Fast, Safe, Propellant-Efficient Spacecraft Motion Planning Under Clohessy–Wiltshire–Hill Dynamics

Joseph A. Starek,* Edward Schmerling,† and Gabriel D. Maher†
Stanford University, Stanford, California 94305
Brent W. Barbee‡
NASA Goddard Space Flight Center, Greenbelt, Maryland 20771
and
Marco Pavone§
Stanford University, Stanford, California 94305

DOI: 10.2514/1.G000324

This paper presents a sampling-based motion planning algorithm for real-time and propellant-optimized autonomous spacecraft trajectory generation in near-circular orbits. Specifically, this paper leverages recent algorithmic advances in the field of robot motion planning to the problem of impulsively actuated, propellant-optimized rendezvous and proximity operations under the Clohessy–Wiltshire–Hill dynamics model. The approach calls upon a modified version of the FMT* algorithm to grow a set of feasible trajectories over a deterministic, low-dispersion set of sample points covering the free state space. To enforce safety, the tree is only grown over the subset of actively safe samples, from which there exists a feasible one-burn collision-avoidance maneuver that can safely circularize the spacecraft orbit along its coasting arc under a given set of potential thruster failures. Key features of the proposed algorithm include 1) theoretical guarantees in terms of trajectory safety and performance, 2) amenability to real-time implementation, and 3) generality, in the sense that a large class of constraints can be handled directly. As a result, the proposed algorithm offers the potential for widespread application, ranging from on-orbit satellite servicing to orbital debris removal and autonomous inspection missions.

I. Introduction

REAL-time guidance for spacecraft proximity operations near circular orbits is an inherently challenging task, particularly for onboard applications for which computational capabilities are limited. Fortunately, for the unconstrained case, many effective real-time solutions have been developed (e.g., state transition matrix manipulation [1], Lambert targeting [2], glideslope methods [3], safety ellipses [4], and others [5]). However, the difficulty of real-time processing increases when there is a need to operate near other objects and/or incorporate some notion of propellant optimality or control-effort minimization. In such cases, care is needed to efficiently handle collision-avoidance, plume-impingement, sensor line-of-sight, and other complex guidance constraints, which are often nonconvex and may depend on time and a mixture of state and control variables. State-of-the-art techniques for collision-free spacecraft proximity operations (both with and without optimality guarantees) include artificial potential function guidance [6,7], nonlinear trajectory optimization with [8,9] or without [10] convexification, line-of-sight or approach corridor constraints [11–13], relative separation techniques [14], Keep-Out Zone (KOZ) constraints with mixed-integer programming [15], and kinodynamic motion planning algorithms [16–19].

Requiring hard assurances of mission safety with respect to a wide variety and number of potential failure modes [20] provides an additional challenge. Often the concept of passive safety (safety certifications on zero-control-effort failure trajectories) over a finite horizon is employed to account for the possibility of control failures, though this potentially neglects mission-saving opportunities and fails to certify safety for all time. A less conservative alternative that more readily adapts to infinite horizons, as we will see, is to use active safety in the form of positively invariant set constraints. For instance, [11] enforces infinite-horizon active safety for a spacecraft by requiring each terminal state to lie on a collision-free orbit of equal period to the target. Reference [17] achieves a similar effect by only planning between waypoints that lie on circular orbits (a more restrictive constraint). Likewise, [21] requires an autonomous spacecraft to maintain access to at least one safe forced equilibrium point nearby. Finally, [22] devises the Safe and Robust Model Predictive Control algorithm, which employs invariant feedback tubes about a nominal trajectory (which guarantee resolvability) together with positively-invariant sets (taken about reference safety states) designed to be available at all times over the planning horizon.

The objective of this paper is to design an automated approach to actively-safe spacecraft trajectory optimization for rendezvous and proximity operations near circular orbits, which we model using Clohessy–Wiltshire–Hill (CWH) dynamics. Our approach leverages recent advances from the field of robot motion planning, particularly sampling-based motion planning [23]. Several decades of research in the robotics community have shown that sampling-based planning algorithms (dubbed "planners" throughout this paper) show promise for tightly constrained, high-dimensional optimal control problems such as the one considered in this paper. Sampling-based motion planning essentially breaks down a complex trajectory control problem into a series of many local relaxed two-point boundary value problems (2PBVPs, or steering problems) between intermediate waypoints (called samples), which are later evaluated *a posteriori* for constraint satisfaction and efficiently strung together into a graph (i.e., a tree or roadmap). By moving complex constraints like obstacle avoidance or plume impingement into *a posteriori* evaluation, we can decouple trajectory generation from constraint checking, a fact we exploit to achieve real-time capability. Critically, this approach avoids the explicit construction of the unconstrained state space, a computationally prohibitive task for complex planning problems. In

Received 4 December 2015; revision received 15 June 2016; accepted for publication 18 June 2016; published online 23 August 2016.

*Graduate Student, Aeronautics and Astronautics, 496 Lomita Mall, Room 009.

†Graduate Student, Institute for Computational and Mathematical Engineering, 475 Via Ortega.

‡Aerospace Engineer, Navigation and Mission Design Branch, Mail Code 595, 8800 Greenbelt Road. Senior Member AIAA.

§Assistant Professor, Aeronautics and Astronautics, 496 Lomita Mall, Room 261. Member AIAA.

this way, sampling-based algorithms can address a large variety of constraints and provide significant computational benefits with respect to traditional optimal control methods and mixed-integer programming [23]. Furthermore, through a property called asymptotic optimality (AO), sampling-based algorithms can be designed to provide guarantees of optimality in the limit that the number of samples taken approaches infinity. This makes sampling-based planners a strong choice for the problem of spacecraft control.

Although the aforementioned works [16–19] on sampling-based planning for spacecraft proximity operations have addressed several components of the safety-constrained, optimal CWH autonomous rendezvous problem, few have addressed the aspect of real-time implementability in conjunction with both a 2-norm propellant-cost metric and active trajectory safety with respect to control failures. This paper seeks to fill this gap. The paper's central theme is a rigorous proof of asymptotic optimality for a particular sampling-based planner, namely a modified version of the FMT* algorithm [24], under impulsive CWH spacecraft dynamics with hard safety constraints. First, a description of the problem scenario is provided in Sec. II, along with a formal definition of the sum-of-2-norms cost metric that we employ as a proxy for propellant consumption. Section III then follows with a thorough discussion of chaser/target vehicle safety, defining precisely how abort trajectories may be designed under CWH dynamics to deterministically avoid for all future times an ellipsoidal region about the CWH frame origin under a prescribed set of control failures. Next, we proceed in Sec. IV to our proposed approach employing the modified FMT* algorithm. The section begins with presentation of a conservative approximation to the propellant-cost reachability set, which characterizes the set of states that are nearby a given initial state in terms of propellant use. These sets, bounded by unions of ellipsoidal balls, are then used to show that the modified FMT* algorithm maintains its (asymptotic) optimality when applied to CWH dynamics under our propellant-cost metric. From there, in Sec. V, the paper presents two techniques for improving motion planning solutions: 1) an analytical technique that can be called both during planning and postprocessing to merge $\Delta \boldsymbol{v}$ vectors between any pair of concatenated graph edges and 2) a continuous trajectory smoothing algorithm, which can reduce the magnitude of $\Delta \boldsymbol{v}$ vectors by relaxing the implicit constraint to pass through sample points while still maintaining solution feasibility.

Put together, these tools yield a flexible, real-time framework for near-circular orbit spacecraft guidance that handles a wide variety of (possibly nonconvex) state, time, and control constraints and provides deterministic guarantees on abort safety and solution quality (propellant cost). The methodology is demonstrated in Sec. VI on a single-chaser, single-target scenario simulating a near-field low-Earth-orbit (LEO) approach with hard constraints on the total maneuver duration, relative positioning (including KOZ and antenna interference constraints), thruster plume-impingement avoidance, individual and net $\Delta \boldsymbol{v}$-vector magnitudes, and a two-fault thruster stuck-off failure tolerance. Trades are then conducted studying the effects of the sample count and a propellant cost threshold on the performance of FMT* (both with and without trajectory smoothing).

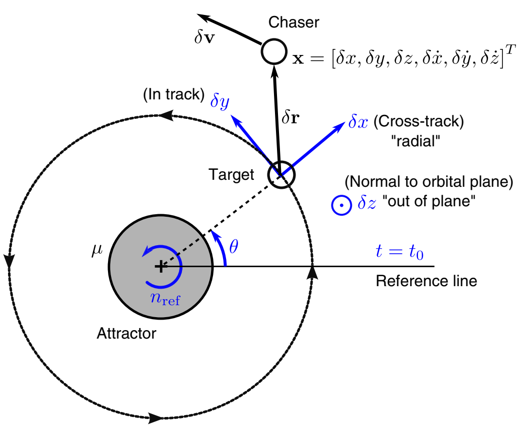
Preliminary versions of this paper appear in [25,26]. This extended and revised work introduces the following as additional contributions: 1) a more detailed presentation of the FMT* optimality proof, 2) improved trajectory smoothing, and 3) a six-dimensional (3 degree-of-freedom) numerical example demonstrating nonplanar LEO rendezvous.

II. Problem Formulation

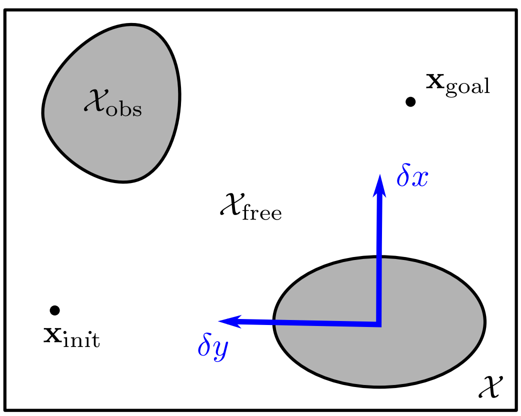
Consider the problem of a chaser spacecraft seeking to maneuver toward a single target moving in a well-defined, circular orbit (see Fig. 1a). Define the state space $\mathcal{X} \subset \mathbb{R}^d$ as a d-dimensional region in the target's local vertical, local horizontal (LVLH) frame, and let the obstacle region or $\mathcal{X}_{\text{obs}}$ be the set of states in $\mathcal{X}$ that result in mission failure (e.g., states outside of an approach corridor or in collision with the target). Let the free space or $\mathcal{X}_{\text{free}}$ be the complement of $\mathcal{X}_{\text{obs}}$. As seen in Fig. 1b, let $\boldsymbol{x}_{\text{init}}$ represent the chaser's initial state relative to the target, and let $\boldsymbol{x}_{\text{goal}} \in \mathcal{X}_{\text{goal}}$ be a goal state inside goal region $\mathcal{X}_{\text{goal}}$. Finally, define a state trajectory (or simply "trajectory") as a piecewise-continuous function of time $\boldsymbol{x}(t)$: $\mathbb{R} \to \mathcal{X}$, and let Σ represent the set of all state trajectories. Every state trajectory is implicitly generated by a control trajectory $\boldsymbol{u}(t)$: $\mathbb{R} \to \mathcal{U}$, where $\mathcal{U}$ is the set of controls, through the system dynamics $\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u}, t)$, where f is the state transition function. A state trajectory is called a feasible solution to the planning problem ($\mathcal{X}_{\text{free}}$, t_{init}, $\boldsymbol{x}_{\text{init}}$, $\boldsymbol{x}_{\text{goal}}$) if: 1) it satisfies the boundary conditions $\boldsymbol{x}(t_{\text{init}}) = \boldsymbol{x}_{\text{init}}$ and $\boldsymbol{x}(t_{\text{final}}) = \boldsymbol{x}_{\text{goal}}$ for some time $t_{\text{final}} > t_{\text{init}}$; 2) it is collision free, that is, $\boldsymbol{x}(t) \in \mathcal{X}_{\text{free}}$ for all $t \in [t_{\text{init}}, t_{\text{final}}]$; and 3) it obeys all other trajectory constraints. The optimal motion planning problem can then be defined as follows.

Definition 1 (Optimal Planning Problem)*:* Given a planning problem ($\mathcal{X}_{\text{free}}, t_{\text{init}}, \boldsymbol{x}_{\text{init}}, \boldsymbol{x}_{\text{goal}}$) and a cost functional J: $\Sigma \times \mathcal{U} \times \mathbb{R} \to \mathbb{R}_{\geq 0}$, find a feasible trajectory $\boldsymbol{x}^*(t)$ with associated control trajectory $\boldsymbol{u}^*(t)$ and time span $t = [t_{\text{init}}, t_{\text{final}}]$ for $t_{\text{final}} \in [t_{\text{init}}, \infty)$ such that $J(\boldsymbol{x}^*(\cdot), \boldsymbol{u}^*(\cdot), t) = \min\{J(\boldsymbol{x}(\cdot), \boldsymbol{u}(\cdot), t) \mid \boldsymbol{x}(t) \text{ and } \boldsymbol{u}(t) \text{ are feasible}\}$. If no such trajectory exists, report failure.

Tailoring Definition 1 to impulsively actuated propellant-optimal motion planning near circular orbits (using a control-effort cost functional J that considers only $\boldsymbol{u}(t)$ and the final time t_{final}, denoted as $J[\boldsymbol{u}(t), t_{\text{final}})]$), the optimal spacecraft motion planning problem may be formulated as



a) Schematic of CWH dynamics, which models relative guidance near a target in circular orbit



b) A representative motion planning query between feasible states $\mathbf{x}_{\text{init}}$ and $\mathbf{x}_{\text{goal}}$

Fig. 1 Illustration of the CWH planning scenario.

$$\text{Given: initial state } \boldsymbol{x}_{\text{init}}(t_{\text{init}}), \quad \text{goal region } \mathcal{X}_{\text{goal}}, \quad \text{free space } \mathcal{X}_{\text{free}}$$

$$\begin{aligned}
&\underset{\boldsymbol{u}(t), t_{\text{final}}}{\text{minimize}} \; J[\boldsymbol{u}(t), t_{\text{final}}] = \int_{t_{\text{init}}}^{t_{\text{final}}} \|\boldsymbol{u}(t)\|_2 \,\mathrm{d}t = \sum_{i=1}^{N} \|\Delta\boldsymbol{v}_i\|_2 \\
&\text{subject to } \boldsymbol{x}(t_{\text{init}}) = \boldsymbol{x}_{\text{init}} \quad \text{initial condition} \\
&\qquad \boldsymbol{x}(t_{\text{final}}) \in \mathcal{X}_{\text{goal}} \quad \text{terminal condition} \\
&\qquad \dot{\boldsymbol{x}}(t) = f[\boldsymbol{x}(t), \boldsymbol{u}(t), t] \quad \text{system dynamics} \\
&\qquad \boldsymbol{x}(t) \in \mathcal{X}_{\text{free}} \;\; \text{for all } t \in [t_{\text{init}}, t_{\text{final}}] \quad \text{obstacle avoidance} \\
&\qquad \left.\begin{array}{l} \boldsymbol{g}[\boldsymbol{x}(t), \boldsymbol{u}(t), t] \le 0 \\ \boldsymbol{h}[\boldsymbol{x}(t), \boldsymbol{u}(t), t] = 0 \end{array}\right. \;\; \text{for all } t \in [t_{\text{init}}, t_{\text{final}}] \quad \text{other constraints} \\
&\qquad \exists \text{ safe } \boldsymbol{x}_{\text{CAM}}(\tau), \quad \tau > t \;\; \text{for all } \boldsymbol{x}(t) \quad \text{active safety}
\end{aligned} \tag{1}$$

where t_{init} and t_{final} are the initial and final times and $\boldsymbol{x}_{\text{CAM}}(\tau)$ refers to an infinite-horizon collision-avoidance maneuver (CAM). Note we restrict our attention to impulsive control laws $\boldsymbol{u}(t) = \sum_{i=1}^{N} \Delta\boldsymbol{v}_i \delta(t - \tau_i)$, where $\delta(\cdot)$ denotes the Dirac delta function, which models finite sequences of instantaneous translational burns $\Delta\boldsymbol{v}_i$ fired at discrete times τ_i (note that the number of burns N is not fixed *a priori*). Although it is possible to consider all control laws, it is both theoretically and computationally simpler to optimize over a finite-dimensional search space of $\Delta\boldsymbol{v}$ vectors; furthermore, these represent the most common forms of propulsion systems used on orbit (including high-impulse cold-gas and liquid bipropellant thrusters), and they can (at least in theory) approximate continuous control trajectories in the limit in which $N \to \infty$.

We now elaborate on the objective function and each constraint in turn.

A. Cost Functional

A critical component of our problem is the choice of cost functional. Consistent with [27], we define our cost as the L^1-norm of the ℓ_p-norm of the control. The best choice for $p \ge 1$ depends on the propulsion system geometry and on the frame within which $\boldsymbol{u}(t) = \sum_{i=1}^{N} \Delta\boldsymbol{v}_i \delta(t - \tau_i)$ in J is resolved. Unfortunately, minimizing the propellant exactly involves resolving vectors $\Delta\boldsymbol{v}_i$ into the spacecraft body-fixed frame, requiring the attitude $\boldsymbol{q}$ to be included in our state $\boldsymbol{x}$. To avoid this, a common standard is to employ $p = 2$ so that each $\Delta\boldsymbol{v}_i$ is as short as possible, allocating the commanded $\Delta\boldsymbol{v}_i$ to thrusters in a separate control allocation step (conducted later, once the attitude is known; see Sec. II.E). Although this moves propellant minimization online, it greatly simplifies the guidance problem in a practical way without neglecting attitude. Because the cost of $\Delta\boldsymbol{v}$ allocation can only grow from the need to satisfy torque constraints or impulse bounds (e.g., necessitating counteropposing thrusters to achieve the same net $\Delta\boldsymbol{v}$ vector), we are in effect minimizing the best-case, unconstrained propellant use of the spacecraft. As we will show in our numerical experiments, however, this does not detract significantly from the technique; the coupling of J with $p = 2$ to the actual propellant use through the minimum control-effort thruster $\Delta\boldsymbol{v}$ allocation problem seems to promote low propellant-cost solutions. Hence, J serves as a good proxy to propellant use, with the added benefit of independence from propulsion system geometry.

B. Boundary Conditions

Planning is assumed to begin at a known initial state $\boldsymbol{x}_{\text{init}}$ and time t_{init} and end at a single goal state $\boldsymbol{x}_{\text{goal}}^T = [\delta\boldsymbol{r}_{\text{goal}}^T, \;\; \delta\boldsymbol{v}_{\text{goal}}^T]$ (exact convergence, $\mathcal{X}_{\text{goal}} = \{\boldsymbol{x}_{\text{goal}}\}$), where $\delta\boldsymbol{r}_{\text{goal}}$ is the goal position and $\delta\boldsymbol{v}_{\text{goal}}$ is the goal velocity. During numerical experiments, however, we sometimes permit termination at any state of which the position and velocity lie within Euclidean balls $\mathcal{B}(\delta\boldsymbol{r}_{\text{goal}}, \epsilon_r)$ and $\mathcal{B}(\delta\boldsymbol{v}_{\text{goal}}, \epsilon_v)$, respectively (inexact convergence, $\mathcal{X}_{\text{goal}} = \mathcal{B}(\boldsymbol{r}_{\text{goal}}, \epsilon_r) \times \mathcal{B}(\boldsymbol{v}_{\text{goal}}, \epsilon_v)$), where the notation $\mathcal{B}(\boldsymbol{r}, \epsilon) = \{\boldsymbol{x} \in \mathcal{X} \,|\, \|\boldsymbol{r} - \boldsymbol{x}\| \le \epsilon\}$ denotes a ball with center $\boldsymbol{r}$ and radius ϵ.

C. System Dynamics

Because spacecraft proximity operations incorporate significant drift, spatially dependent external forces, and changes on fast time scales, any realistic solution must obey dynamic constraints; we cannot assume straight-line trajectories. In this paper, we employ the classical CWH equations [28,29] for impulsive linearized motion about a circular reference orbit at radius r_{ref} about an inverse-square-law gravitational attractor with parameter μ. This model provides a first-order approximation to a chaser spacecraft's motion relative to a rotating target-centered coordinate system (see Fig. 1). The linearized equations of motion for this scenario as resolved in the LVLH frame of the target are given by

$$\delta\ddot{x} - 3n_{\text{ref}}^2 \delta x - 2n_{\text{ref}} \delta\dot{y} = \frac{F_{\delta x}}{m} \tag{2a}$$

$$\delta\ddot{y} + 2n_{\text{ref}} \delta\dot{x} = \frac{F_{\delta y}}{m} \tag{2b}$$

$$\delta\ddot{z} + n_{\text{ref}}^2 \delta z = \frac{F_{\delta z}}{m} \tag{2c}$$

where $n_{\text{ref}} = \sqrt{\frac{\mu}{r_{\text{ref}}^3}}$ is the orbital frequency (mean motion) of the reference spacecraft orbit; m is the spacecraft mass; $\boldsymbol{F} = [F_{\delta x}, F_{\delta y}, F_{\delta z}]^T$ is some applied force; and $(\delta x, \delta y, \delta z)$ and $(\delta\dot{x}, \delta\dot{y}, \delta\dot{z})$ represent the cross-track (radial), in-track, and out-of-plane relative position and velocity vectors, respectively. The CWH model is used often, especially for short-duration rendezvous and proximity operations in LEO and for leader–follower formation flight dynamics.

