y,i} + v_{y,i}\Delta t - 0.5g\Delta t^2 < 0 \\ A x_i + b & \text{if} \quad s_{y,i} + v_{y,i}\Delta t - 0.5g\Delta t^2 \geq 0, \end{cases}$$

where,

$$A_c = \begin{bmatrix} \kappa_N & (1+\kappa_N)\Delta t \\ 0 & \kappa_N \end{bmatrix}, \quad A = \begin{bmatrix} 0 & \Delta t \\ 0 & 0 \end{bmatrix}, \quad b = \begin{bmatrix} -0.5g\Delta t^2 \\ -g\Delta t \end{bmatrix}, \quad x_i = \begin{bmatrix} s_{y,i} \\ v_{y,i} \end{bmatrix},$$

and Δt is the timestep. Note that since 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 the condition: $\zeta_i = 1 \iff s_{N,i} + v_{N,i}\Delta t - 0.5g\Delta t^2 < 0$. This is transformed to the constraints,

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

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

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

which are applied at $i = 1, \ldots, \tau - 1$. The system is simulated with a MIP using $\Delta t = 0.01$, and $\tau = 150$. The solution to

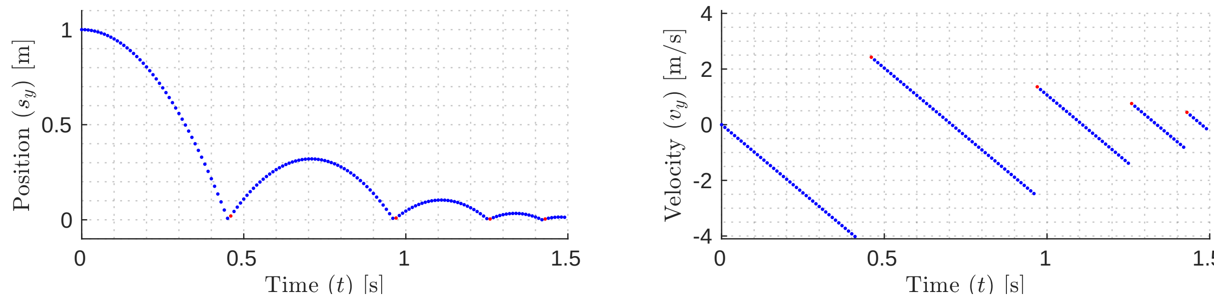

Fig. 7 Position and velocity of bouncing particle with normal restitution coefficient $\kappa_N = -1.55$ released 1 m from the ground. Points found with the collision update equation are represented in red, and nominal update equation in blue. The system is modelled with a MIP with a time resolution of 0.01 s, and horizon of $\tau = 150$.

this is plotted in Figure 7.

C. Truth Tables for Lemmas 1 and 2

Table 4 Lemmas 1 and 2

$c^T z < d$	$\zeta = 1$	$c^T z + M\zeta \geq d$	$c^T z - M(1-\zeta) < d$	$c^T z < d \implies \zeta = 1$	$\zeta = 1 \implies c^T z < d$
T	T	T	T	T	T
T	F	F	T	F	T
F	T	T	F	T	F
F	F	T	T	T	T

As shown in Table 4, Lemmas 1, 2 may be conveniently expressed in the form of a truth table. The constant M is assumed to have a value satisfying the appropriate assumptions from these lemmas over the variable domains.

References

[1] Pedersen, L., Kortenkamp, D., Wettergreen, D., Nourbakhsh, I., and Korsmeyer, D., "A survey of space robotics," 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 Society for Optics and Photonics, 2003, pp. 97–112.

[3] Williams, T., and Tanygin, S., "On-orbit engineering tests of the AERCam Sprint robotic camera vehicle," *Spaceflight mechanics 1998*, 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," *IDA Paper P-8335, Institute for Defense Analysis, 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, 2017 IEEE*, IEEE, 2017, pp. 1–9.

[6] Acquatella, P., "Development of automation & robotics in space exploration," *Proceedings of the AIAA SPACE 2009 Conference & Exposition. California*, Citeseer, 2009, pp. 1–7.