Defining the state $\boldsymbol{x}$ as $[\delta x, \delta y, \delta z, \delta\dot{x}, \delta\dot{y}, \delta\dot{z}]^T$ and the control $\boldsymbol{u}$ as the applied force per unit mass $\frac{1}{m}\boldsymbol{F}$, the CWH equations can be described by the linear time-invariant (LTI) system,

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u}, t) = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} \tag{3}$$

where the dynamics matrix $\boldsymbol{A}$ and input matrix $\boldsymbol{B}$ are given by

$$\boldsymbol{A} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 3n_{\text{ref}}^2 & 0 & 0 & 0 & 2n_{\text{ref}} & 0 \\ 0 & 0 & 0 & -2n_{\text{ref}} & 0 & 0 \\ 0 & 0 & -n_{\text{ref}}^2 & 0 & 0 & 0 \end{bmatrix} \qquad \boldsymbol{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

As for any LTI system, there exists a unique solution to Eq. (3) given initial condition $\boldsymbol{x}(t_0)$ and integrable input $\boldsymbol{u}(t)$ that can be expressed using superposition and the convolution integral as $\boldsymbol{x}(t) = e^{A(t-t_0)}\boldsymbol{x}(t_0) + \int_{t_0}^{t} e^{A(t-\tau)}\boldsymbol{B}\boldsymbol{u}(\tau)\,\mathrm{d}\tau$ for any time $t \ge t_0$. The expression $\boldsymbol{\Phi}(t, \tau) \triangleq e^{A(t-\tau)}$ is called the state transition matrix, which importantly provides an analytical mechanism for computing state trajectories that we rely on heavily in simulations. Note, throughout this work, we sometimes represent $\boldsymbol{\Phi}(t, \tau)$ as $\boldsymbol{\Phi}$ when its arguments are understood.

We now specialize our solution to the case of N impulsive velocity changes at times $t_0 \le \tau_i \le t_f$ for $i \in [1, \ldots, N]$, in which case $\boldsymbol{u}(\tau) = \sum_{i=1}^{N} \Delta\boldsymbol{v}_i \delta(\tau - \tau_i)$, where $\delta(y) = \{1 \text{ where } y = 0, \text{ or } 0 \text{ otherwise}\}$ signifies the Dirac-delta distribution. Substituting for $\boldsymbol{\Phi}$ and $\boldsymbol{u}(\tau)$,

$$\begin{aligned}
\boldsymbol{x}(t) &= \boldsymbol{\Phi}(t, t_0)\boldsymbol{x}(t_0) + \int_{t_0}^{t} \boldsymbol{\Phi}(t, \tau)\boldsymbol{B}\left[\sum_{i=1}^{N} \Delta\boldsymbol{v}_i \delta(\tau - \tau_i)\right]\mathrm{d}\tau \\
&= \boldsymbol{\Phi}(t, t_0)\boldsymbol{x}(t_0) + \sum_{i=1}^{N} \int_{t_0}^{t} \boldsymbol{\Phi}(t, \tau)\boldsymbol{B}\Delta\boldsymbol{v}_i \delta(\tau - \tau_i)\,\mathrm{d}\tau
\end{aligned}$$

where on the second line we used the linearity of the integral operator. By the sifting property of δ, denoting $N_t = \sum_{i=1}^{N} \mathbb{1}(\tau_i \le t)$ as the number of burns applied from t_0 up to time t, we have for all times $t \ge t_0$ the following expression for the impulsive solution to Eq. (3):

$$\boldsymbol{x}(t) = \boldsymbol{\Phi}(t, t_0)\boldsymbol{x}(t_0) + \sum_{i=1}^{N_t} \boldsymbol{\Phi}(t, \tau_i)\boldsymbol{B}\boldsymbol{\Delta v}_i \tag{4a}$$

$$= \boldsymbol{\Phi}(t, t_0)\boldsymbol{x}(t_0) + \underbrace{[\boldsymbol{\Phi}(t, \tau_1)\boldsymbol{B} \quad \ldots \quad \boldsymbol{\Phi}(t, \tau_{N_t})\boldsymbol{B}]}_{\triangleq \Phi_v(t,\{\tau_i\}_i)} \underbrace{\begin{bmatrix} \boldsymbol{\Delta v}_1 \\ \vdots \\ \boldsymbol{\Delta v}_{N_t} \end{bmatrix}}_{\triangleq \Delta V} \tag{4b}$$

$$= \boldsymbol{\Phi}(t, t_0)\boldsymbol{x}(t_0) + \boldsymbol{\Phi}_v(t, \{\tau_i\}_i)\boldsymbol{\Delta V} \tag{4c}$$

Throughout this paper, the notations $\boldsymbol{\Delta V}$ for the stacked Δv vector and $\boldsymbol{\Phi}_v(t, \{\tau_i\}_i)$ for the aggregated impulse state transition matrix (or simply $\boldsymbol{\Phi}_v$ for short, when the parameters t and $\{\tau_i\}_i$ are clear) implicitly imply only those burns i occurring before time t.

D. Obstacle Avoidance

Obstacle avoidance is imposed by requiring that the solution $\boldsymbol{x}(t)$ stay within $\mathcal{X}_{\text{free}}$ (or, equivalently, outside of $\mathcal{X}_{\text{obs}}$), typically a difficult nonconvex constraint. For CWH proximity operations, $\mathcal{X}_{\text{obs}}$ might include positions in collision with a nearby object, position/velocity pairs outside of a given approach corridor, etc. In our numerical experiments, to prevent the chaser from interfering with the target, we assume $\mathcal{X}_{\text{obs}}$ comprises an ellipsoidal KOZ centered at the origin and a conical nadir-pointing region that approximates the target's antenna beam pattern.

Note that, according to the definition of $\mathcal{X}_{\text{free}}$, this also requires $\boldsymbol{x}(t)$ to stay within $\mathcal{X}$ (CWH dynamics do not guarantee that state trajectories will lie inside $\mathcal{X}$ despite the fact that their endpoints do). Although not always necessary in practice, if $\mathcal{X}$ marks the extent of reliable sensor readings or the boundary inside which linearity assumptions hold, then this can be useful to enforce.

E. Other Trajectory Constraints

Many other types of constraints may be included to encode additional restrictions on state and control trajectories, which we represent here by a set of inequality constraints $\boldsymbol{g}$ and equality constraints $\boldsymbol{h}$ (note that $\boldsymbol{g}$ and $\boldsymbol{h}$ denote vector functions). To illustrate the flexibility of sampling-based planning, we encode the following into constraints $\boldsymbol{g}$ (for brevity, we omit their exact representation, which is a straightforward exercise in vector geometry):

$$\begin{aligned}
&T_{\text{plan,max}} \le t_{\text{final}} - t_{\text{init}} \le T_{\text{plan,max}} && \text{plan duration bounds} \\
&\boldsymbol{\Delta v}_i \in \mathcal{U}(\boldsymbol{x}(\tau_i)) \quad \text{for all } i = [1, \ldots, N] && \text{control feasibility} \\
&\bigcup_{k \in [1,\ldots,K]} \mathcal{P}_{ik}(\Delta \hat{\boldsymbol{v}}_k, \beta_{\text{plume}}, H_{\text{plume}}) \cap \mathbb{S}_{\text{target}} = \varnothing \quad \text{for all } i = [1, \ldots, N] && \text{plume impingement avoidance}
\end{aligned}$$

Here, $0 \le T_{\text{plan,min}} < T_{\text{plan,max}}$ represent minimum and maximum motion plan durations, $\mathcal{U}[\boldsymbol{x}(\tau_i)]$ is the admissible control set corresponding to state $\boldsymbol{x}(\tau_i)$, $\mathcal{P}_{ik}$ is the exhaust plume emanating from thruster k of the chaser spacecraft while executing burn $\boldsymbol{\Delta v}_i$ at time τ_i, and $\mathbb{S}_{\text{target}}$ is the target spacecraft circumscribing sphere. We motivate each constraint in turn.

1. Plan Duration Bounds

Plan duration bounds facilitate the inclusion of rendezvous windows based on the epoch of the chaser at $\boldsymbol{x}_{\text{init}}(t_{\text{init}})$, as determined by illumination requirements, ground communication opportunities, or mission timing restrictions, for example. $T_{\text{plan,max}}$ can also ensure that the errors incurred by linearization, which grow with time, do not exceed acceptable tolerances.

2. Control Feasibility

Control set constraints are intended to encapsulate limitations on control authority imposed by propulsive actuators and their geometric distribution about the spacecraft. For example, given the maximum burn magnitude $0 < \Delta v_{\max}$, the constraint

$$\|\boldsymbol{\Delta v}_i\|_2 \le \Delta v_{\max} \quad \text{for all } i = [1, \ldots, N] \tag{5}$$

might represent an upper bound on the achievable impulses of a gimbaled thruster system that is free to direct thrust in all directions. In our case, we use $\mathcal{U}[\boldsymbol{x}(\tau_i)]$ to represent all commanded net $\boldsymbol{\Delta v}$ vectors that 1) satisfy Eq. (5) and also 2) can be successfully allocated to thrusters along trajectory $\boldsymbol{x}(t)$ at time τ_i according to a simple minimum-control-effort thruster allocation problem (a straightforward linear program [30]). To keep the paper self-contained, we repeat the problem here and in our own notation. Let $\boldsymbol{\Delta v}_i|_{\text{bf}}$ and $\boldsymbol{M}_i|_{\text{bf}}$ be the desired net $\boldsymbol{\Delta v}$ and moment vectors at burn time τ_i, resolved in the body-fixed frame according to attitude $\boldsymbol{q}(\tau_i)$ (we henceforth drop the bar, for clarity). Note the attitude $\boldsymbol{q}(\tau_i)$ must either be included in the state $\boldsymbol{x}(\tau_i)$ or be derived from it, as we assume here by imposing (along nominal trajectories) a nadir-pointing attitude profile for the chaser spacecraft. Let $\Delta v_{ik} = \|\boldsymbol{\Delta v}_{ik}\|_2$ be the Δv magnitude allocated to thruster k, which generates an impulse in direction $\Delta \hat{\boldsymbol{v}}_{ik}$ at position $\boldsymbol{\rho}_{ik}$ from the spacecraft center-of-mass (both are constant vectors if resolved in the body-fixed frame). Finally, to account for the possibility of on or off thrusters, let η_{ik} be equal to 1, if thruster k is available for burn i, or zero otherwise. Then, the minimum-effort control allocation problem can be represented as

$$\begin{aligned}
\text{Given: } & \text{on-of flags } \eta_{ik}, \text{thruster positions } \boldsymbol{\rho}_{ik}, \text{thruster axes } \Delta \hat{\boldsymbol{v}}_{ik}, \\
& \text{commanded } \boldsymbol{\Delta v}\text{-vector } \boldsymbol{\Delta v}_i, \text{and commanded} \\
& \text{moment vector } \boldsymbol{M}_i \\
\underset{\Delta v_{ik}}{\text{minimize}} \ & \sum_{k=1}^{K} \Delta v_{ik} \\
\text{subject to } & \sum_{k=1}^{K} \Delta \hat{\boldsymbol{v}}_{ik}(\eta_{ik}\Delta v_{ik}) = \boldsymbol{\Delta v}_i \qquad \text{net } \boldsymbol{\Delta v}\text{-vector allocation} \\
& \sum_{k=1}^{K} (\boldsymbol{\rho}_{ik} \times \Delta \hat{\boldsymbol{v}}_i)(\eta_{ik}\Delta v_{ik}) = \boldsymbol{M}_i \qquad \text{net moment allocation} \\
& \Delta v_{\min,k} \le \Delta v_{ik} \le \Delta v_{\max,k} \qquad \text{thruster } \boldsymbol{\Delta v} \text{ bounds}
\end{aligned} \tag{6}$$

where $\Delta v_{\min,k}$ and $\Delta v_{\max,k}$ represent minimum- and maximum-impulse limits on thruster k (due to actuator limitations, minimum impulse bits, pulse-width constraints, or maximum on-time restrictions, for example). Note that, by combining the minimization of commanded $\boldsymbol{\Delta v}$-vector lengths $\|\boldsymbol{\Delta v}_i\|$ with minimum-effort allocation Δv^*_{ik} to thrusters k in Eq. (6), the previous formulation

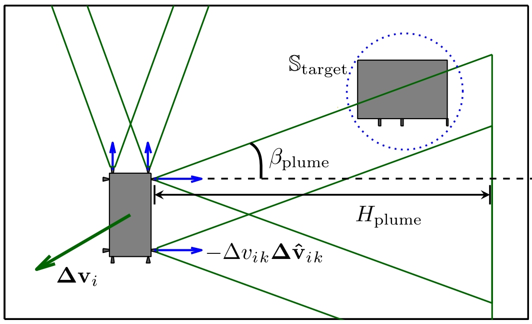


Fig. 2 Illustration of exhaust plume impingement from thruster firings.

corresponds directly to minimum-propellant consumption (by the Tsiolkovsky rocket equation, subject to our thrust bounds and net torque constraints); see also Sec. II.A. In this work, we set $\boldsymbol{M}_i = 0$ to enforce torque-free burns and minimize disturbances to our assumed attitude trajectory $\boldsymbol{q}(t)$.

Note that we do not consider a minimum-norm constraint in Eq. (5) for $\boldsymbol{\Delta v}_i$, as it is not necessary and would significantly complicate the theoretical characterization of our proposed planning algorithms, provided in Sec. IV. As discussed in Sec. II.A, $\|\boldsymbol{\Delta v}_i\|$ is only a proxy for the true propellant cost computed from the thrust allocation problem [Eq. (6)].

3. Plume Impingement

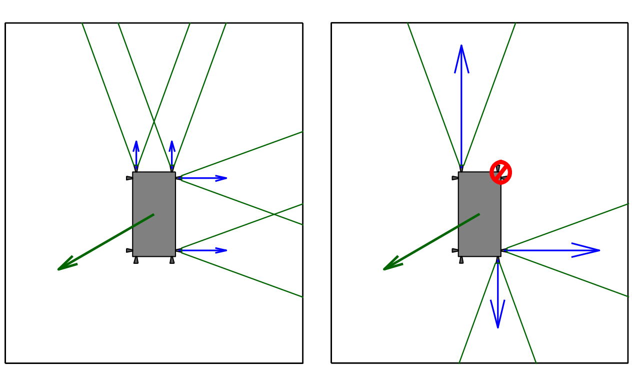
Impingement of thruster exhaust on neighboring spacecraft can lead to dire consequences, including destabilizing effects on attitude caused by exhaust gas pressure, degradation of sensitive optical equipment and solar arrays, and unexpected thermal loading [31,32]. To account for this during guidance, we first generate representative exhaust plumes at the locations of each thruster firing. For burn i occurring at time τ_i, a right circular cone with axis $-\boldsymbol{\Delta}\hat{\boldsymbol{v}}_{ik}$, half-angle β_{plume}, and height H_{plume} is projected from each active thruster k ($\eta_{ik} = 1$), the allocated thrust Δv^*_{ik} of which is nonzero, as determined by Eq. (6). Intersections are then checked with the target spacecraft circumscribing sphere $\mathbb{S}_{\text{target}}$, which is used as a simple conservative approximation to the exact target geometry. For an illustration of the process, refer to Fig. 2.

4. Other Constraints

Other constraints may easily be added. Solar array shadowing, pointing constraints, approach corridors, and so forth all fit within the framework and may be represented as additional inequality or equality constraints. For more, we refer the interested reader to [33].

F. Active Safety

An additional feature we include in our work is the concept of active safety, in which we require the target spacecraft to maintain a feasible CAM to a safe higher or lower circular orbit from every point along its solution trajectory in the event that any mission-threatening control degradations take place, such as stuck-off thrusters (as in Fig. 3). This reflects our previous work [25] and is detailed more thoroughly in Sec. III.



a) Thruster allocation without stuck-off failures

b) The same allocation problem, with both upper-right thrusters stuck off

Fig. 3 Changes to torque-free control allocation in response to thruster failures.

III. Vehicle Safety

In this section, we devise a general strategy for handling the active safety constraints introduced in Eq. (1) and Sec. II.F, which we use to guarantee solution safety under potential control failures. Specifically, we examine how to ensure in real-time that safe abort trajectories are always available to the spacecraft up to a given number of thruster stuck-off failures. As will be motivated, the idea behind our approach is to couple positively invariant set safety constraints with escape trajectory generation and embed them into the sampling routines of deterministic sampling-based motion planners. We prioritize active safety measures in this section (which allow actuated CAMs over passive safety guarantees (which shut off all thrusters and restrict the system to zero control) in order to broaden the search space for abort trajectories. Because of the propellant-limited nature of many spacecraft proximity operations missions, emphasis is placed on finding minimum-$\boldsymbol{\Delta v}$ escape maneuvers in order to improve mission reattempt opportunities. In many ways, we emulate the rendezvous design process taken by Barbee et al. [34] but numerically optimize abort propellant consumption and remove much of its reliance on user intuition by automating the satisfaction of safety constraints.

Consistent with the notions proposed by Schouwenaars et al. [35]; Fehse [36] Sec. 4.1.2; and Fraichard [37], our general definition for vehicle safety is taken to be the following.

Definition 2 (vehicle safety)*:* A vehicle state is safe if and only if there exists, under the worst-possible environment and failure conditions, a collision-free, dynamically feasible trajectory satisfying the constraints that navigates the vehicle to a set of states in which it can remain indefinitely.

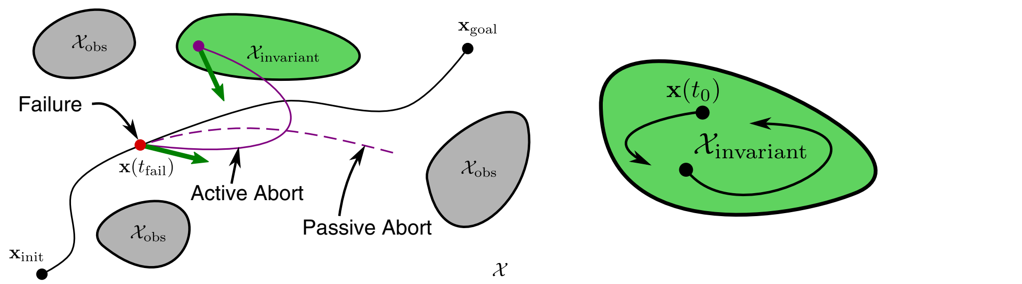
Note "indefinitely" (or "sufficiently long" for all practical purposes under the accuracy of the dynamics model) is a critical component of the definition. Trajectories without infinite-horizon safety guarantees can ultimately violate constraints [11], thereby posing a risk that can defeat the purpose of using a hard guarantee in the first place. For this reason, we impose safety constraints over an infinite-horizon (or, as we will show using invariant sets, an effectively infinite horizon).

Consider the scenario described in Sec. II for a spacecraft with nominal state trajectory $\boldsymbol{x}(t) \in \mathcal{X}$ and control trajectory $\boldsymbol{u}(t) \in \mathcal{U}[\boldsymbol{x}(t)]$ evolving over time t in time span $\mathcal{T} = [t_{\text{init}}, \infty)$. Let $\mathcal{T}_{\text{fail}} \subseteq \mathcal{T}$ represent the set of potential failure times we wish to certify (for instance, a set of prescribed burn times $\{\tau_i\}$, the final approach phase $\mathcal{T}_{\text{approach}}$, or the entire maneuver span $\mathcal{T}$). When a failure occurs, control authority is lost through a reduction in actuator functionality, negatively impacting system controllability. Let $\mathcal{U}_{\text{fail}}(\boldsymbol{x}) \subset \mathcal{U}(\boldsymbol{x})$ represent the new control set, where we assume that $0 \in \mathcal{U}_{\text{fail}}$ for all $\boldsymbol{x}$ (i.e., we assume that no actuation is always a feasible option). Mission safety is commonly imposed in two different ways ([36] Sec. 4.4):

1) For all $t_{\text{fail}} \in \mathcal{T}_{\text{fail}}$, ensure that $\boldsymbol{x}_{\text{CAM}}(t)$ satisfies Definition 2 with $\boldsymbol{u}_{\text{CAM}}(t) = 0$ for all $t \geq t_{\text{fail}}$ (called passive safety). For a spacecraft, this means its coasting arc from the point of failure must be safe for all future time (though practically this is imposed only over a finite horizon).

2) For all $t_{\text{fail}} \in \mathcal{T}_{\text{fail}}$ and failure modes $\mathcal{U}_{\text{fail}}$, devise actuated collision-avoidance maneuvers $\boldsymbol{x}_{\text{CAM}}(t)$ that satisfy Definition 2 with $\boldsymbol{u}_{\text{CAM}}(t) \in \mathcal{U}_{\text{fail}}$ for all $t \geq t_{\text{fail}}$, where $\boldsymbol{u}_{\text{CAM}}(t)$ is not necessarily restricted to 0 (called active safety).

See Fig. 4a for an illustration. In much of the literature, only passive safety is considered out of a need for tractability (to avoid verification over a combinatorial explosion of failure mode possibilities) and to capture the common case in which control authority is lost completely. Although considerably simpler to



a) Comparing passive and active abort safety following the occurrence of a failure

b) A positively invariant set, within which state trajectories stay once entered

Fig. 4 Illustrations of various vehicle abort safety concepts.

implement, this approach potentially neglects many mission-saving control policies.

A. Active Safety Using Positively-Invariant Sets

Instead of ensuring safety for all future times $t \geq t_{\text{fail}}$, it is more practical to consider finite-time abort maneuvers starting at $\boldsymbol{x}(t_{\text{fail}})$ that terminate inside a safe positively invariant set $\mathcal{X}_{\text{invariant}}$. If the maneuver is safe and the invariant set is safe for all time, then vehicle safety is assured.

Definition 3 (positively invariant set)*:* A set $\mathcal{X}_{\text{invariant}}$ is positively invariant with respect to the autonomous system $\dot{\boldsymbol{x}}_{\text{CAM}} = f(\boldsymbol{x}_{\text{CAM}})$ if and only if $\boldsymbol{x}_{\text{CAM}}(t_{\text{fail}}) \in \mathcal{X}_{\text{invariant}}$ implies $\boldsymbol{x}_{\text{CAM}}(t) \in \mathcal{X}_{\text{invariant}}$ for all $t \geq t_{\text{fail}}$.

See Fig. 4b. This yields the following definition for finite-time verification of trajectory safety.

Definition 4 (finite-time trajectory safety verification)*:* For all $t_{\text{fail}} \in \mathcal{T}_{\text{fail}}$ and for all $\mathcal{U}_{\text{fail}}[\boldsymbol{x}(t_{\text{fail}})] \subset \mathcal{U}[\boldsymbol{x}(t_{\text{fail}})]$, there exists $\{\boldsymbol{u}(t), t \geq t_{\text{fail}}\} \in \mathcal{U}_{\text{fail}}[\boldsymbol{x}(t_{\text{fail}})]$ and $T_h > t_{\text{fail}}$ such that $\boldsymbol{x}(t)$ is feasible for all $t_{\text{fail}} \leq t \leq T_h$ and $\boldsymbol{x}(T_h) \in \mathcal{X}_{\text{invariant}} \subseteq \mathcal{X}_{\text{free}}$.

Here, T_h is some finite safety horizon time. Although, in principle, any safe positively invariant set $\mathcal{X}_{\text{invariant}}$ is acceptable, not just any will do in practice; in real-world scenarios, unstable trajectories caused by model uncertainties could cause state divergence toward configurations of which the safety has not been verified. Hence, care must be taken to use only stable positively-invariant sets.

Combining Definition 4 with our constraints in Eq. (1) from Sec. II, spacecraft trajectory safety after a failure at $\boldsymbol{x}(t_{\text{fail}}) = \boldsymbol{x}_{\text{fail}}$ can be expressed in its full generality as the following optimization problem in decision variables $T_h \in [t_{\text{fail}}, \infty)$, $\boldsymbol{x}_{\text{CAM}}(t)$, and $\boldsymbol{u}_{\text{CAM}}(t)$, for $t \in [t_{\text{fail}}, T_h]$:

original rendezvous. Typically, any feasible solution is sought following a failure, in which case one may use $J = 1$. However, to enhance the possibility of mission recovery, we assume the same minimum-propellant cost functional as before, but with the exception that here, as we will motivate, we use a single-burn strategy with $N = 1$.

B. Fault-Tolerant Safety Strategy

The difficulty of solving the finite-time trajectory safety problem lies in the fact that a feasible solution must be found for all possible failure times (typically assumed to be any time during the mission) as well as for all possible failures. To illustrate, for an F-fault tolerant spacecraft with K control components (thrusters, momentum wheels, control moment gyroscopes, etc.), each of which we each model as either operational or failed, this yields a total of $N_{\text{fail}} = \sum_{f=0}^{F} \binom{K}{f} = \sum_{f=0}^{F} \frac{K!}{(K-f)!f!}$ possible optimization problems that must be solved for every time t_{fail} along the nominal trajectory. By any standard, this is intractable and hence explains why so often passive safety guarantees are selected (requiring only one control configuration check instead of N_{fail}, since we prescribe $\boldsymbol{u}_{\text{CAM}} = 0$, which must lie in $\mathcal{U}_{\text{fail}}$ given our assumption; this is analogous to setting $f = K$ with $F \triangleq K$). One idea for simplifying this problem while still satisfying safety [the constraints of Eq. (7)] consists of the following strategy.

Definition 5 (fault-tolerant active safety strategy)*:* As a conservative solution to the optimization problem in Eq. (7), it is sufficient (but not necessary) to implement the following procedure:

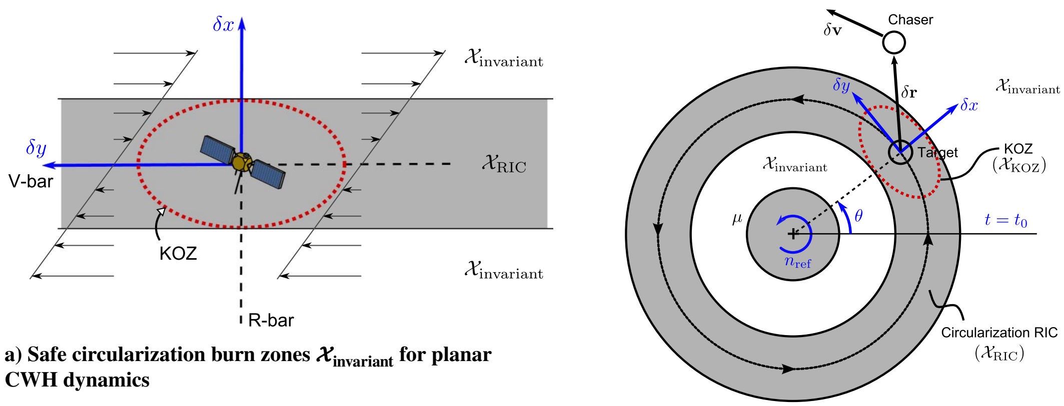
1) From each $\boldsymbol{x}(t_{\text{fail}})$, prescribe a CAM policy Π_{CAM} that gives a horizon time T_h and escape control sequence $\boldsymbol{u}_{\text{CAM}} = \Pi_{\text{CAM}}[\boldsymbol{x}(t_{\text{fail}})]$ designed to automatically satisfy $\boldsymbol{u}_{\text{CAM}}(\tau) \subset \mathcal{U}$ for all $t_{\text{fail}} \leq \tau \leq T_h$ and $\boldsymbol{x}(T_h) \in \mathcal{X}_{\text{invariant}}$.

$$
\begin{aligned}
&\text{Given: failure state } \boldsymbol{x}_{\text{fail}}(t_{\text{fail}}), \text{ failure control set } \mathcal{U}_{\text{fail}}(\boldsymbol{x}_{\text{fail}}), \text{ the free space } \mathcal{X}_{\text{free}}, \\
&\qquad \text{a safe, stable invariant set } \mathcal{X}_{\text{invariant}}, \text{ and a fixed number of impulses } N \\
&\underset{\substack{\boldsymbol{u}_{\text{CAM}}(t) \in \mathcal{U}_{\text{fail}}(\boldsymbol{x}_{\text{fail}}) \\ T_h, \boldsymbol{x}_{\text{CAM}}(t)}}{\text{minimize}} \quad J(\boldsymbol{x}_{\text{CAM}}(t), \boldsymbol{u}_{\text{CAM}}(t), t) = \int_{t_{\text{fail}}}^{T_h} \|\boldsymbol{u}_{\text{CAM}}(t)\|_2 \, \mathrm{d}t = \sum_{i=1}^{N} \|\Delta \boldsymbol{v}_{\text{CAM},i}\|_2 \\
&\text{subject to } \dot{\boldsymbol{x}}_{\text{CAM}}(t) = f[\boldsymbol{x}_{\text{CAM}}(t), \boldsymbol{u}_{\text{CAM}}(t), t] \qquad \text{system dynamics} \\
&\qquad \boldsymbol{x}_{\text{CAM}}(t_{\text{fail}}) = \boldsymbol{x}_{\text{fail}} \qquad \text{initial condition} \\
&\qquad \boldsymbol{x}_{\text{CAM}}(T_h) \in \mathcal{X}_{\text{invariant}} \qquad \text{safe termination} \\
&\qquad \boldsymbol{x}_{\text{CAM}}(t) \in \mathcal{X}_{\text{free}} \qquad \text{for all } t \in [t_{\text{fail}}, T_h] \qquad \text{obstacle avoidance} \\
&\qquad \left.\begin{array}{l} \boldsymbol{g}(\boldsymbol{x}_{\text{CAM}}, \boldsymbol{u}_{\text{CAM}}, t) \leq 0 \\ \boldsymbol{h}(\boldsymbol{x}_{\text{CAM}}, \boldsymbol{u}_{\text{CAM}}, t) = 0 \end{array}\right. \quad \text{for all } t \in [t_{\text{fail}}, T_h] \qquad \text{other constraints}
\end{aligned}
\tag{7}
$$

This is identical to Eq. (1), except that now, under failure mode $\mathcal{U}_{\text{fail}}(\boldsymbol{x}_{\text{fail}})$, we abandon the attempt to terminate at a goal state in $\mathcal{X}_{\text{goal}}$ and instead replace it with a constraint to terminate at a safe, stable positively invariant set $\mathcal{X}_{\text{invariant}}$. We additionally neglect any timing constraints encoded in $\boldsymbol{g}$ as we are no longer concerned with our

2) For each failure mode $\mathcal{U}_{\text{fail}}[\boldsymbol{x}(t_{\text{fail}})] \subset \mathcal{U}[\boldsymbol{x}(t_{\text{fail}})]$ up to tolerance F, determine if the control law is feasible; that is, see if $\boldsymbol{u}_{\text{CAM}} = \Pi_{\text{CAM}}[\boldsymbol{x}(t_{\text{fail}})] \subset \mathcal{U}_{\text{fail}}$ for the particular failure in question.

This effectively removes decision variables $\boldsymbol{u}_{\text{CAM}}$ from Eq. (7), allowing simple numerical integration for the satisfaction of the



a) Safe circularization burn zones $\mathcal{X}_{\text{invariant}}$ for planar CWH dynamics

b) Inertial view of the radial band spanned by the KOZ that defines the unsafe RIC

Fig. 5 Visualizing the safe and unsafe circularization regions used by the CAM safety policy.

dynamic constraints and a straightforward *a posteriori* verification of the other trajectory constraints (inclusion in $\mathcal{X}_{\text{free}}$ and satisfaction of constraints $\boldsymbol{g}$ and $\boldsymbol{h}$). This checks if the prescribed CAM, guaranteed to provide a safe escape route, can actually be accomplished in the given failure situation. The approach is conservative due to the fact that the control law is imposed and not derived; however, the advantage is a greatly simplified optimal control problem with difficult-to-handle constraints relegated to *a posteriori* checks, exactly identical to the way that steering trajectories are derived and verified during the planning process of sampling-based planning algorithms. Note that formal definitions of safety require that this be satisfied for all possible failure modes of the spacecraft; we do not avoid the combinatorial explosion of N_{fail}. However, each instance of problem Eq. (7) is greatly simplified, and with F typically at most 3, the problem remains tractable. The difficult part, then, lies in computing Π_{CAM}, but this can easily be generated offline. Hence, the strategy should work well for vehicles with difficult, nonconvex objective functions and constraints, as is precisely the case for CWH proximity operations.

Note that it is always possible to reduce this approach to the (more-conservative) definition of passive safety that has traditionally been seen in the literature by choosing some finite horizon T_h and setting $\boldsymbol{u}_{\text{CAM}} = \Pi_{\text{CAM}}[\boldsymbol{x}(t_{\text{fail}})] = 0$ for all potential failure times $t_{\text{fail}} \in \mathcal{T}_{\text{fail}}$.

C. Safety in CWH Dynamics

We now specialize these ideas to proximity operations under impulsive CWH dynamics. Because many missions require stringent avoidance (before the final approach and docking phase, for example), it is quite common for a KOZ $\mathcal{X}_{\text{KOZ}}$, typically ellipsoidal in shape, to be defined about the target in the CWH frame. Throughout its approach, the chaser must certify that it will not enter this KOZ under any circumstance up to a specified thruster fault tolerance F, where here faults imply zero-output (stuck-off) thruster failures. Per Definition 4, this necessitates a search for a safe invariant set for finite-time escape along with, as outlined by Definition 5, the definition of an escape policy Π_{CAM}, which we describe next.

1. CAM Policy

We now have all the tools we need to formulate an active abort policy for spacecraft maneuvering under CWH dynamics. Recall from Definition 4 that for mission safety following a failure we must find a terminal state in an invariant set $\mathcal{X}_{\text{invariant}}$ entirely contained within the free state space $\mathcal{X}_{\text{free}}$. To that end, we choose for $\mathcal{X}_{\text{invariant}}$ the set of circularized orbits of which the planar projections lie outside of the radial band spanned by the KOZ. The reasons we choose this particular set for abort termination are threefold: circular orbits are 1) stable (assuming Keplerian motion, which is reasonable even under perturbations because the chaser and target are perturbed together and it is their relative state differences that matter), 2) accessible (given the proximity of the chaser to the target's circular orbit), and 3) passively safe (once reached, provided there is no intersection with the KOZ). In the planar case, this set of safe circularized orbits can fortunately be identified by inspection. As shown in Fig. 5, the set of orbital radii spanning the KOZ are excluded in order to prevent an eventual collision with the KOZ ellipsoid, either in the short term or after nearly one full synodic period. In the event of an unrecoverable failure or an abort scenario taking longer than one synodic period to resolve, circularization within this region would jeopardize the target, a violation of Definition 2. Such a region is called a zero-thrust region of inevitable collision (RIC), which we denote as $\mathcal{X}_{\text{ric}}$, as without additional intervention a collision with the KOZ is imminent and certain. To summarize this mathematically,

$$\mathcal{X}_{\text{KOZ}} = \{\boldsymbol{x} | \boldsymbol{x}^T \boldsymbol{E} \boldsymbol{x} \le 1\} \tag{8}$$

where $\boldsymbol{E} = \text{diag}(\rho_{\delta x}^{-2}, \rho_{\delta y}^{-2}, \rho_{\delta z}^{-2}, 0, 0, 0)$, with ρ_i representing the ellipsoidal KOZ semi-axis in the i-th LVLH frame axis direction,

$$\mathcal{X}_{\text{ric}} = \left\{ \boldsymbol{x} \middle| |\delta x| < \rho_{\delta x}, \delta \dot{x} = 0, \delta \dot{y} = -\frac{3}{2} n_{\text{ref}} \delta x \right\} \supset \mathcal{X}_{\text{KOZ}} \tag{9}$$

$$\mathcal{X}_{\text{invariant}} = \left\{ \boldsymbol{x} \middle| |\delta x| \ge \rho_{\delta x}, \delta \dot{x} = 0,\ \delta \dot{y} = -\frac{3}{2} n_{\text{ref}} \delta x \right\} = \mathcal{X}_{\text{ric}}^c \tag{10}$$

In short, our CAM policy to safely escape from a state $\boldsymbol{x}$ at which the spacecraft arrives (possibly under failures) at time t_{fail}, as visualized in Fig. 6, consists of the following:

1) Coast from $\boldsymbol{x}(t_{\text{fail}})$ to some new $T_h > t_{\text{fail}}$ such that $\boldsymbol{x}_{\text{CAM}}(T_h^-)$ lies at a position in $\mathcal{X}_{\text{invariant}}$.

2) Circularize the (in-plane) orbit at $\boldsymbol{x}_{\text{CAM}}(T_h)$ such that $\boldsymbol{x}_{\text{CAM}}(T_h^+) \in \mathcal{X}_{\text{invariant}}$.

3) Coast along the new orbit (horizontal drift along the in-track axis in the CWH relative frame) in $\mathcal{X}_{\text{invariant}}$ until allowed to continue the mission (e.g., after approval from ground operators).

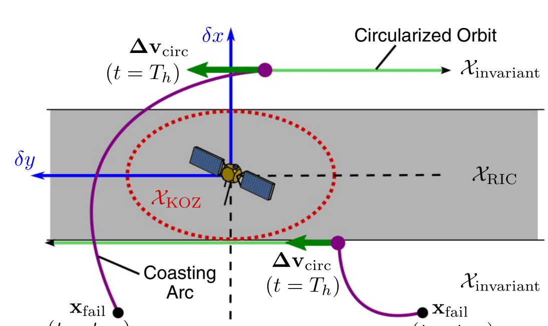


Fig. 6 Examples of safe abort CAMs $\boldsymbol{x}_{\text{CAM}}$ following failures.

2. Optimal CAM Circularization

In the event of a thruster failure at state $\boldsymbol{x}(t_{\text{fail}})$ that requires an emergency CAM, the time $T_h > t_{\text{fail}}$ at which to attempt a circularization maneuver after coasting from $\boldsymbol{x}(t_{\text{fail}})$ becomes a degree of freedom. As we intend to maximize the recovery chances of the chaser after a failure, we choose T_h so as to minimize the cost of the circularization burn $\boldsymbol{\Delta v}_{\text{circ}}$, the magnitude of which we denote Δv_{circ}. Details on the approach, which can be solved analytically, can be found in Appendix A.

3. CAM Policy Feasibility

Once the circularization time T_h is determined, feasibility of the escape trajectory under every possible failure configuration at $\boldsymbol{x}(t_{\text{fail}})$ must be assessed in order to declare a particular CAM as actively safe. To show this, the constraints of Eq. (7) must be evaluated under every combination of stuck-off thrusters (up to fault tolerance F), with the exception of KOZ avoidance as this is embedded into the CAM design process. How quickly this may be done depends on how many of these constraints may be considered static (unchanging, i.e., independent of t_{fail}, in the LVLH frame of reference) or time varying (otherwise).

Fortunately, most practical mission constraints are static (i.e., imposed in advance by mission planners), allowing CAM trajectory feasibility verification to be moved offline. For example, considering our particular constraints in Sec. II.E, if we can assume that the target remains enclosed within its KOZ near the origin and that it maintains a fixed attitude profile in the LVLH frame, then obstacle and antenna lobe avoidance constraints become time invariant (independent of the arrival time t_{fail}). If we further assume the attitude $\boldsymbol{q}(t)$ of the chaser is specified as a function of $\boldsymbol{x}(t)$, then control allocation feasibility and plume-impingement constraints become verifiable offline as well. Better still, because of their time independence, we need only to evaluate the safety of arriving at each failure state $\boldsymbol{x}_{\text{fail}}$ once; this means the active safety of a particular state can be cached, a fact we will make extensive use of in the design of our planning algorithm.

Some constraints, on the other hand, cannot be defined *a priori*. These must be evaluated online, once the time t_{fail} and current environment are known, which can be expensive due to the combinatorial explosion of thruster failure combinations. In such cases, these constraints can instead be conservatively approximated by equivalent static constraints, or otherwise omitted from online guidance until after a nominal guidance plan has been completely determined (called lazy evaluation). These strategies can help ensure active safety while maintaining real-time capability.

IV. Planning Algorithm and Theoretical Characterization

With the proximity operations scenario established, we are now in position to describe our approach. As previously described, the constraints that must be satisfied in Eq. (1) are diverse, complex, and difficult to satisfy numerically. In this section, we propose a guidance algorithm to solve this problem, followed by a detailed proof of its optimality with regard to the sum-of-2-norms propellant-cost metric J under impulsive CWH dynamics. As will be seen, the proof relies on an understanding of: 1) the steering connections between sampled points assuming no obstacles or other trajectory constraints and 2) the nearest-neighbors or reachable states from a given state. We hence start by characterizing these two concepts, in Secs. IV.A and IV.B, respectively. We then proceed to the algorithm presentation (Sec. IV. C) and its theoretical characterization (Sec. IV.D), before closing with a description of two smoothing techniques for rapidly reducing the costs of sampling-based solution trajectories for systems with impulsive actuation (Sec. V).

A. State Interconnections: Steering Problem

For sample-to-sample interconnections, we consider the unconstrained minimal-propellant 2PBVP, or steering problem, between an initial state $\boldsymbol{x}_0$ and a final state $\boldsymbol{x}_f$ under CWH dynamics. Solutions to these steering problems provide the local building blocks from which we construct solutions to the more complicated problem formulation in Eq. (1). Steering solutions serve two main purposes:

1) They represent a class of short-horizon controlled trajectories that are filtered online for constraint satisfaction and efficiently strung together into a state-space spanning graph (i.e., a tree or roadmap)

2) The costs of steering trajectories are used to inform the graph construction process by identifying the unconstrained nearest neighbors as edge candidates.

Because these problems can be expressed independently of the arrival time t_0 (as will be shown), our solution algorithm does not need to solve these problems online; the solutions between every pair of samples can be precomputed and stored before receiving a motion query. Hence, the 2PBVP presented here need not be solved quickly. However, we mention techniques for speedups due to the reliance of our smoothing algorithm (Algorithm 2) on a fast solution method.