[7] Smith, T., Barlow, J., Bualat, M., Fong, T., Provencher, C., Sanchez, H., and Smith, E., "Astrobee: A new platform for free-flying robotics on the international space station," 2016.

[8] Miller, D., Saenz-Otero, A., Wertz, J., Chen, A., Berkowski, G., Brodel, C., Carlson, S., Carpenter, D., Chen, S., 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*, Clearwater, Florida, January, 2000, pp. 167–179.

[9] 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.

[10] Virgili-Llop, J., and Romano, M., "ASTROBATICS: Demonstrating Propellantless Robotic Maneuvering Onboard the International Space Station," 2018.

[11] Alsup, K. P., "Robotic Spacecraft Hopping: Application And Analysis," Tech. rep., Naval Postgraduate School Monterey United States, 2018.

[12] Xi, W., and Remy, C. D., "Optimal gaits and motions for legged robots," *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, IEEE, 2014, pp. 3259–3265.

[13] 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.

[14] 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." *ICINCO (2)*, 2016, pp. 90–101.

[15] Spong, M. W., "Impact controllability of an air hockey puck," *Systems & Control Letters*, Vol. 42, No. 5, 2001, pp. 333–345.

[16] 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, p. 031003.

[17] Lee, S. H., Yi, B.-J., Kim, S. H., and Kwak, Y. K., "Analysis on impact propagation of docking platform for spacecraft," *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, Vol. 1, IEEE, 2001, pp. 413–420.

[18] 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," *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, IEEE, 2016, pp. 4907–4913.

[19] 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.

[20] 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.

[21] 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.

[22] 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*, 2018.

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

[24] Mote, M., Afman, J. P., and Feron, E., "Robotic trajectory planning through collisional interaction," *Decision and Control (CDC), 2017 IEEE 56th Annual Conference on*, IEEE, 2017, pp. 1144–1149.

[25] 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.

[26] Kerrigan, E. C., and Maciejowski, J. M., "Soft constraints and exact penalty functions in model predictive control," 2000.

[27] Bemporad, A., and Morari, M., "Control of systems integrating logic, dynamics, and constraints," *Automatica*, Vol. 35, No. 3, 1999, pp. 407–427.

[28] Lubin, M., Zadik, I., and Vielma, J. P., "Mixed-integer convex representability," *International Conference on Integer Programming and Combinatorial Optimization*, Springer, 2017, pp. 392–404.

[29] 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.

[30] Mellinger, D., Kushleyev, A., and Kumar, V., "Mixed-integer quadratic program trajectory generation for heterogeneous quadrotor teams," *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, IEEE, 2012, pp. 477–483.

[31] 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, 2018, pp. 4409–4416.

[32] Del Pia, A., Dey, S. S., and Molinaro, M., "Mixed-integer quadratic programming is in NP," *Mathematical Programming*, Vol. 162, No. 1-2, 2017, pp. 225–240.

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

[34] OPTIMIZATION, G., "INC. Gurobi optimizer reference manual, 2015," *URL: http://www. gurobi. com*, 2014.

[35] Mosek, A., "The MOSEK optimization software," *Online at http://www. mosek. com*, Vol. 54, No. 2-1, 2010, p. 5.

[36] Hillier, F. S., Lieberman, G. J., Hillier, F., and Lieberman, G., "MP Introduction to Operations Research," , 2004.

[37] Morari, M., "Hybrid system analysis and control via mixed integer optimization," *IFAC Proceedings Volumes*, Vol. 34, No. 25, 2001, pp. 1–12.

[38] Schouwenaars, T., "Safe trajectory planning of autonomous vehicles," Ph.D. thesis, Massachusetts Institute of Technology, 2006.

[39] Borrelli, F., Bemporad, A., and Morari, M., *Predictive Control for Linear and Hybrid Systems*, 1st ed., Cambridge University Press, New York, NY, USA, 2017.

[40] 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.

[41] Chatterjee, A., *Rigid body collisions: some general considerations, new collision laws, and some experimental data*, Cornell University Ithaca, NY, 1997.

[42] 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.

[43] 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.

[44] 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.