Substituting our boundary conditions into Eq. (4), evaluating at $t = t_f$, and rearranging, we seek a stacked burn vector $\boldsymbol{\Delta V}$ such that

$$\boldsymbol{\Phi}_v(t_f, \{\tau_i\}_i)\boldsymbol{\Delta V} = \boldsymbol{x}_f - \boldsymbol{\Phi}(t_f, t_0)\boldsymbol{x}_0 \tag{11}$$

for some number N of burn times $\tau_i \in [t_0, t_f]$. Formulating this as an optimal control problem that minimizes our sum-of-2-norms cost functional (as a proxy for the actual propellant consumption, as described in Sec. II.A), we wish to solve

$$\begin{array}{lll}
\text{Given:} & \multicolumn{2}{l}{\text{initial state } \boldsymbol{x}_0, \text{final state } \boldsymbol{x}_f, \text{burn magnitude bound } \Delta v_{\max},} \\
& \multicolumn{2}{l}{\text{and maneuver duration bound } T_{\max}} \\
\underset{\Delta v_i, \tau_i, t_f, N}{\text{minimize}} & \displaystyle\sum_{i=1}^{N} \|\Delta \boldsymbol{v}_i\|_2 & \\
\text{subject to} & \boldsymbol{\Phi}_v(t_f, \{\tau_i\}_i)\Delta \boldsymbol{V} = \boldsymbol{x}_f - \Phi(t_f, t_0)\boldsymbol{x}_0 & \text{dynamics/boundary conditions} \\
& 0 \le t_f - t_0 \le T_{\max} & \text{maneuver duration bounds} \\
& t_0 \le \tau_i \le t_f \quad \text{for burns } i & \text{burn time bounds} \\
& \|\Delta \boldsymbol{v}_i\|_2 \le \Delta v_{\max} \quad \text{for burns } i & \text{burn magnitude bounds}
\end{array} \tag{12}$$

Notice that this is a relaxed version of the original problem presented as Eq. (1), with only its boundary conditions, dynamic constraints, and control norm bound. As it stands, because of the nonlinearity of the dynamics with respect to τ_i, t_f, and N, Eq. (12) is nonconvex and inherently difficult to solve. However, we can make the problem tractable if we make a few assumptions. Given that we plan to string many steering trajectories together to form our overall solution, let us ensure they represent the most primitive building blocks possible such that their concatenation will adequately represent any arbitrary trajectory. Set $N = 2$ (the smallest number of burns required to transfer between any pair of arbitrary states, as it makes $\boldsymbol{\Phi}_v(t_f, \{\tau_i\}_i)$ square), and select burn times $\tau_1 = t_0$ and $\tau_2 = t_f$ (which automatically satisfy our burn time bounds). This leaves $\boldsymbol{\Delta v}_1 \in \mathbb{R}^{d/2}$ (an intercept burn applied just after $\boldsymbol{x}_0$ at time t_0), $\boldsymbol{\Delta v}_2 \in \mathbb{R}^{d/2}$ (a rendezvous burn applied just before $\boldsymbol{x}_f$ at time t_f), and t_f as our only remaining decision variables. If we conduct a search for $t_f^* \in [t_0, t_0 + T_{\max}]$, the relaxed 2PBVP can now be solved iteratively as a relatively simple bounded one-dimensional nonlinear minimization problem, where at each iteration one computes

$$\boldsymbol{\Delta V}(t_f) = \boldsymbol{\Phi}_v^{-1}(t_f, \{t_0, t_f\})[\boldsymbol{x}_f - \Phi(t_f, t_0)\boldsymbol{x}_0]$$

where the argument t_f is shown for $\boldsymbol{\Delta V}$ to highlight its dependence. By uniqueness of the matrix inverse (provided $\boldsymbol{\Phi}_v^{-1}$ is nonsingular, discussed in what follows), we need only check that the resulting impulses $\boldsymbol{\Delta v}_i(t_f)$ satisfy the magnitude bound to declare the solution to an iteration feasible. Notice that, because $\boldsymbol{\Phi}$ and $\boldsymbol{\Phi}_v^{-1}$ depend only on the difference between t_f and t_0, we can equivalently search over maneuver durations $T = t_f - t_0 \in [0, T_{\max}]$ instead, solving the following relaxation of Eq. (12):

$$\begin{aligned}
\text{Given:}\ & \text{initial state}\ \boldsymbol{x}_0, \text{final state}\ \boldsymbol{x}_f, \text{burn magnitude bound}\ \Delta v_{\max} \\
& \text{and maneuver duration bound}\ T_{\max} < \frac{2\pi}{n_{\text{ref}}} \\
\underset{T\in[0,T_{\max}]}{\text{minimize}}\ & \sum_{i=1}^{2} \|\Delta \boldsymbol{v}_i\|_2 \\
\text{subject to}\ & \Delta \boldsymbol{V} = \boldsymbol{\Phi}_v^{-1}(T, \{0, T\})(\boldsymbol{x}_f - \Phi(T, 0)\boldsymbol{x}_0) \qquad \text{dynamics/boundary conditions} \\
& \|\Delta \boldsymbol{v}_i\|_2 \le \Delta v_{\max} \qquad \text{for burns}\ i \qquad \text{burn magnitude bounds}
\end{aligned} \tag{13}$$

This dependence on the maneuver duration T only (and not on the time t_0 at which we arrive at $\boldsymbol{x}_0$) turns out to be indispensable for precomputation, as it allows steering trajectories to be generated and stored offline. Observe, however, that our steering solution $\boldsymbol{\Delta V}^*$ requires $\boldsymbol{\Phi}_v$ to be invertible, i.e., that $(t_f - \tau_1) - (t_f - \tau_2) = t_f - t_0 = T$ avoids singular values (including zero, orbital period multiples, and other values longer than one period [38]), and we ensure this by enforcing $T_{\max}$ to be shorter than one period. To handle the remaining case of $T = 0$, note a solution exists if and only if $\boldsymbol{x}_0$ and $\boldsymbol{x}_f$ differ in velocity only; in such instances, we take the solution to be $\boldsymbol{\Delta v}_2^*$ set as this velocity difference (with $\boldsymbol{\Delta v}_1^* = 0$).

B. Neighborhoods: Cost Reachability Sets

Armed with a steering solution, we can now define and identify state neighborhoods. This idea is captured by the concept of reachability sets. In keeping with Eq. (13), since $\boldsymbol{\Delta V}^*$ depends only on the trajectory endpoints $\boldsymbol{x}_f$ and $\boldsymbol{x}_0$, we henceforth refer to the cost of a steering trajectory by the notation $J(\boldsymbol{x}_0, \boldsymbol{x}_f)$. We then define the forward reachability set from a given state $\boldsymbol{x}_0$ as follows.

Definition 6 (forward reachable set)*:* The forward reachable set $\mathcal{R}$ from state $\boldsymbol{x}_0$ is the set of all states $\boldsymbol{x}_f$ that can be reached from $\boldsymbol{x}_0$ with a cost $J(\boldsymbol{x}_0, \boldsymbol{x}_f)$ below a given cost threshold $\bar{J}$, i.e.,

$$\mathcal{R}(\boldsymbol{x}_0, \bar{J}) \triangleq \{\boldsymbol{x}_f \in \mathcal{X} | J(\boldsymbol{x}_0, \boldsymbol{x}_f) < \bar{J}\}$$

Recall from Eq. (13) in Sec. IV.A that the steering cost may be written as

$$J(\boldsymbol{x}_0, \boldsymbol{x}_f) = \|\boldsymbol{\Delta v}_1\| + \|\boldsymbol{\Delta v}_2\| = \|\boldsymbol{S}_1 \boldsymbol{\Delta V}\| + \|\boldsymbol{S}_2 \boldsymbol{\Delta V}\| \tag{14}$$

where $\boldsymbol{S}_1 = [\boldsymbol{I}_{d/2\times d/2}, 0_{d/2\times d/2}]$, $\boldsymbol{S}_2 = [0_{d/2\times d/2}, \boldsymbol{I}_{d/2\times d/2}]$, and $\boldsymbol{\Delta V}$ is given by

$$\boldsymbol{\Delta V}(\boldsymbol{x}_0, \boldsymbol{x}_f) = \begin{bmatrix} \boldsymbol{\Delta v}_1 \\ \boldsymbol{\Delta v}_2 \end{bmatrix} = \boldsymbol{\Phi}_v^{-1}(t_f, \{t_0, t_f\})[\boldsymbol{x}_f - \Phi(t_f, t_0)\boldsymbol{x}_0]$$

The cost function $J(\boldsymbol{x}_0, \boldsymbol{x}_f)$ is difficult to gain insight into directly; however, as we shall see, we can work with its bounds much more easily.

Lemma 1 (fuel burn cost bounds)*:* For the cost function in Eq. (14), the following bounds hold:

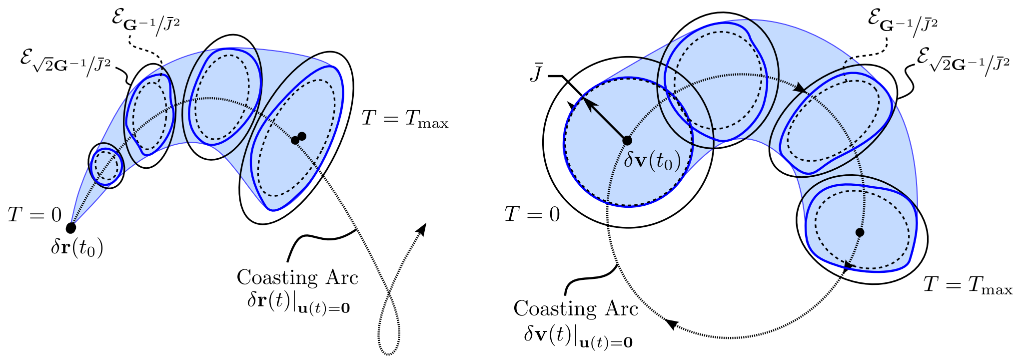
$$\|\boldsymbol{\Delta V}\| \le J(\boldsymbol{x}_0, \boldsymbol{x}_f) \le \sqrt{2}\|\boldsymbol{\Delta V}\|$$

Proof: For the upper bound, note that by the Cauchy–Schwarz inequality we have $J = \|\boldsymbol{\Delta v}_1\| \cdot 1 + \|\boldsymbol{\Delta v}_2\| \cdot 1 \le \sqrt{\|\boldsymbol{\Delta v}_1\|^2 + \|\boldsymbol{\Delta v}_2\|^2} \cdot \sqrt{1^2 + 1^2}$. That is, $J \le \sqrt{2}\|\boldsymbol{\Delta V}\|$. Similarly, for the lower bound, note that $J = \sqrt{(\|\boldsymbol{\Delta v}_1\| + \|\boldsymbol{\Delta v}_2\|)^2} \ge \sqrt{\|\boldsymbol{\Delta v}_1\|^2 + \|\boldsymbol{\Delta v}_2\|^2} = \|\boldsymbol{\Delta V}\|$. □

Now, observe that $\|\boldsymbol{\Delta V}\| = \sqrt{(\boldsymbol{x}_f - \boldsymbol{\Phi}(t_f, t_0)\boldsymbol{x}_0)^T \boldsymbol{G}^{-1} (\boldsymbol{x}_f - \boldsymbol{\Phi}(t_f, t_0)\boldsymbol{x}_0)}$, where $\boldsymbol{G}^{-1} = \boldsymbol{\Phi}_v^{-T}\boldsymbol{\Phi}_v^{-1}$. This is the expression for an ellipsoid $\mathcal{E}(\boldsymbol{x}_f)$ resolved in the LVLH frame with matrix $\boldsymbol{G}^{-1}$ and center $\boldsymbol{\Phi}(t_f, t_0)\boldsymbol{x}_0$ (the state $T = t_f - t_0$ time units ahead of $\boldsymbol{x}_0$ along its coasting arc). Combined with Lemma 1, we see that for a fixed maneuver time T and propellant cost threshold $\bar{J}$ the spacecraft at $\boldsymbol{x}_0$ can reach all states inside an area underapproximated by an ellipsoid with matrix $\boldsymbol{G}^{-1}/\bar{J}^2$ and overapproximated by an ellipsoid of matrix $\sqrt{2}\boldsymbol{G}^{-1}/\bar{J}^2$. The forward reachable set for impulsive CWH dynamics under our propellant-cost metric is therefore bounded by the union over all maneuver times of these under- and overapproximating ellipsoidal sets, respectively. To better visualize this, see Fig. 7 for a geometric interpretation.

C. Motion Planning: Modified FMT* Algorithm

We now have all the tools we need to adapt sampling-based motion planning algorithms to optimal vehicle guidance under impulsive CWH dynamics, as represented by Eq. (1). Sampling-based planning essentially breaks down a continuous trajectory optimization problem into a series of relaxed, local steering problems (as in Sec. IV.A) between intermediate waypoints (called samples) before piecing them together to form a global solution to the original problem. This framework can yield significant computational benefits if: 1) the relaxed subproblems are simple enough, and 2) the *a posteriori* evaluation of trajectory constraints is fast compared to a single solution of the full-scale problem. Furthermore, provided samples are sufficiently dense in the free state-space $\mathcal{X}_{\text{free}}$ and graph exploration is spatially symmetric, sampling-based planners can closely approximate global optima without fear of convergence to local minima. Although many candidate planners could be used here, we rely on the AO FMT* algorithm for its efficiency (see [24] for details on the advantages of FMT* over its state-of-the-art counterparts) and its compatibility with deterministic (as opposed to random) batch sampling [39], a key benefit that leads to a number of algorithmic simplifications (including the use of offline knowledge).



a) The set of reachable positions δr_f within duration T_{max} and propellant cost Δv_{max}

b) The set of reachable velocities δv_f within duration T_{max} and propellant cost Δv_{max}

Fig. 7 Bounds on reachability sets from initial state $x(t_0)$ under propellant-cost threshold $\bar{J}$.

The FMT* algorithm, tailored to our application, is presented as Algorithm 1 (we shall henceforth refer to our modified version of FMT* as simply FMT*, for brevity). Like its path-planning variant, our modified FMT* efficiently expands a tree of feasible trajectories from an initial state $\boldsymbol{x}_{\text{init}}$ to a goal state $\boldsymbol{x}_{\text{goal}}$ around nearby obstacles. It begins by taking a set of samples distributed in the free state space $\mathcal{X}_{\text{free}}$ using the SAMPLEFREE routine, which restricts state sampling to actively safe feasible states (which lie outside of $\mathcal{X}_{\text{obs}}$ and have access to a safe CAM as described in Sec. III.C). In our implementation, we assume samples are taken using the Halton sequence [40], though any deterministic, low-dispersion sampling sequence may be used [39]. After selecting $\boldsymbol{x}_{\text{init}}$ first for further expansion as the minimum cost-to-come node z, the algorithm then proceeds to look at reachable samples or neighbors (samples that can be reached with less than a given propellant cost threshold $\bar{J}$, as described in the previous subsection) and attempts to connect those with the cheapest cost-to-come back to the tree (using Steer). The cost threshold $\bar{J}$ is a free parameter of which the value can have a significant effect on performance; see Theorem 2 for a theoretical characterization and Sec. VI for a representative numerical trade study. Those trajectories satisfying the constraints of Eq. (1), as determined by CollisionFree, are saved. As feasible connections are made, the algorithm relies on adding and removing nodes (saved waypoint states) from three sets: a set of unexplored samples $\mathcal{V}_{\text{unvisited}}$ not yet connected to the tree, a frontier $\mathcal{V}_{\text{open}}$ of nodes likely to make efficient connections to unexplored neighbors, and an interior $\mathcal{V}_{\text{closed}}$ of nodes that are no longer useful for exploring the state space $\mathcal{X}$. More details on FMT* can be found in its original work [24].

To make FMT* amenable to a real-time implementation, we consider an online–offline approach that relegates as much computation as possible to a preprocessing phase. To be specific, the sample set $\mathcal{S}$ (Line 2), nearest-neighbor sets (used in Lines 5 and 6), and steering trajectory solutions (Line 7) may be entirely preprocessed, assuming the planning problem satisfies the following conditions:

1) The state space $\mathcal{X}$ is known *a priori*, as is typical for most LEO missions (note we do not impose this on the obstacle space $\mathcal{X}_{\text{obs}} \subset \mathcal{X}$, which must generally be identified online using onboard sensors upon arrival to $\mathcal{X}$),

2) Steering solutions are independent of sample arrival times t_0, as we show in Sec. IV.A.

Here, Item 1 allows samples to be precomputed, while Item 2 enables steering trajectories to be stored onboard or uplinked from the ground up to the spacecraft, since their values remain relevant regardless of the times at which the spacecraft actually follows them during the mission. This leaves only constraint checking, graph construction, and termination checks as parts of the online phase, greatly improving the online run time and leaving the more intensive work to offline resources for which running time is less important. This breakdown into online and offline components (inspired by [41]) is a valuable technique for imbuing kinodynamic motion planning problems with real-time online solvability using fast batch planners like FMT*.

D. Theoretical Characterization

It remains to show that FMT* provides similar asymptotic optimality guarantees under the sum-of-2-norms propellant-cost metric and impulsive CWH dynamics (which enter into Algorithm 1 under Lines 6 and 7), as has already been shown for kinematic (straight-line path-planning) problems [24]. For sampling-based algorithms, asymptotic optimality refers to the property that, as the number of samples $n \to \infty$, the cost of the trajectory (or path) returned by the planner approaches that of the optimal cost. Here, a

Algorithm 1 The fast marching tree algorithm (FMT*): Computes a minimal-cost trajectory from an initial state $x(t_0) = x_{\text{init}}$ to a target state x_{goal} through a fixed number n of samples $\mathcal{S}$

1) Add $\boldsymbol{x}_{\text{init}}$ to the root of the tree $\mathcal{T}$, as a member of the frontier set $\mathcal{V}_{\text{open}}$
2) Generate samples $\mathcal{S} \leftarrow$ SAMPLEFREE ($\mathcal{X}$, n, t_0), and add them to the unexplored set $\mathcal{V}_{\text{unvisited}}$
3) Set the minimum cost-to-come node in the frontier set as $z \leftarrow \boldsymbol{x}_{\text{init}}$
4) **while true do**
5) **for** each neighbor $\boldsymbol{x}$ of z in $\mathcal{V}_{\text{unvisited}}$ **do**
6)   Find the neighbor $\boldsymbol{x}_{\min}$ in $\mathcal{V}_{\text{open}}$ of cheapest cost-to-go to $\boldsymbol{x}$
7)   Compute the trajectory between them as $[\boldsymbol{x}(t), \boldsymbol{u}(t), t] \leftarrow \text{Steer}(\boldsymbol{x}_{\min}, \boldsymbol{x})$ (see Sec. IV.A)
8)   **if** COLLISIONFREE $[\boldsymbol{x}(t), \boldsymbol{u}(t), t]$, **then**
9)   Add the trajectory from $\boldsymbol{x}_{\min}$ to $\boldsymbol{x}$ to tree $\mathcal{T}$
10) Remove all $\boldsymbol{x}$ from the unexplored set $\mathcal{V}_{\text{unvisited}}$
11) Add any new connections $\boldsymbol{x}$ to the frontier $\mathcal{V}_{\text{open}}$
12) Remove z from the frontier $\mathcal{V}_{\text{open}}$, and add it to $\mathcal{V}_{\text{closed}}$
13) **if** $\mathcal{V}_{\text{open}}$ is empty, **then**
14)   **return** failure
15) Reassign z as the node in $\mathcal{V}_{\text{open}}$ with smallest cost-to-come from the root ($\boldsymbol{x}_{\text{init}}$)
16) **if** z is in the goal region $\mathcal{X}_{\text{goal}}$, **then**
17)   **return** success, and the unique trajectory from the root ($\boldsymbol{x}_{\text{init}}$) to z

proof is presented showing asymptotic optimality for the planning algorithm and problem setup used in this paper. We note that, while CWH dynamics are the primary focus of this work, the following proof methodology extends to any general linear system controlled by a finite sequence of impulsive actuations, the fixed-duration 2-impulse steering problem of which is uniquely determined (e.g., a wide array of second-order control systems).

The proof proceeds analogously to [24] by showing that it is always possible to construct an approximate path using points in $\mathcal{S}$ that closely follows the optimal path. Similarly to [24] are great, we will make use here of a concept called the ℓ_2 dispersion of a set of points, which places upper bounds on how far away a point in $\mathcal{X}$ can be from its nearest point in $\mathcal{S}$ as measured by the ℓ_2-norm.

Definition 7 (ℓ_2 dispersion)*:* For a finite, nonempty set $\mathcal{S}$ of points in a d-dimensional compact Euclidean subspace $\mathcal{X}$ with positive Lebesgue measure, its ℓ_2 dispersion $D(\mathcal{S})$ is defined as

$$\begin{aligned} D(\mathcal{S}) &\triangleq \sup_{\boldsymbol{x}\in\mathcal{X}} \min_{\boldsymbol{s}\in\mathcal{S}} \|\boldsymbol{s}-\boldsymbol{x}\| \\ &= \sup\{R>0 \mid \exists\, \boldsymbol{x}\in\mathcal{X} \text{ with } \mathcal{B}(\boldsymbol{x},R)\cap\mathcal{S}=\varnothing\} \end{aligned}$$

where $\mathcal{B}(\boldsymbol{x},R)$ is a Euclidean ball with radius R centered at state $\boldsymbol{x}$.

We also require a means for quantifying the deviation that small endpoint perturbations can bring about in the two-impulse steering control. This result is necessary to ensure that the particular placement of the points of $\mathcal{S}$ is immaterial; only its low-dispersion property matters.

Lemma 2 (steering with perturbed endpoints) *:* For a given steering trajectory $\boldsymbol{x}(t)$ with initial time t_0 and final time t_f, let $\boldsymbol{x}_0 := \boldsymbol{x}(t_0)$, $\boldsymbol{x}_f := \boldsymbol{x}(t_f)$, $T := t_f - t_0$, and $J := J(\boldsymbol{x}_0, \boldsymbol{x}_f)$. Consider now the perturbed steering trajectory $\tilde{\boldsymbol{x}}(t)$ between perturbed start and endpoints $\tilde{\boldsymbol{x}}_0 = \boldsymbol{x}_0 + \delta\boldsymbol{x}_0$ and $\tilde{\boldsymbol{x}}_f = \boldsymbol{x}_f + \delta\boldsymbol{x}_f$ and its corresponding cost $J(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{x}}_f)$.

Case 1 ($T = 0$): There exists a perturbation center $\delta\boldsymbol{x}_c$ (consisting of only a position shift) with $\|\delta\boldsymbol{x}_c\| = O(J^2)$ such that, if $\|\delta\boldsymbol{x}_0\| \le \eta J^3$ and $\|\delta\boldsymbol{x}_f - \delta\boldsymbol{x}_c\| \le \eta J^3$, then $J(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{x}}_f) \le J[1 + 4\eta + O(J)]$ and the spatial deviation of the perturbed trajectory $\tilde{\boldsymbol{x}}(t)$ from $\boldsymbol{x}(t)$ is $O(J)$.

Case 2 ($T > 0$): If $\|\delta\boldsymbol{x}_0\| \le \eta J^3$ and $\|\delta\boldsymbol{x}_f\| \le \eta J^3$, then $J(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{x}}_f) \le J[1 + O(\eta J^2 T^{-1})]$, and the spatial deviation of the perturbed trajectory $\tilde{\boldsymbol{x}}(t)$ from $\boldsymbol{x}(t)$ is $O(J)$.

Proof: For the proof, see Appendix B. □

We are now in a position to prove that the cost of the trajectory returned by FMT* approaches that of an optimal trajectory as the number of samples $n \to \infty$. The proof proceeds in two steps. First, we establish that there is a sequence of waypoints in $\mathcal{S}$ that are placed closely along the optimal path and approximately evenly spaced in cost. Then, we show that the existence of these waypoints guarantees that FMT* finds a path with a cost close to that of the optimal cost. The theorem and proof combine elements from Theorem 1 in [24] and Theorem IV.6 from [42].

Definition 8 (strong δ clearance)*:* A trajectory $\boldsymbol{x}(t)$ is said to have strong δ clearance if, for some $\delta > 0$ and all t, the Euclidean distance between $\boldsymbol{x}(t)$ and any point in $\mathcal{X}_{\text{obs}}$ is greater than δ.

Theorem 1 (existence of waypoints near an optimal path)*:* Let $\boldsymbol{x}(t)$ be a feasible trajectory for the motion planning problem Eq. (1) with strong δ clearance, let $\boldsymbol{u}^*(t) = \sum_{i=1}^{N} \boldsymbol{\Delta v}_i^* \cdot \delta(t - \tau_i^*)$ be its associated control trajectory, and let J^* be its cost. Furthermore, let $\mathcal{S} \cup \{\boldsymbol{x}_{\text{init}}\}$ be a set of $n \in \mathbb{N}$ points from $\mathcal{X}_{\text{free}}$ with dispersion $D(\mathcal{S}) \le \gamma n^{-1/d}$. Let $\epsilon > 0$, and choose $\bar{J} = 4(\gamma n^{-1/d}/\epsilon)^{1/3}$. Then, provided that n is sufficiently large, there exists a sequence of points $\{\boldsymbol{y}_k\}_{k=0}^{K}$, $\boldsymbol{y}_k \in \mathcal{S}$ such that $J(\boldsymbol{y}_k, \boldsymbol{y}_{k+1}) \le \bar{J}$, the cost of the path $\boldsymbol{y}(t)$ made by joining all of the steering trajectories between $\boldsymbol{y}_k$ and $\boldsymbol{y}_{k+1}$ is $\sum_{k=0}^{K-1} J(\boldsymbol{y}_k, \boldsymbol{y}_{k+1}) \le (1+\epsilon)J^*$, and $\boldsymbol{y}(t)$ is itself strong $(\delta/2)$ clear.

Proof: We first note that, if $J^* = 0$, then we can pick $\boldsymbol{y}_0 = \boldsymbol{x}^*(t_0)$ and $\boldsymbol{y}_1 = \boldsymbol{x}^*(t_f)$ as the only points in $\{\boldsymbol{y}_k\}$, and the result is trivial. Thus, assume that $J^* > 0$. Construct a sequence of times $\{t_k\}_{k=0}^{K}$ and corresponding points $\boldsymbol{x}_k^* = \boldsymbol{x}^*(t_k)$ spaced along $\boldsymbol{x}^*(t)$ in cost intervals of $\bar{J}/2$. We admit a slight abuse of notation here in that $\boldsymbol{x}^*(\tau_i^*)$ may represent a state with any velocity along the length of the impulse $\boldsymbol{\Delta v}_i^*$; to be precise, pick $\boldsymbol{x}_0^* = \boldsymbol{x}_{\text{init}}$, $t_0 = 0$, and for $k = 1, 2, \ldots$ define $j_k = \min\{j \mid \sum_{i=1}^{j} \|\boldsymbol{\Delta v}_i^*\| > k\frac{\bar{J}}{2}\}$, and select t_k and $\boldsymbol{x}_k^*$ as

$$\begin{aligned} t_k &= \tau_{j_k}^* \\ \boldsymbol{x}_k^* &= \lim_{t\to t_k^-} \boldsymbol{x}^*(t) + \left(k\frac{\bar{J}}{2} - \sum_{i=1}^{j_k-1} \|\boldsymbol{\Delta v}_i^*\|\right) \boldsymbol{B} \frac{\boldsymbol{\Delta v}_i^*}{\|\boldsymbol{\Delta v}_i^*\|} \end{aligned}$$

Let $K = \lceil J^* \rceil / (\bar{J}/2)$, and set $t_K = t_f$, $\boldsymbol{x}_K^* = \boldsymbol{x}^*(t_f)$. Since the trajectory $\boldsymbol{x}^*(t)$ to be approximated is fixed, for sufficiently small $\bar{J}$ (equivalently, sufficiently large n), we may ensure that the control applied between each $\boldsymbol{x}_k^*$ and $\boldsymbol{x}_{k+1}^*$ occurs only at the endpoints. In particular this may be accomplished by choosing n large enough so that $\bar{J} < \min_i \|\boldsymbol{\Delta v}_i^*\|$. In the limit $\bar{J} \to 0$, the vast majority of the two-impulse steering connections between successive $\boldsymbol{x}_k^*$ will be zero-time maneuvers (arranged along the length of each burn $\boldsymbol{\Delta v}_i^*$) with only N positive-time maneuvers spanning the regions of $\boldsymbol{x}^*(t)$ between burns. By considering this regime of n, we note that applying two-impulse steering between successive $\boldsymbol{x}_k^*$ (which otherwise may only approximate the performance of a more complex control scheme) requires cost no greater than that of $\boldsymbol{x}^*$ itself along that step, i.e., $\bar{J}/2$.

We now inductively define a sequence of points $\{\hat{\boldsymbol{x}}_k^*\}_{k=0}^{K}$ by $\hat{\boldsymbol{x}}_0^* = \boldsymbol{x}_0^*$ and for each $k > 0$:

1) If $t_k = t_{k-1}$, pick $\hat{\boldsymbol{x}}_k^* = \boldsymbol{x}_k^* + \delta\boldsymbol{x}_{c,k} + (\hat{\boldsymbol{x}}_{k-1}^* - \boldsymbol{x}_{k-1}^*)$, where $\delta\boldsymbol{x}_{c,k}$ comes from Lemma 2 for zero-time approximate steering between $\boldsymbol{x}_{k-1}^*$ and $\boldsymbol{x}_k^*$ subject to perturbations of size ϵJ^3.

2) Otherwise, if $t_k > t_{k-1}$, pick $\hat{\boldsymbol{x}}_k^* = \boldsymbol{x}_k^* + (\hat{\boldsymbol{x}}_{k-1}^* - \boldsymbol{x}_{k-1}^*)$. The reason for defining these $\hat{\boldsymbol{x}}_k^*$ is that the process of approximating each $\boldsymbol{\Delta v}_i^*$ by a sequence of small burns necessarily incurs some short-term position drift. Since $\delta\boldsymbol{x}_{c,k} = O(\bar{J}^2)$ for each k, and since $K = O(\bar{J}^{-1})$, the maximum accumulated difference satisfies $\max_k \|\hat{\boldsymbol{x}}_k^* - \boldsymbol{x}_k^*\| = O(\bar{J})$.

For each k, consider the Euclidean ball centered at $\hat{\boldsymbol{x}}_k^*$ with radius $\gamma n^{-\frac{1}{d}}$; i.e., let $\mathcal{B}_k := \mathcal{B}(\hat{\boldsymbol{x}}_k^*, \gamma n^{-\frac{1}{d}})$. By Definition 7 and our restriction on $\mathcal{S}$, each $\mathcal{B}_k$ contains at least one point from $\mathcal{S}$. Hence, for every $\mathcal{B}_k$, we can pick a waypoint $\boldsymbol{y}_k$ such that $\boldsymbol{y}_k \in \mathcal{B}_k \cap \mathcal{S}$. Then, $\|\boldsymbol{y}_k - \hat{\boldsymbol{x}}_k^*\| \le \gamma n^{-\frac{1}{d}} = \epsilon(\bar{J}/2)^3/8$ for all k, and thus by Lemma 2 (with $\eta = \epsilon/8$), we have that

$$J(\boldsymbol{y}_k, \boldsymbol{y}_{k+1}) \le \frac{\bar{J}}{2}\left[1 + \frac{\epsilon}{2} + O(\bar{J})\right] \le \frac{\bar{J}}{2}(1+\epsilon)$$

for sufficiently large n. In applying Lemma 2 to Case 2 for k such that $t_{k+1} > t_k$, we note that the T^{-1} term is mitigated by the fact that there is only a finite number of burn times τ_i^* along $\boldsymbol{x}^*(t)$. Thus, for each such k, $t_{k+1} - t_k \ge \min_j(t_{j+1} - t_j) > 0$, so in every case, we have $J(\boldsymbol{y}_k, \boldsymbol{y}_{k+1}) \le (\bar{J}/2)(1+\epsilon)$. That is, each segment connecting $\boldsymbol{y}_k$ to $\boldsymbol{y}_{k+1}$ approximates the cost of the corresponding $\boldsymbol{x}_k^*$ to $\boldsymbol{x}_{k+1}^*$ segment of $\boldsymbol{x}^*(t)$ up to a multiplicative factor of ϵ, and thus

$$\sum_{k=0}^{K-1} J(\boldsymbol{y}_k, \boldsymbol{y}_{k+1}) \le (1+\epsilon)J^*$$

Finally, to establish that $\boldsymbol{y}(t)$, the trajectory formed by steering through the $\boldsymbol{y}_k$ in succession, has sufficient obstacle clearance, we note that its distance from $\boldsymbol{x}^*(t)$ is bounded by $\max_k \|\hat{\boldsymbol{x}}_k^* - \boldsymbol{x}_k^*\| = O(\bar{J})$ plus the deviation bound from Definition 7, again $O(\bar{J})$. For sufficiently large n, the total distance $O(\bar{J})$ will be bounded by $\delta/2$, and thus $\boldsymbol{y}(t)$ will have strong $(\delta/2)$ clearance. □

We now prove that FMT* is asymptotically optimal in the number of points n, provided the conditions required in Theorem 1 hold; note the proof is heavily based on Theorem VI.1 from [43].

Theorem 2 (asymptotic performance of FMT*)*:* Let $\boldsymbol{x}^*(t)$ be a feasible trajectory satisfying Eq. (1) with strong δ clearance and cost J^*. Let $\mathcal{S} \cup \{\boldsymbol{x}_0\}$ be a set of $n \in \mathbb{N}$ samples from $\mathcal{X}_{\text{free}}$ with dispersion $D(\mathcal{S}) \le \gamma n^{-1/d}$. Finally, let J_n denote the cost of the path returned by FMT* with n points in $\mathcal{S}$ while using a cost threshold $\bar{J}(n) = \omega(n^{-1/3d})$ and $\bar{J} = o(1)$. [That is, $\bar{J}(n)$ asymptotically dominates

$n^{-1/3d}$ and is asymptotically dominated by 1.] Then, $\lim_{n\to\infty} J_n \le J^*$.

Proof: Let $\epsilon > 0$. Pick n sufficiently large so that $\delta/2 \ge \bar{J} \ge 4(\gamma n^{-1/d}/\epsilon)^{1/3}$ such that Theorem 1 holds. That is, there exists a sequence of waypoints $\{\boldsymbol{y}_k\}_{k=0}^{K}$ approximating $\boldsymbol{x}^*(t)$ such that the trajectory $\boldsymbol{y}(t)$ created by sequentially steering through the $\boldsymbol{y}_k$ is strong $\delta/2$ clear, the connection costs of which satisfy $J(\boldsymbol{y}_k, \boldsymbol{y}_{k+1}) \le \bar{J}$ and the total cost of which satisfies $\sum_{k=0}^{K-1} J(\boldsymbol{y}_k, \boldsymbol{y}_{k+1}) \le (1+\epsilon)J^*$. We show that FMT* recovers a path with cost at least as good as $\boldsymbol{y}(t)$; that is, we show that $\lim_{n\to\infty} J_n \le J^*$.

Consider running FMT* to completion, and for each $\boldsymbol{y}_k$, let $c(\boldsymbol{y}_k)$ denote the cost-to-come of $\boldsymbol{y}_k$ in the generated graph (with value ∞ if $\boldsymbol{y}_k$ is not connected). We show by induction that

$$\min[c(\boldsymbol{y}_m), J_n] \le \sum_{k=0}^{m-1} J(\boldsymbol{y}_k, \boldsymbol{y}_{k+1}) \tag{15}$$

for all $m \in [1, \ldots, K]$. For the base case $m = 1$, we note by the initialization of FMT* in Line 1 of Algorithm 1 that $\boldsymbol{x}_{\text{init}}$ is in $\mathcal{V}_{\text{open}}$; therefore, by the design of FMT* (per Lines 5–9), every possible feasible connection is made between the first waypoint $\boldsymbol{y}_0 = \boldsymbol{x}_{\text{init}}$ and its neighbors. Since $J(\boldsymbol{y}_0, \boldsymbol{y}_1) \le \bar{J}$ and the edge $(\boldsymbol{y}_0, \boldsymbol{y}_1)$ is collision free, it is also in the FMT* graph. Then, $c(\boldsymbol{y}_1) = J(\boldsymbol{y}_0, \boldsymbol{y}_1)$. Now assuming that Eq. (15) holds for $m-1$, one of the following statements holds:

1) $J_n \le \sum_{k=0}^{m-2} J(\boldsymbol{y}_k, \boldsymbol{y}_{k+1})$.

2) $c(\boldsymbol{y}_{m-1}) \le \sum_{k=0}^{m-2} J(\boldsymbol{y}_k, \boldsymbol{y}_{k+1})$, and FMT* ends before considering $\boldsymbol{y}_m$.

3) $c(\boldsymbol{y}_{m-1}) \le \sum_{k=0}^{m-2} J(\boldsymbol{y}_k, \boldsymbol{y}_{k+1})$, and $\boldsymbol{y}_{m-1} \in \mathcal{V}_{\text{open}}$ when $\boldsymbol{y}_m$ is first considered.

4) $c(\boldsymbol{y}_{m-1}) \le \sum_{k=0}^{m-2} J(\boldsymbol{y}_k, \boldsymbol{y}_{k+1})$, and $\boldsymbol{y}_{m-1} \notin \mathcal{V}_{\text{open}}$ when $\boldsymbol{y}_m$ is first considered.

We now show for each case that our inductive hypothesis holds.

Case 1: $J_n \le \sum_{k=0}^{m-2} J(\boldsymbol{y}_k, \boldsymbol{y}_{k+1}) \le \sum_{k=0}^{m-1} J(\boldsymbol{y}_k, \boldsymbol{y}_{k+1})$.

Case 2: Since at every step FMT* considers the node that is the endpoint of the path with the lowest cost, if FMT* ends before considering $\boldsymbol{y}_m$, we have

$$J_n \le c(\boldsymbol{y}_m) \le c(\boldsymbol{y}_{m-1}) + J(\boldsymbol{y}_{m-1}, \boldsymbol{y}_m) \le \sum_{k=0}^{m-1} J(\boldsymbol{y}_k, \boldsymbol{y}_{k+1})$$

Case 3: Since the neighborhood of $\boldsymbol{y}_m$ is collision free by the clearance property of $\boldsymbol{y}$, and since $\boldsymbol{y}_{m-1}$ is a possible parent candidate for connection, $\boldsymbol{y}_m$ will be added to the FMT* tree as soon as it is considered with $c(\boldsymbol{y}_m) \le c(\boldsymbol{y}_{m-1}) + J(\boldsymbol{y}_{m-1}, \boldsymbol{y}_m) \le \sum_{k=0}^{m-1} J(\boldsymbol{y}_k, \boldsymbol{y}_{k+1})$.

Case 4: When $\boldsymbol{y}_m$ is considered, it means there is a node $z \in \mathcal{V}_{\text{open}}$ (with minimum cost to come through the FMT* tree) and $\boldsymbol{y}_m \in \mathcal{R}(z, \bar{J})$. Then, $c(\boldsymbol{y}_m) \le c(z) + J(z, \boldsymbol{y}_m)$. Since $c(\boldsymbol{y}_{m-1}) < \infty$, $\boldsymbol{y}_{m-1}$ must be added to the tree by the time FMT* terminates. Consider the path from $\boldsymbol{x}_{\text{init}}$ to $\boldsymbol{y}_{m-1}$ in the final FMT* tree, and let $\boldsymbol{w}$ be the last vertex along this path, which is in $\mathcal{V}_{\text{open}}$ at the time when $\boldsymbol{y}_m$ is considered. If $\boldsymbol{y}_m \in \mathcal{R}(\boldsymbol{w}, \bar{J})$, i.e., $\boldsymbol{w}$ is a parent candidate for connection, then

$$\begin{aligned} c(\boldsymbol{y}_m) &\le c(\boldsymbol{w}) + J(\boldsymbol{w}, \boldsymbol{y}_m) \\ &\le c(\boldsymbol{w}) + J(\boldsymbol{w}, \boldsymbol{y}_{m-1}) + J(\boldsymbol{y}_{m-1}, \boldsymbol{y}_m) \\ &\le c(\boldsymbol{y}_{m-1}) + J(\boldsymbol{y}_{m-1}, \boldsymbol{y}_m) \\ &\le \sum_{k=0}^{m-1} J(\boldsymbol{y}_k, \boldsymbol{y}_{k+1}) \end{aligned}$$

Otherwise, if $\boldsymbol{y}_m \notin \mathcal{R}(\boldsymbol{w}, \bar{J})$, then $J(\boldsymbol{w}, \boldsymbol{y}_m) > \bar{J}$, and

$$\begin{aligned} c(\boldsymbol{y}_m) &\le c(z) + J(z, \boldsymbol{y}_m) \\ &\le c(\boldsymbol{w}) + \bar{J} \\ &\le c(\boldsymbol{w}) + J(\boldsymbol{w}, \boldsymbol{y}_m) \\ &\le c(\boldsymbol{w}) + J(\boldsymbol{w}, \boldsymbol{y}_{m-1}) + J(\boldsymbol{y}_{m-1}, \boldsymbol{y}_m) \\ &\le c(\boldsymbol{y}_{m-1}) + J(\boldsymbol{y}_{m-1}, \boldsymbol{y}_m) \\ &\le \sum_{k=0}^{m-1} J(\boldsymbol{y}_k, \boldsymbol{y}_{k+1}) \end{aligned}$$

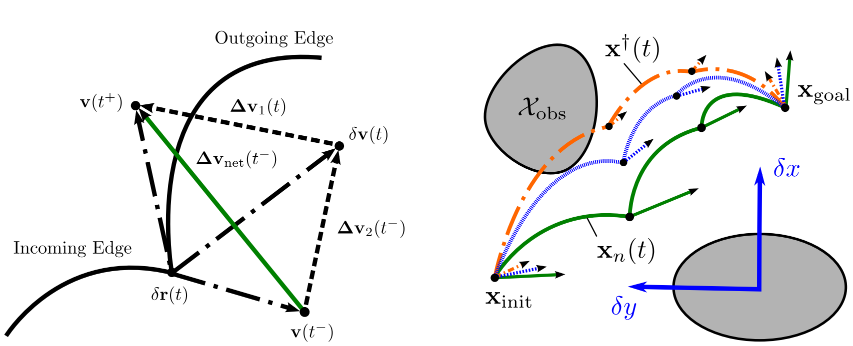
where we used the fact that $\boldsymbol{w}$ is on the path of $\boldsymbol{y}_{m-1}$ to establish $c(\boldsymbol{w}) + J(\boldsymbol{w}, \boldsymbol{y}_{m-1}) \le c(\boldsymbol{y}_{m-1})$. Thus, by induction, Eq. (15) holds for all m. Taking $m = K$, we finally have that $J_n \le c(\boldsymbol{y}_K) \le \sum_{k=0}^{K-1} J(\boldsymbol{y}_k, \boldsymbol{y}_{k+1}) \le (1+\epsilon)J^*$, as desired.

Remark 1: (asymptotic optimality of FMT*)*:* If the planning problem at hand admits an optimal solution that does not itself have strong δ clearance, but is arbitrarily approximable both pointwise and in cost by trajectories with strong clearance (see [43] for additional discussion on why such an assumption is reasonable), then Theorem 2 implies the asymptotic optimality of FMT*.

V. Trajectory Smoothing

Because of the discreteness caused by using a finite number of samples, sampling-based solutions will necessarily be approximations to true optima. In an effort to compensate for this limitation, we offer in this section two techniques to improve the quality of solutions returned by our planner from Sec. IV.C. We first describe a straightforward method for reducing the sum of $\boldsymbol{\Delta v}$-vector magnitudes along concatenated sequences of edge trajectories that can also be used to improve the search for propellant-efficient trajectories in the feasible state space $\mathcal{X}_{\text{free}}$. We then follow with a fast postprocessing algorithm for further reducing the propellant cost after a solution has been reported.

The first technique removes unnecessary $\boldsymbol{\Delta v}$ vectors that occur when joining subtrajectories (edges) in the planning graph. Consider merging two edges at a node with position $\delta\boldsymbol{r}(t)$ and velocity $\delta\boldsymbol{v}(t)$ as in Fig. 8a. A naive concatenation would retain both $\boldsymbol{\Delta v}_2(t^-)$ (the rendezvous burn added to the incoming velocity $\boldsymbol{v}(t^-)$) and $\boldsymbol{\Delta v}_1(t)$



a) Smoothing during graph construction (merges Δv vectors at edge endpoints)

b) Smoothing during postprocessing (see algorithm 2)

Fig. 8 Improving sampling-based solutions under minimal-propellant impulsive dynamics.

(the intercept burn used to achieve the outgoing velocity $\boldsymbol{v}(t^+)$) individually within the combined control trajectory. Yet, because these impulses occur at the same time, a more realistic approach should merge them into a single net $\Delta\boldsymbol{v}$ vector $\boldsymbol{\Delta v}_i(t^-)$. By the triangle inequality, we have that

$$\|\boldsymbol{\Delta v}_{\mathrm{net}}(t^-)\| = \|\boldsymbol{\Delta v}_2(t^-) + \boldsymbol{\Delta v}_1(t)\| \le \|\boldsymbol{\Delta v}_2(t^-)\| + \|\boldsymbol{\Delta v}_1(t)\|$$

Hence, merging edges in this way guarantees $\Delta\boldsymbol{v}$ savings for solution trajectories under our propellant-cost metric. Furthermore, incorporating net $\Delta\boldsymbol{v}$ into the cost to come during graph construction can make the exploration of the search space more efficient; the cost to come $c(z)$ for a given node z would then reflect the cost to rendezvous with z from $\boldsymbol{x}_{\mathrm{init}}$ through a series of intermediate intercepts rather than a series of rendezvous maneuvers (as a trajectory designer might normally expect). Note, on the other hand, that these new net $\Delta\boldsymbol{v}$ vectors may be larger than either individual burn, which may violate control constraints; control feasibility tests (allocation feasibility to thrusters, plume impingement, *etc.*) must thus be reevaluated for each new impulse. Furthermore, observe that the node velocity $\delta\boldsymbol{v}(t)$ is skipped altogether at edge endpoints when two edges as in Fig. 8a are merged in this fashion. This can be problematic for the actively safe abort CAM (see Sec. III.C) from states along the incoming edge, which relies on rendezvousing with the endpoint $\boldsymbol{x} = [\delta\boldsymbol{r}(t), \delta\boldsymbol{v}(t)]$ exactly before executing a one-burn circularization maneuver. To compensate for this, care must be taken to ensure that the burn $\boldsymbol{\Delta v}_2(t^-)$ that is eliminated during merging is appropriately appended to the front of the escape control trajectory and verified for all possible failure configurations. Hence, we see the price of smoothing in this way is 1) reevaluating control-dependent constraints at edge endpoints before accepting smoothing and 2) that our original one-burn policy now requires an extra burn, which may not be desirable in some applications.

The second technique attempts to reduce the solution cost by adjusting the magnitudes of $\Delta\boldsymbol{v}$ vectors in the trajectory returned by FMT* [denoted by $\boldsymbol{x}_n(t)$ with associated stacked impulse vector $\boldsymbol{\Delta V}_n$]. By relaxing FMT*'s constraint to pass through state samples, strong cost improvements may be gained. The main idea is to deform our low-cost, feasible solution $\boldsymbol{x}_n(t)$ as much as possible toward the unconstrained minimum-propellant solution $\boldsymbol{x}^*(t)$ between $\boldsymbol{x}_{\mathrm{init}}$ and $\boldsymbol{x}_{\mathrm{goal}}$, as determined by the 2PBVP [Eq. (12)] solution from Sec. IV.A [in other words, use a homotopic transformation from $\boldsymbol{x}_n(t)$ to $\boldsymbol{x}^*(t)$]. However, a naive attempt to solve Eq. (12) in its full generality would be too time consuming to be useful and would threaten the real-time capability of our approach. Assuming our sampling-based trajectory is near optimal (or at least in a low-cost solution homotopy), we can relax Eq. (12) by keeping the number of burns N, end time $t_f := t_{\mathrm{final}}$, and burn times τ_i fixed from our planning solution and solve for an approximate unconstrained minimum-propellant solution $\boldsymbol{\Delta V}^\dagger$ with associated state trajectory $\boldsymbol{x}^\dagger(t)$ via

$$\begin{aligned}
&\underset{\Delta v_i}{\text{minimize}} \quad \sum_{i=1}^{N} \|\Delta\boldsymbol{v}_i\|_2 \\
&\text{subject to } \boldsymbol{\Phi}_v(t_{\mathrm{final}}, \{\tau_i\}_i)\boldsymbol{\Delta V} = \boldsymbol{x}_{\mathrm{goal}} - \Phi(t_{\mathrm{final}}, t_{\mathrm{init}})\boldsymbol{x}_{\mathrm{init}} \\
&\qquad\qquad \text{dynamics/boundary conditions} \\
&\qquad\qquad \|\Delta\boldsymbol{v}_i\|_2 \le \Delta v_{\max} \quad \text{for all burns } i \\
&\qquad\qquad \text{burn magnitude bounds}
\end{aligned} \tag{16}$$

(see Sec. II.C for definitions). It can be shown that Eq. (16) is a second-order cone program and is hence quickly solved using standard convex solvers. As the following theorem shows explicitly, we can safely deform the trajectory $\boldsymbol{x}_n(t)$ toward $\boldsymbol{x}^\dagger(t)$ without violating our dynamics and boundary conditions if we use a convex combination of our two control trajectories $\boldsymbol{\Delta V}_n$ and $\boldsymbol{\Delta V}^\dagger$. This follows from the principle of superposition, given that the CWH equations are LTI and the fact that both solutions already satisfy the boundary conditions.

Theorem 3 (dynamic feasibility of CWH trajectory smoothing)*:* Suppose $\boldsymbol{x}_n(t)$ and $\boldsymbol{x}^\dagger(t)$ with respective control vectors $\boldsymbol{\Delta V}_n$ and $\boldsymbol{\Delta V}^\dagger$ are two state trajectories satisfying the steering problem Eq. (11) between states $\boldsymbol{x}_{\mathrm{init}}$ and $\boldsymbol{x}_{\mathrm{goal}}$. Then, the trajectory $\boldsymbol{x}(t)$ generated by the convex combination of $\boldsymbol{\Delta V}_n$ and $\boldsymbol{\Delta V}^\dagger$ is itself a convex combination of $\boldsymbol{x}_n(t)$ and $\boldsymbol{x}^\dagger(t)$ and hence also satisfies Eq. (11).

Proof: Let $\boldsymbol{\Delta V} = \alpha\boldsymbol{\Delta V}_n + (1-\alpha)\boldsymbol{\Delta V}^\dagger$ for some value $\alpha \in [0, 1]$. From our dynamics equation,

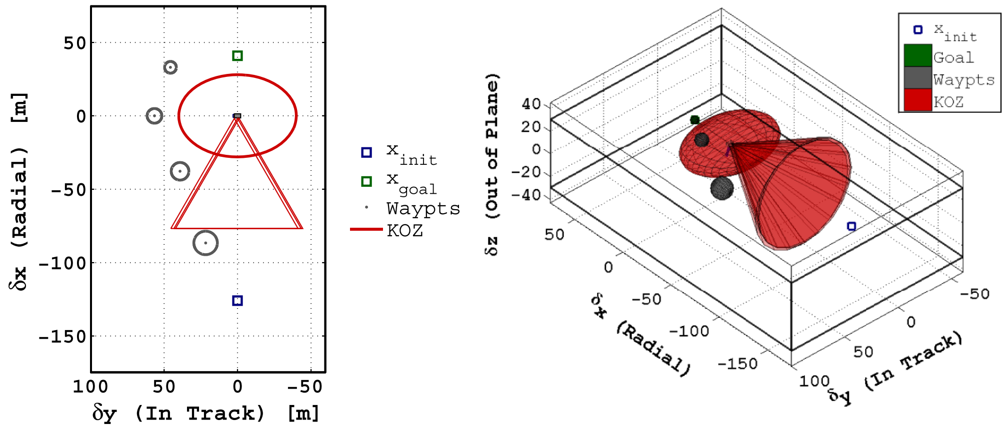
$$\begin{aligned}
\boldsymbol{x}(t) &= \boldsymbol{\Phi}(t, t_{\mathrm{init}})\boldsymbol{x}_{\mathrm{init}} + \boldsymbol{\Phi}_v(t, \{\tau_i\}_i)\boldsymbol{\Delta V} \\
&= [\alpha + (1-\alpha)]\Phi(t, t_{\mathrm{init}})\boldsymbol{x}_{\mathrm{init}} + \boldsymbol{\Phi}_v(t, \{\tau_i\}_i)[\alpha\boldsymbol{\Delta V}_n + (1-\alpha)\Delta\boldsymbol{V}^\dagger] \\
&= \alpha[\Phi(t, t_{\mathrm{init}})\boldsymbol{x}_{\mathrm{init}} + \boldsymbol{\Phi}_v(t, \{\tau_i\}_i)\Delta\boldsymbol{V}_n] \\
&\quad + (1-\alpha)[\Phi(t, t_{\mathrm{init}})\boldsymbol{x}_0 + \boldsymbol{\Phi}_v(t, \{\tau_i\}_i)\Delta\boldsymbol{V}^\dagger] \\
&= \alpha\boldsymbol{x}_n(t) + (1-\alpha)\boldsymbol{x}^\dagger(t)
\end{aligned}$$

which is a convex combination, as required. Substituting $t = t_{\mathrm{init}}$ or $t = t_{\mathrm{goal}}$, we see that $\boldsymbol{x}(t)$ satisfies the boundary conditions given that $\boldsymbol{x}_n(t)$ and $\boldsymbol{x}^\dagger(t)$ do.

We take advantage of this fact for trajectory smoothing. Our algorithm, reported as Algorithm 2 and illustrated in Fig. 8b, computes the approximate unconstrained minimum- propellant solution $\boldsymbol{x}^\dagger(t)$ and returns it (if feasible) or otherwise conducts a bisection line search on α, returning a convex combination of our original planning solution $\boldsymbol{x}_n(t)$ and $\boldsymbol{x}^\dagger(t)$ that comes as close to $\boldsymbol{x}^\dagger(t)$

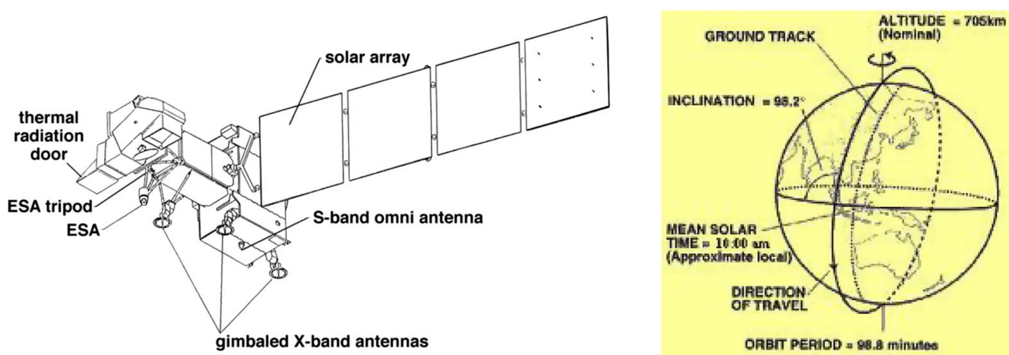
Algorithm 2 Trajectory smoothing algorithm for impulsive CWH dynamics: Given a trajectory $\boldsymbol{x}_n(t)$, $t \in [t_{\mathrm{init}}, t_{\mathrm{goal}}]$ between initial and goal states $\boldsymbol{x}_{\mathrm{init}}$ and $\boldsymbol{x}_{\mathrm{goal}}$ satisfying Eq. (1) with impulses $\boldsymbol{\Delta V}_n$ applied at times $\{\tau_i\}_i$, returns another feasible trajectory with reduced propellant cost

Initialize the smoothed trajectory $\boldsymbol{x}_{\mathrm{smooth}}(t)$ as $\boldsymbol{x}_n(t)$, with $\boldsymbol{\Delta V}_{\mathrm{smooth}} = \boldsymbol{\Delta V}_n$
Compute the unconstrained optimal control vector $\boldsymbol{\Delta V}^\dagger$ by solving Eq. (16)
Compute the unconstrained optimal state trajectory $\boldsymbol{x}^\dagger(t)$ using Eq. (4) *(See Sec. II.C)*
Initialize weight α and its lower and upper bounds as $\alpha \leftarrow 1$, $\alpha_\ell \leftarrow 0$, $\alpha_u \leftarrow 1$
while true **do**
  $\boldsymbol{x}(t) \leftarrow (1-\alpha)\boldsymbol{x}_n(t) + \alpha\boldsymbol{x}^\dagger(t)$
  $\boldsymbol{\Delta V} \leftarrow (1-\alpha)\boldsymbol{\Delta V}_n + \alpha\boldsymbol{\Delta V}^\dagger$
  if COLLISIONFREE $(\boldsymbol{x}(t), \boldsymbol{\Delta V}, t)$ **then**
    $\alpha_\ell \leftarrow \alpha$
    Save the smoothed trajectory $\boldsymbol{x}_{\mathrm{smooth}}(t)$ as $\boldsymbol{x}(t)$ and control $\boldsymbol{\Delta V}_{\mathrm{smooth}}$ as $\boldsymbol{\Delta V}$
  else
    $\alpha_u \leftarrow \alpha$
  if $\alpha_u - \alpha_\ell$ is less than tolerance $\delta\alpha_{\min} \in (0, 1)$ **then**
    break
  $\alpha \leftarrow (\alpha_\ell + \alpha_u)/2$
return the smoothed trajectory $\boldsymbol{x}_{\mathrm{smooth}}(t)$, with $\boldsymbol{\Delta V}_{\mathrm{smooth}}$



a) Planar motion planning query **b) 3-dimensional motion planning query**

Fig. 9 Illustrations of the planar and 3D motion plan queries in the LVLH frame.



a) LandSat-7 schematic [nadir (-δx) direction points down, while the in track (+δy) direction points left] **b) LandSat-7 orbit (courtesy of the landsat-7 handbook)**

Fig. 10 Target spacecraft geometry and orbital scenario used in numerical experiments.

as possible without violating trajectory constraints. Note because $\boldsymbol{\Delta V}_n$ lies in the feasible set of Eq. (16) the algorithm can only improve the final propellant cost. By design, Algorithm 2 is geared toward reducing our original solution propellant cost as quickly as possible while maintaining feasibility; the most expensive computational components are the calculation of $\boldsymbol{\Delta V}^\dagger$ and constraint checking (consistent with our sampling-based algorithm). Fortunately, the number of constraint checks is limited by the maximum number of iterations $\lceil \log_2(\frac{1}{\delta\alpha_{\min}}) \rceil + 1$, given tolerance $\delta\alpha_{\min} \in (0,1)$. As an added bonus, for strictly time-constrained applications that require a solution in a fixed amount of time, the algorithm can be easily modified to return the α_ℓ-weighted trajectory $\boldsymbol{x}_{\text{smooth}}(t)$ when time runs out, as the feasibility of this trajectory is maintained as an algorithm invariant.

VI. Numerical Experiments

Consider the two scenarios shown in Fig. 9 modeling both planar and nonplanar near-field approaches of a chaser spacecraft in close proximity[¶] to a target moving on a circular LEO trajectory (as in Fig. 1). We imagine the chaser, which starts in a circular orbit of lower radius, must be repositioned through a sequence of prespecified CWH waypoints (e.g., required for equipment checks, surveying, *etc.*) to a coplanar position located radially above the target, arriving with zero relative velocity in preparation for a final radial (R-bar) approach. Throughout the maneuver, as described in detail in Sec. II, the chaser must avoid entering the elliptic target KOZ, enforce a hard two-fault tolerance to stuck-off thruster failures, and otherwise avoid interfering with the target. This includes avoiding the target's nadir-pointing communication lobes (represented by truncated half-cones) and preventing exhaust plume impingement on its surfaces. For context, we use the Landsat-7 [44] spacecraft and orbit as a reference ([45] Sec. 3.2) (see Figs. 10 and 11).

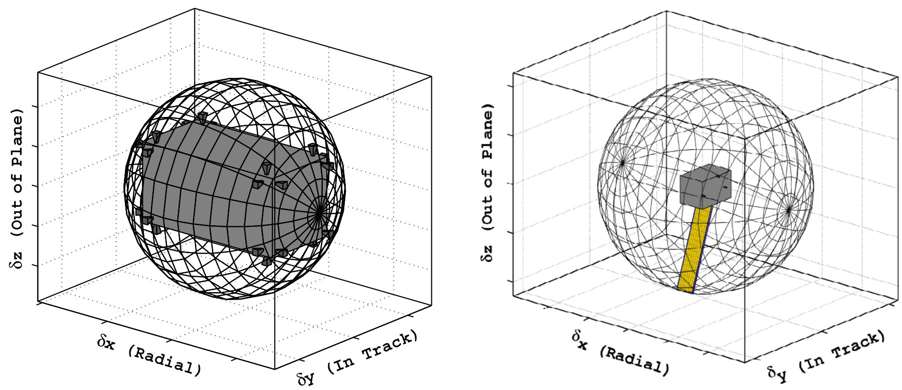
A. Simulation Setup

Before proceeding to the results, we first outline some key features of our setup. Taking the prescribed query waypoints one at a time as individual goal points $\boldsymbol{x}_{\text{goal}}$, we solve the given scenario as a series of motion planning problems (or subplans) linked together into an overall solution, calling FMT* from Sec. IV.C once for each subplan. For this multiplan problem, we take the solution cost to be the sum of individual subplan costs (when using trajectory smoothing, the endpoints between two plans are merged identically to two edges within a plan, as described in Sec. V).

As our steering controller from Sec. IV.A is attitude independent, states $\boldsymbol{x} \in \mathbb{R}^d$ are either $\boldsymbol{x}^T = [\delta x, \delta y, \delta\dot{x}, \delta\dot{y}]$ with $d = 4$ (planar case) or $\boldsymbol{x}^T = [\delta x, \delta y, \delta z, \delta\dot{x}, \delta\dot{y}, \delta\dot{z}]$ with $d = 6$ (nonplanar case). This omission of the attitude $\boldsymbol{q}$ from the state is achieved by employing an attitude policy (assuming a stabilizing attitude controller), which produces $\boldsymbol{q}(t)$ from the state trajectory $\boldsymbol{x}(t)$. For illustration purposes, a simple nadir-pointing attitude profile is chosen during nominal guidance, representing a mission requiring constant communication with the ground; for actively safe abort, we assume a simple turn–burn–turn policy, which orients the closest-available thruster under each failure mode as quickly as possible into the direction required for circularization (see Sec. III.C).

Given the hyperrectangular shape of the state space, we call upon the deterministic, low-dispersion d-dimensional Halton sequence [40] to sample positions and velocities. To improve sampling densities, each subplan uses its own sample space defined around its respective initial and goal waypoints, with some arbitrary threshold

[¶]Close proximity in this context implies that any higher-order terms of the linearized relative dynamics are negligible, e.g., within a few percent of the target orbit mean radius.



a) Chaser spacecraft **b) Target spacecraft**

Fig. 11 Models of the chaser and target, together with their circumscribing spheres.

space added around them. Additionally, extra samples n_{goal} are taken inside each waypoint ball to facilitate convergence.

Finally, we make note of three additional implementation details. First, for clarity, we list all relevant simulation parameters in Table 1. Second, all position-related constraint checks regard the chaser spacecraft as a point at its center of mass, with all other obstacles artificially inflated by the radius of its circumscribing sphere. Third and finally, all trajectory constraint checking is implemented by pointwise evaluation with a fixed time-step resolution Δt, using the analytic state transition equations Eq. (4) together with steering solutions from Sec. IV.A to propagate graph edges; for speed, the line segments between points are excluded. Except near very sharp obstacle corners, this approximation is generally not a problem in practice (obstacles can always be inflated further to account for this). To improve performance, each obstacle primitive (ellipsoid, right-circular cone, hypercube, *etc.*) employs hierarchical collision checking using hyperspherical and/or hyperrectangular bounding volumes to quickly prune points from consideration.

B. Planar Motion Planning Solution

A representative solution to the posed planning scenario, both with and without the trajectory smoothing algorithm (Algorithm 2), is shown in Fig. 12. As shown, the planner successfully finds safe trajectories within each subplan, which are afterward linked to form an overall solution. The state space of the first subplan shown at the bottom is essentially unconstrained, as the chaser at this point is too far away from the target for plume impingement to come into play.

Table 1 List of parameters used during near-circular orbit rendezvous simulations

Parameter	Value
Chaser plume half-angle, β_{plume}	10 deg
Chaser plume height, H_{plume}	16 m
Chaser thruster fault tolerance, F	2
Cost threshold, $\bar{J}$	0.1–0.4 m/s
Dimension, d	4 (planar), 6 (nonplanar)
Goal sample count, n_{goal}	$0.04n$
Goal position tolerance, ϵ_r	3–8 m
Goal velocity tolerance, ϵ_v	0.1–0.5 m/s
Max. allocated thruster $\Delta\boldsymbol{v}$ magnitude, $\Delta v_{\max,k}$	∞ m/s
Max. commanded $\Delta\boldsymbol{v}$-vector magnitude $\lVert\Delta\boldsymbol{v}_i\rVert$, $\Delta v_{\max}$	∞ m/s
Max. plan duration, $T_{\text{plan,max}}$	∞ s
Min. plan duration, $T_{\text{plan,min}}$	0 s
Max. steering maneuver duration, $T_{\max}$	$0.1 \cdot (2\pi/n_{\text{ref}})$
Min. steering maneuver duration, $T_{\min}$	0 s
Sample count, n	50–400 per plan
Simulation time step, Δt	$0.0005 \cdot (2\pi/n_{\text{ref}})$
Smoothing tolerance, $\delta\alpha_{\min}$	0.01
Target antenna lobe height	75 m
Target antenna beamwidth	60°
Target KOZ semi-axes, $[\rho_{\delta x}, \rho_{\delta y}, \rho_{\delta z}]$	[35 50 15] m

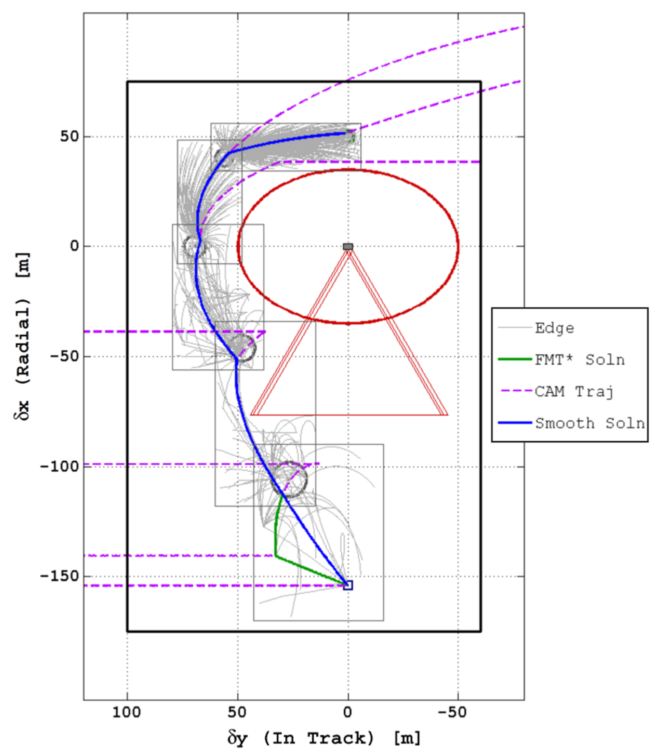
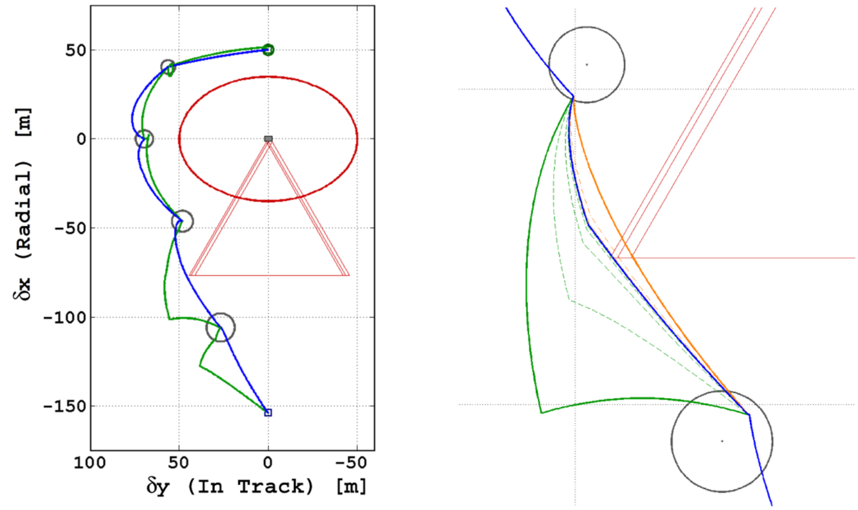


Fig. 12 Representative planar motion planning solution using the FMT* algorithm (Algorithm 1) with $n = 2000$ (400 per subplan), $\bar{J} = 0.3$ m/s, and relaxed waypoint convergence (Soln, Solution; Traj, Trajectory).

This means every edge connection attempted here is added, so the first subplan illustrates well a discrete subset of the reachable states around $\boldsymbol{x}_{\text{init}}$ and the unrestrained growth of FMT*. As the second subplan is reached, the effects of the KOZ position constraints come in to play, and we see edges begin to take more leftward loops. In subplans 3 and 4, plume impingement begins to play a role. Finally, in subplan 5 at the top, where it becomes very cheap to move between states (as the spacecraft can simply coast to the right for free), we see the initial state connecting to nearly every sample in the subspace, resulting in a straight shot to the final goal. As is evident, straight-line path planning would not approximate these trajectories well, particularly near coasting arcs, along which our dynamics allow the spacecraft to transfer for free.

To understand the smoothing process, examine Fig. 13. Here, we see how the discrete trajectory sequence from our sampling-based algorithm may be smoothly and continuously deformed toward the unconstrained minimal-propellant trajectory until it meets trajectory constraints (as outlined in Sec. V); if these constraints happen to be inactive, then the exact minimal-propellant trajectory is returned, as



a) Paths before and after smoothing (n = 2000, $\bar{J}$ = 0.2 m/s, exact waypoint convergence)

b) Smoothing algorithm iterations (n = 15000, $\bar{J}$ = 0.3 m/s, inexact waypoint convergence)

Fig. 13 Visualizing trajectory smoothing (Algorithm 2) for the solution shown in Fig. 12.

Fig. 13a shows. This computational approach is generally quite fast, assuming a well-implemented convex solver is used, as will be seen in the results of the next subsection.

The net $\Delta \boldsymbol{v}$ costs of the two reported trajectories in this example come to 0.835 (unsmoothed) and 0.811 m/s (smoothed). Compare this to 0.641 m/s, the cost of the unconstrained direct solution that intercepts each of the goal waypoints on its way to rendezvousing with $\boldsymbol{x}_{\mathrm{goal}}$ (this trajectory exits the state space along the positive in-track direction, a violation of our proposed mission; hence, its cost represents an underapproximation to the true optimal cost J of the constrained problem). This suggests that our solutions are quite close to the constrained optimum, and certainly on the right order of magnitude. Particularly with the addition of smoothing at lower sample counts, the approach appears to be a viable one for spacecraft planning.

If we compare the net $\Delta \boldsymbol{v}$ costs to the actual measured propellant consumption given by the sum total of all allocated thruster $\Delta \boldsymbol{v}$ magnitudes [which equal 1.06 (unsmoothed) and 1.01 m/s (smoothed)], we find increases of 27.0 and 24.5%; as expected, our sum-of-2-norms propellant-cost metric underapproximates the true propellant cost. For point masses with isotropic control authority (e.g., a steerable or gimbaled thruster that is able to point freely in any direction), our cost metric would be exact. Although inexact for our distributed attitude-dependent propulsion system (see Fig. 11a), it is clearly a reasonable proxy for allocated propellant use, returning values on the same order of magnitude. Although we cannot make a strong statement about our proximity to the propellant-optimal solution without directly optimizing over thruster $\Delta \boldsymbol{v}$ allocations, our solution clearly seems to promote low propellant consumption.

C. Nonplanar Motion Planning Solution

For the nonplanar case, representative smoothed and unsmoothed FMT* solutions can be found in Fig. 14. Here, the spacecraft is required to move out of plane to survey the target from above before reaching the final goal position located radially above the target. The first subplan involves a long reroute around the conical region spanned by the target's communication lobes. Because the chaser begins in a coplanar circular orbit at $\boldsymbol{x}_{\mathrm{init}}$, most steering trajectories require a fairly large cost to maneuver out of plane to the first waypoint. Consequently, relatively few edges that both lie in the reachable set of $\boldsymbol{x}_{\mathrm{init}}$ and safely avoid the large conical obstacles are added. As we progress to the second and third subplans, the corresponding trees become denser (more steering trajectories are both safe and within our cost threshold $\bar{J}$) as the state space becomes more open. Compared with the planar case, the extra degree of freedom associated with the out-of-plane dimension appears to allow more edges ahead of the target in the in-track direction than before, likely because now the exhaust plumes generated by the chaser are well out of plane from the target spacecraft. Hence, the spacecraft smoothly and tightly curls around the ellipsoidal KOZ to the goal.

The net $\Delta \boldsymbol{v}$ costs for this example come to 0.611 (unsmoothed) and 0.422 m/s (smoothed). Counterintuitively, these costs are on the same order of magnitude and slightly cheaper than the planar case; the added freedom given by the out-of-plane dimension appears to outweigh the high costs typically associated with inclination changes and out-of-plane motion. These cost values correspond to total thruster $\Delta \boldsymbol{v}$ allocation costs of 0.893 and 0.620 m/s, respectively, increases of 46 and 47% above their counterpart cost metric values. Again, our cost metric appears to be a reasonable proxy for actual propellant use.

D. Performance Evaluation

To evaluate the performance of our approach, an assessment of solution quality is necessary as a function of planning parameters, i.e., the number of samples n taken and the reachability set cost threshold $\bar{J}$. As proven in Sec. IV.D, the solution cost will eventually reduce to the optimal value as we increase the sample size n. Alternatively, we can attempt to reduce the cost by increasing the cost threshold $\bar{J}$ used for nearest-neighbor identification so that more connections are explored. Both, however, work at the expense of the running time. To understand the effects of these changes on quality, especially at finite sample counts for which the asymptotic guarantees of FMT* do not necessarily imply cost improvements, we measure the cost vs computation time for the planar planning scenario parameterized over several values of n and $\bar{J}$.

Results are reported in Figs. 15 and 16. Here, we call FMT* once each for a series of sample count/cost threshold pairs, plotting the total cost of successful runs at their respective run times** as measured by wall clock time (CVXGEN and CVX [46], disciplined convex programming solvers, are used to implement $\Delta \boldsymbol{v}$ allocation and trajectory smoothing, respectively). Note only the online components of each FMT* call, i.e., graph search/construction, constraint checking, and termination evaluations, constitute the run times reported; everything else may either be stored onboard before mission launch or otherwise computed offline on ground computers and later uplinked to the spacecraft. See Sec. IV.C for details. Samples are stored as a $d \times n$ array, while intersample steering controls $\Delta \boldsymbol{v}_i$ and times τ_i are precomputed as $n \times n$ arrays of $d/2 \times N$ and $N \times 1$ array elements, respectively. Steering state and attitude

**All simulations are implemented in MATLAB® 2012b and run on a PC operated by Windows 10, clocked at 4.00 GHz, and equipped with 32.0 GB of RAM.

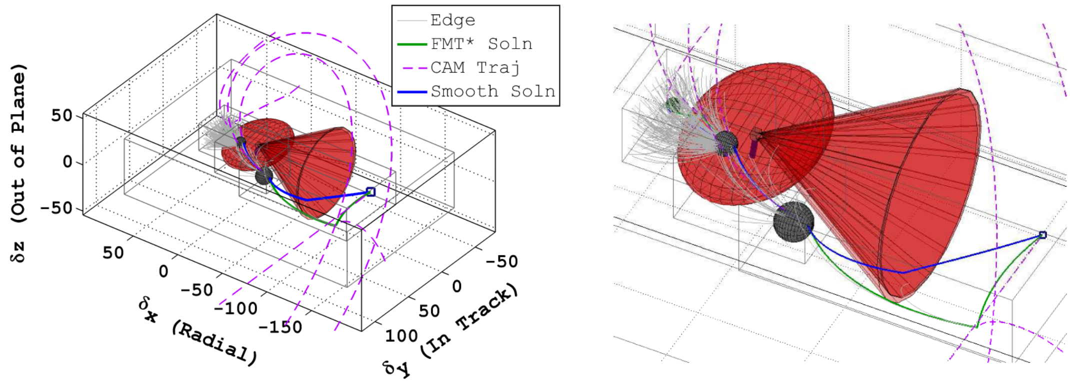
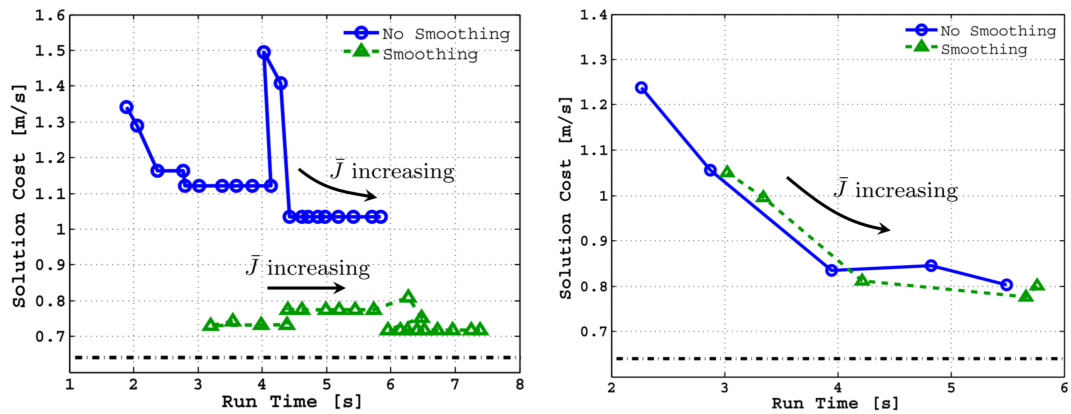
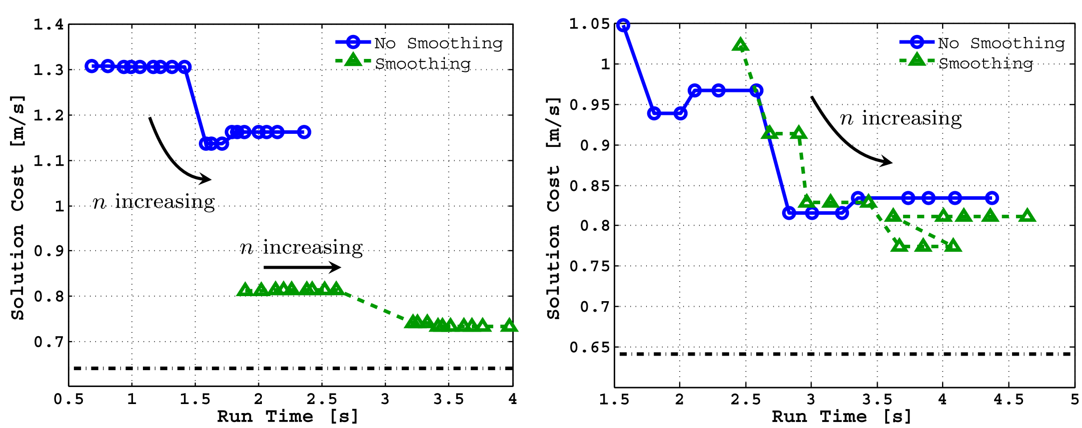


Fig. 14 **Representative nonplanar motion planning solution using the FMT* algorithm (Algorithm 1) with $n = 900$ (300 per subplan), $\bar{J} = 0.4$ m/s, and relaxed waypoint convergence.**



a) Exact waypoint convergence (n = 2000) b) Inexact waypoint convergence (n = 2000)

Fig. 15 **Algorithm performance for the given LEO proximity operations scenario as a function of varying cost threshold ($\bar{J} \in [0.2, 0.4]$) with n held constant.**



a) Exact waypoint convergence ($\bar{J}$ = 0.22 m/s) b) Inexact waypoint convergence ($\bar{J}$ = 0.3 m/s)

Fig. 16 **Algorithm performance for the given LEO proximity operations scenario as a function of varying sample count ($n \in [650, 2000]$) with $\bar{J}$ held constant.**

trajectories $\boldsymbol{x}(t)$ and $\boldsymbol{q}(t)$, on the other hand, are generated online through Eq. (4) and our nadir-pointing attitude policy, respectively. This reduces memory requirements, though nothing precludes them from being generated and stored offline as well, to save additional computation time.

Figure 15 reports the effects on the solution cost from varying the cost threshold $\bar{J}$ while keeping n fixed. As described in Sec. IV.B, increasing $\bar{J}$ implies a larger reachability set size and hence increases the number of candidate neighbors evaluated during graph construction. Generally, this gives a cost improvement at the expense of extra processing, although there are exceptions, as in Fig. 15a at $\bar{J} \approx 0.3$ m/s. Likely, this arises from a single new neighbor (connected at the expense of another, since FMT* only adds one edge per neighborhood) that readjusts the entire graph subtree, ultimately increasing the cost of exact termination at the goal. Indeed, we see that this does not occur where inexact convergence is permitted, given the same sample distribution.

We can also vary the sample count n while holding $\bar{J}$ constant. From Figs. 15a and 15b, we select $\bar{J} = 0.22$ and 0.3 m/s, respectively, for each of the two cases (the values that suggest the best solution cost per unit of run time). Repeating the simulation for varying sample count values, we obtain Fig. 16. Note the general downward trend as the run time increases (corresponding to larger sample counts), a classical tradeoff in sampling-based planning. However, there is bumpiness. Similar to before, this is likely due to new connections previously unavailable at lower sample counts that cause a slightly different graph with an unlucky jump in the propellant cost.

As the figures show, the utility of trajectory smoothing is clearly affected by the fidelity of the planning simulation. In each, trajectory smoothing yields a much larger improvement in cost at modest increases in computation time when we require exact waypoint convergence. It provides little improvement, on the other hand, when we relax these waypoint tolerances; FMT* (with goal region sampling) seems to return trajectories with costs much closer to the optimum in such cases, making the additional overhead of smoothing less favorable. This conclusion is likely highly problem dependent; these tools must always be tested and tuned to the particular application.

Note that the overall run times for each simulation are on the order of 1–5 s, including smoothing. This clearly indicates that FMT* can return high-quality solutions in real time for spacecraft proximity operations. Although run on a computer currently unavailable to spacecraft, we hope that our examples serve as a reasonable proof of concept; we expect that with a more efficient coding language and implementation our approach would be competitive on spacecraft hardware.

VII. Conclusions

A technique has been presented for automating minimum-propellant guidance during near-circular orbit proximity operations, enabling the computation of near-optimal collision-free trajectories in real time (on the order of 1–5 s for our numerical examples). The approach uses a modified version of the FMT* sampling-based motion planning algorithm to approximate the solution to minimal-propellant guidance under impulsive Clohessy–Wiltshire–Hill (CWH) dynamics. Our method begins by discretizing the feasible space of Eq. (1) through state-space sampling in the CWH local-vertical local-horizontal (LVLH) frame. Next, state samples and their cost reachability sets, which we have shown comprise sets bounded by unions of ellipsoids taken over the steering maneuver duration, are precomputed offline and stored onboard the spacecraft together with all pairwise steering solutions. Finally, FMT* is called online to efficiently construct a tree of trajectories through the feasible state space toward a goal region, returning a solution that satisfies a broad range of trajectory constraints (e.g., plume impingement and obstacle avoidance, control allocation feasibility, *etc.*) or else reporting failure. If desired, trajectory smoothing using the techniques outlined in Sec. V can be employed to reduce the solution propellant cost.

The key breakthrough of our solution for autonomous spacecraft guidance is its judicious distribution of computations; in essence, only what must be computed onboard, such as collision checking and graph construction, is computed online; everything else, including the most intensive computations, are relegated to the ground where computational effort and execution time are less critical. Furthermore, only minimal information (steering problem control trajectories, costs, and nearest-neighbor sets) requires storage onboard the spacecraft. Although we have illustrated through simulations the ability to tackle a particular minimum-propellant LEO homing maneuver problem, it should be noted that the methodology applies equally well to other objectives, such as the minimum-time problem, and can be generalized to other dynamic models and environments. The approach is flexible enough to handle nonconvexity and mixed state–control–time constraints without compromising real-time implementability, so long as constraint function evaluation is relatively efficient. In short, the proposed approach appears to be useful for automating the mission planning process for spacecraft proximity operations, enabling real-time computation of low-cost trajectories.

The proposed guidance framework for impulsively actuated spacecraft offers several avenues for future research. For example, though nothing in the methodology forbids it outside of computational limitations, it would be interesting to revisit the problem with attitude states included in the state (instead of assuming an attitude profile). This would allow attitude constraints into the planning process (e.g., enforcing the line of sight, keeping solar panels oriented towards the sun, maintaining a communication link with the ground, *etc.*). Also of interest are other actively safe policies that relax the need to circularize escape orbits (potentially costly in terms of propellant use) or that mesh better with trajectory smoothing, without the need to add compensating impulses (see Sec. V). Extensions to dynamic obstacles (such as debris or maneuvering spacecraft, which are unfixed in the LVLH frame), elliptical target orbits, higher-order gravitation, or dynamics under relative orbital elements also represent key research areas useful for extending applicability to more general maneuvers. Finally, memory and run time performance evaluations of our algorithms on spacelike hardware are needed to assess our method's true practical benefit to spacecraft planning.

Appendix A: Optimal Circularization Under Impulsive CWH Dynamics

As detailed in Sec. III.C.1, a vital component of our CAM policy is the generation of one-burn minimal-propellant transfers to circular orbits located radially above or below the target. Assuming we need to abort from some state $\boldsymbol{x}(t_{\text{fail}}) = \boldsymbol{x}_{\text{fail}}$, the problem we wish to solve in order to assure safety (per Definition 4 and as seen in Fig. 6) is

$$
\begin{aligned}
&\text{Given: failure state } \boldsymbol{x}_{\text{fail}}, \text{ and CAM } \boldsymbol{u}_{\text{CAM}}(t_{\text{fail}} \le t < T_h^-) \triangleq 0, \boldsymbol{u}_{\text{CAM}}(T_h) \triangleq \boldsymbol{\Delta v}_{\text{circ}}(\boldsymbol{x}(T_h)) \\
&\underset{T_h}{\text{minimize}} \ \ \Delta v_{\text{circ}}^2(T_h) \\
&\text{subject to } \ \boldsymbol{x}_{\text{CAM}}(t_{\text{fail}}) = \boldsymbol{x}_{\text{fail}} \qquad \text{initial condition} \\
&\qquad \boldsymbol{x}_{\text{CAM}}(T_h^+) \in \mathcal{X}_{\text{invariant}} \qquad \text{invariant set termination} \\
&\qquad \dot{\boldsymbol{x}}_{\text{CAM}}(t) = f(\boldsymbol{x}_{\text{CAM}}(t), 0, t), \quad \text{for all } t_{\text{fail}} \le t \le T_h \qquad \text{system dynamics} \\
&\qquad \boldsymbol{x}_{\text{CAM}}(t) \notin \mathcal{X}_{\text{KOZ}}, \quad \text{for all } t_{\text{fail}} \le t \le T_h \qquad \text{KOZ collision avoidance}
\end{aligned}
$$

Because of the analytical descriptions of state transitions, as given by Eq. (4), it is a straightforward task to express the decision variable T_h, invariant set constraint, and objective function analytically in terms of $\theta(t) = n_{\text{ref}}(t - t_{\text{fail}})$, the polar angle of the target spacecraft. The problem is therefore one dimensional in terms of θ. We can reduce the invariant set termination constraint to an invariant set positioning constraint if we ensure the spacecraft ends up at a position inside $\mathcal{X}_{\text{invariant}}$ and circularize the orbit, since $\boldsymbol{x}(\theta_{\text{circ}}^+) = \boldsymbol{x}(\theta_{\text{circ}}^-) + \begin{bmatrix} 0 \\ \boldsymbol{\Delta v}_{\text{circ}}(\theta_{\text{circ}}) \end{bmatrix} \in \mathcal{X}_{\text{invariant}}$. Denote $\theta_{\text{circ}} = n_{\text{ref}}(T_h - t_{\text{fail}})$ as the target anomaly at which we enforce circularization. Now, suppose the failure state $\boldsymbol{x}(t)$ lies outside of the KOZ (otherwise, there exists no safe CAM, and we conclude that $\boldsymbol{x}_{\text{fail}}$ is unsafe). We can define a new variable $\theta_{\min} = n_{\text{ref}}(t - t_{\text{fail}})$ and integrate the coasting dynamics forward from time t_{fail} until the chaser touches the boundary of the KOZ ($\theta_{\max} = \theta_{\text{collision}}^-$) or until we have reached one full orbit ($\theta_{\max} = \theta_{\min} + 2\pi$) such that, between these two bounds, the CAM trajectory satisfies the dynamics and contains only the coasting segment outside of the KOZ. Replacing the dynamics and collision-avoidance constraints with the bounds on θ as a box constraint, the problem becomes

$$\begin{aligned}&\underset{\theta_{\mathrm{circ}}}{\text{minimize}}\ \Delta v^2_{\mathrm{circ}}(\theta_{\mathrm{circ}})\\ &\text{subject to } \theta_{\min} \le \theta_{\mathrm{circ}} \le \theta_{\max} \qquad \text{theta bounds}\\ &\qquad \delta x^2(\theta^-_{\mathrm{circ}}) \ge \rho^2_{\delta x} \qquad \text{invariant set positioning}\end{aligned}$$

Restricting our search range to $\theta \in [\theta_{\min}, \theta_{\max}]$, this is a function of one variable and one constraint. Solving by the method of Lagrange multipliers, we seek to minimize the Lagrangian, $\mathcal{L} = \Delta v^2_{\mathrm{circ}} + \lambda g_{\mathrm{circ}}$, where $g_{\mathrm{circ}}(\theta) = \rho^2_{\delta x} - \delta x^2(\theta^-_{\mathrm{circ}})$. There are two cases to consider:

Case 1 is the inactive invariant set positioning constraint. We set $\lambda = 0$ such that $\mathcal{L} = \Delta v^2_{\mathrm{circ}}$. Candidate optimizers θ must satisfy $\nabla_\theta \mathcal{L}(\theta) = 0$. Taking the gradient of $\mathcal{L}$,

$$\begin{aligned}\nabla_\theta \mathcal{L} &= \frac{\partial \Delta v^2_{\mathrm{circ}}}{\partial \theta}\\ &= \left[\frac{3}{4}(3n_{\mathrm{ref}}\delta x_{\mathrm{fail}} + 2\delta\dot{y}_{\mathrm{fail}})^2 - \frac{3}{4}\delta\dot{x}^2_{\mathrm{fail}} + n^2_{\mathrm{ref}}\delta z^2_{\mathrm{fail}} - \delta\dot{z}^2_{\mathrm{fail}}\right]\sin 2\theta\\ &\quad + \left[\frac{3}{2}\delta\dot{x}_{\mathrm{fail}}(3n_{\mathrm{ref}}\delta x_{\mathrm{fail}} + 2\delta\dot{y}_{\mathrm{fail}}) - 2n_{\mathrm{ref}}\delta\dot{z}_{\mathrm{fail}}\delta z_{\mathrm{fail}}\right]\cos 2\theta\end{aligned}$$

and setting $\nabla_\theta \mathcal{L}(\theta) = 0$, we find that

$$\tan 2\theta^* = \frac{-[(3/2)\delta\dot{x}_{\mathrm{fail}}(3n_{\mathrm{ref}}\delta x_{\mathrm{fail}} + 2\delta\dot{y}_{\mathrm{fail}}) - 2n_{\mathrm{ref}}\delta\dot{z}_{\mathrm{fail}}\delta z_{\mathrm{fail}}]}{(3/4)(3n_{\mathrm{ref}}\delta x_{\mathrm{fail}} + 2\delta\dot{y}_{\mathrm{fail}})^2 - (3/4)\delta\dot{x}^2_{\mathrm{fail}} + n^2_{\mathrm{ref}}\delta z^2_{\mathrm{fail}} - \delta\dot{z}^2_{\mathrm{fail}}}$$

Denote the set of candidate solutions that satisfy Case 1 by Θ_1.

Case 2 is the active invariant set positioning constraint. Here, the chaser attempts to circularize its orbit at the boundary of the zero-thrust RIC shown in Fig. 5a. The positioning constraint is active, and therefore $g_{\mathrm{circ}}(\theta) = \rho^2_{\delta x} - \delta x^2(\theta^-_{\mathrm{circ}}) = 0$. This is equivalent to finding where the coasting trajectory from $\boldsymbol{x}(t_{\mathrm{fail}}) = \boldsymbol{x}_{\mathrm{fail}}$ crosses $\delta x(\theta) = \pm\rho_{\delta x}$ for $\theta \in [\theta_{\min}, \theta_{\max}]$. This can be achieved using standard root-finding algorithms. Denote the set of candidate solutions that satisfy Case 2 by Θ_2.

For the solution to the minimal-cost circularization burn, the global optimizer θ either lies on the boundary of the box constraint, at an unconstrained optimum ($\theta \in \Theta_1$), or at the boundary of the zero-thrust RIC ($\theta \in \Theta_2$), all of which are economically obtained through either numerical integration or a root-finding solver. Therefore, the minimal-cost circularization burn time T^*_h satisfies

$$\theta^* = n_{\mathrm{ref}}(T^*_h - t_{\mathrm{fail}}) = \underset{\theta \in \{\theta_{\min}, \theta_{\max}\} \cup \Theta^*_1 \cup \Theta^*_2}{\operatorname{argmin}} \Delta v^2_{\mathrm{circ}}(\theta)$$

If no solution exists (which can happen if and only if $\boldsymbol{x}_{\mathrm{fail}}$ starts inside the KOZ), there is no safe circularization CAM, and we therefore declare $\boldsymbol{x}_{\mathrm{fail}}$ unsafe. Otherwise, the CAM is saved for future trajectory feasibility verification under all failure modes of interest (see Sec. III.C.3). This forms the basis for our actively safe sampling routine in Algorithm 1, as described in detail in Sec. IV.C.

Appendix B: Intermediate Results for FMT* Optimality Proof

We report here two useful lemmas concerning bounds on the trajectory costs between samples, which are used throughout the asymptotic optimality proof for FMT* in Sec. IV. We begin with a lemma bounding the sizes of the minimum and maximum eigenvalues of $\boldsymbol{G}$, useful for bounding reachable volumes from $\boldsymbol{x}_0$. We then prove Lemma 2, which forms the basis of our asymptotic optimality analysis for FMT*. Here, $\boldsymbol{\Phi}(t_f, t_0) = e^{AT}$ is the state transition matrix, $T = t_f - t_0$ is the maneuver duration, and $\boldsymbol{G}$ is the $N = 2$ impulse Gramian matrix,

$$\boldsymbol{G}(T) = \boldsymbol{\Phi}_v \boldsymbol{\Phi}_v^{-1} = [\, e^{AT}\boldsymbol{B} \quad \boldsymbol{B}\,][\, e^{AT}\boldsymbol{B} \quad \boldsymbol{B}\,]^T \qquad \text{(B1)}$$

where $\boldsymbol{\Phi}_v(t, \{\tau_i\}_i)$ is the aggregate $\Delta\boldsymbol{v}$ transition matrix corresponding to burn times $\{\tau_i\}_i = \{t_0, t_f\}$.

Lemma 3 (bounds on Gramian eigenvalues)*:* Let $T_{\max}$ be less than one orbital period for the system dynamics of Sec. II.C, and let $\boldsymbol{G}(T)$ be defined as in Eq. (B1). Then, there exist constants $M_{\min}$, $M_{\max} > 0$ such that $\lambda_{\min}\boldsymbol{G}(T) \ge M_{\min}T^2$ and $\lambda_{\max}\boldsymbol{G}(T) \le M_{\max}$ for all $T \in (0, T_{\max}]$.

Proof: We bound the maximum eigenvalue of $\boldsymbol{G}$ through norm considerations, yielding $\lambda_{\max}[\boldsymbol{G}(T)] \le (\|e^{AT}\|\boldsymbol{B} + \|\boldsymbol{B}\|)^2 \le (e^{\|A\|T_{\max}} + 1)^2$, and take $M_{\max} = (e^{\|A\|T_{\max}} + 1)^2$. As long as $T_{\max}$ is less than one orbital period, $\boldsymbol{G}(T)$ only approaches singularity near $T = 0$ [38]. Explicitly Taylor expanding $\boldsymbol{G}(T)$ about $T = 0$ reveals that $\lambda_{\min}\boldsymbol{G}(T) = T^2/2 + O(T^3)$ for small T, and thus $\lambda_{\min}\boldsymbol{G}(T) = \Omega(T^2)$ for all $T \in (0, T_{\max}]$. □

Reiteration of Lemma 2 (steering with perturbed endpoints)*:* For a given steering trajectory $\boldsymbol{x}(t)$ with initial time t_0 and final time t_f, let $\boldsymbol{x}_0 \coloneqq \boldsymbol{x}(t_0)$, $\boldsymbol{x}_f \coloneqq \boldsymbol{x}(t_f)$, $T \coloneqq t_f - t_0$, and $J\!: = J(x_0, x_f)$. Consider now the perturbed steering trajectory $\tilde{x}(t)$ between perturbed start and end points $\tilde{\boldsymbol{x}}_0 = \boldsymbol{x}_0 + \delta\boldsymbol{x}_0$ and $\tilde{\boldsymbol{x}}_f = \boldsymbol{x}_f + \delta\boldsymbol{x}_f$ and its corresponding cost $J(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{x}}_f)$.

Case 1 ($T = 0$): There exists a perturbation center $\delta\boldsymbol{x}_c$ (consisting of only a position shift) with $\|\delta\boldsymbol{x}_c\| = O(J^2)$ such that, if $\|\delta\boldsymbol{x}_c\| \le \eta J^3$ and $\|\delta\boldsymbol{x}_f - \delta\boldsymbol{x}_f\| \le \eta J^3$, then $J(\tilde{x}_0, \tilde{x}_f) \le J[1 + 4\eta + O(J)]$, and the spatial deviation of the perturbed trajectory $\tilde{\boldsymbol{x}}(t)$ is $O(J)$.

Case 2 ($T > 0$): If $\|\delta\boldsymbol{x}_0\| \le \eta J^3$ and $\|\delta\boldsymbol{x}_f\| \le \eta J^3$, then $J(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{x}}_f) \le J[1 + O(\eta J^2 T^{-1})]$, and the spatial deviation of the perturbed trajectory $\tilde{\boldsymbol{x}}(t)$ from $\boldsymbol{x}(t)$ is $O(J)$.

Proof: For bounding the perturbed cost and $J(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{x}}_f)$, we consider the two cases separately.

Case 1 ($T = 0$): Here, two-impulse steering degenerates to a single-impulse $\Delta\boldsymbol{v}$; that is, $\boldsymbol{x}_f = \boldsymbol{x}_0 + \boldsymbol{B}\Delta\boldsymbol{v}$ with $\|\Delta\boldsymbol{v}\| = J$. To aid in the ensuing analysis, denote the position and velocity components of states $\boldsymbol{x} = [\boldsymbol{r}^T, \boldsymbol{v}^T]^T$ as $\boldsymbol{r} = [\boldsymbol{I}, 0]\boldsymbol{x}$ and $\boldsymbol{v} = [0, \boldsymbol{I}]\boldsymbol{x}$. Since $T = 0$, we have $\boldsymbol{r}_f = \boldsymbol{r}_0$ and $\boldsymbol{v}_f = \boldsymbol{v}_0 + \Delta\boldsymbol{v}$. We pick the perturbed steering duration $\tilde{T} = J^2$ (which will provide an upper bound on the optimal steering cost) and expand the steering system [Eq. (4)] for small time $\tilde{T}$ as

$$\boldsymbol{r}_f + \delta\boldsymbol{r}_f = \boldsymbol{r}_0 + \delta\boldsymbol{r}_0 + \tilde{T}(\boldsymbol{v}_0 + \delta\boldsymbol{v}_0 + \widetilde{\Delta\boldsymbol{v}}_1) + O(\tilde{T}^2) \qquad \text{(B2)}$$

$$\begin{aligned}\boldsymbol{v}_f + \delta\boldsymbol{v}_f &= \boldsymbol{v}_0 + \delta\boldsymbol{v}_0 + \widetilde{\Delta\boldsymbol{v}}_1 + \widetilde{\Delta\boldsymbol{v}}_2 + \tilde{T}[\boldsymbol{A}_{21}(\boldsymbol{r}_0 + \delta\boldsymbol{r}_0)\\ &\quad + \boldsymbol{A}_{22}(\boldsymbol{v}_0 + \delta\boldsymbol{v}_0 + \widetilde{\Delta\boldsymbol{v}}_1)] + O(\tilde{T}^2) \qquad \text{(B3)}\end{aligned}$$

where $\boldsymbol{A}_{21} = \begin{bmatrix} 3n^2_{\mathrm{ref}} & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & -n^2_{\mathrm{ref}}\end{bmatrix}$ and $\boldsymbol{A}_{22} = \begin{bmatrix} 0 & 2n_{\mathrm{ref}} & 0\\ -2n_{\mathrm{ref}} & 0 & 0\\ 0 & 0 & 0\end{bmatrix}$. Solving Eq. (B2) for $\widetilde{\Delta\boldsymbol{v}}_1$ to first order, we find $\widetilde{\Delta\boldsymbol{v}}_1 = \tilde{T}^{-1}(\delta\boldsymbol{r}_f - \delta\boldsymbol{r}_0) - \boldsymbol{v}_0 - \delta\boldsymbol{v}_0 + O(\tilde{T})$. By selecting $\delta\boldsymbol{x}_c = [\tilde{T}\boldsymbol{v}_0^T\ 0^T]^T$ [note that $\|\delta\boldsymbol{x}_c\| = J^2\|\boldsymbol{v}_0\| = O(J^2)$] and supposing that $\|\delta\boldsymbol{x}_0\| \le \eta J^3$ and $\|\delta\boldsymbol{x}_f - \delta\boldsymbol{x}_c\| \le \eta J^3$, we have that

$$\begin{aligned}\|\widetilde{\Delta\boldsymbol{v}}_1\| &\le J^{-2}(\|\delta\boldsymbol{x}_0\| + \|\delta\boldsymbol{x}_f - \delta\boldsymbol{x}_c\|) + \|\delta\boldsymbol{x}_0\| + O(J^2)\\ &= 2\eta J + O(J^2)\end{aligned}$$

Now, solving Eq. (B3) for $\widetilde{\Delta\boldsymbol{v}}_2 = \Delta\boldsymbol{v} + (\delta\boldsymbol{v}_f - \delta\boldsymbol{v}_0) - \widetilde{\Delta\boldsymbol{v}}_1 + O(J^2)$ and taking norm of both sides,

$$\begin{aligned}\|\widetilde{\Delta\boldsymbol{v}}_2\| &\le \|\Delta\boldsymbol{v}\| + (\|\delta\boldsymbol{x}_0\| + \|\delta\boldsymbol{x}_f - \delta\boldsymbol{x}_c\|) + 2\eta J + O(J^2)\\ &\le J + 2\eta J + O(J^2)\end{aligned}$$

Therefore, the perturbed cost satisfies

$$J(\tilde{x}_0, \tilde{x}_f) \le \|\widetilde{\Delta\boldsymbol{v}}_1\| + \|\widetilde{\Delta\boldsymbol{v}}_2\| \le J[1 + 4\eta + O(J)]$$

Case 2 ($T > 0$): We pick $\tilde{T} = T$ to compute an upper bound on the perturbed cost. Applying the explicit form of the steering control $\boldsymbol{\Delta V}$ [see Eq. (13)] along with the norm bound $\|\boldsymbol{\Phi}_v^{-1}\| = \lambda_{\min}(\boldsymbol{G})^{-1/2} \leq M_{\min}^{-1/2} T^{-1}$ from Lemma 3, we have

$$\begin{aligned} J(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{x}}_f) &\leq \|\boldsymbol{\Phi}_v^{-1}(t_f, \{t_0, t_f\})(\tilde{\boldsymbol{x}}_f - \Phi(t_f, t_0)\tilde{\boldsymbol{x}}_0)\| \\ &\leq \|\boldsymbol{\Phi}_v^{-1}(\boldsymbol{x}_f - \Phi\boldsymbol{x}_0)\| + \|\boldsymbol{\Phi}_v^{-1}\delta\boldsymbol{x}_f\| + \|\boldsymbol{\Phi}_v^{-1}\boldsymbol{\Phi}\delta\boldsymbol{x}_0\| \\ &\leq J + M_{\min}^{-1/2} T^{-1}\|\delta\boldsymbol{x}_f\| + M_{\min}^{-1/2} T^{-1} e^{\|A\|T_{\max}}\|\delta\boldsymbol{x}_0\| \\ &\leq J(1 + O(\eta J^2 T^{-1})) \end{aligned}$$

In both cases, the deviation of the perturbed steering trajectory $\tilde{\boldsymbol{x}}(t)$ from its closest point on the original trajectory is bounded (quite conservatively) by the maximum propagation of the difference in initial conditions; that is, the initial disturbance $\delta\boldsymbol{x}_0$ plus the difference in intercept burns $\widetilde{\boldsymbol{\Delta v}}_1 - \boldsymbol{\Delta v}_1$, over the maximum maneuver duration $T_{\max}$. Thus,

$$\begin{aligned} \|\tilde{\boldsymbol{x}}(t) - \boldsymbol{x}(t)\| &\leq e^{\|A\|T_{\max}}(\|\delta\boldsymbol{x}_0\| + \|\widetilde{\boldsymbol{\Delta v}}_1\| + \|\boldsymbol{\Delta v}_1\|) \\ &\leq e^{\|A\|T_{\max}}(\eta J^3 + 2J + o(J)) = O(J) \end{aligned}$$

where we have used $\|\boldsymbol{\Delta v}_1\| \leq J$ and $\|\widetilde{\boldsymbol{\Delta v}}_1\| \leq J(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{x}}_f) \leq J + o(J)$ from our previous arguments. □

Acknowledgments

This work was supported by an Early Career Faculty grant from NASA's Space Technology Research Grants Program (grant number NNX12AQ43G).

References

[1] Dannemiller, D. P., "Multi-Maneuver Clohessy-Wiltshire Targeting," *AAS Astrodynamics Specialist Conference*, American Astronomical Soc., Girdwood, AK, July 2011, pp. 1–15.

[2] Kriz, J., "A Uniform Solution of the Lambert Problem," *Celestial Mechanics*, Vol. 14, No. 4, Dec. 1976, pp. 509–513.

[3] Hablani, H. B., Tapper, M. L., and Dana Bashian, D. J., "Guidance and Relative Navigation for Autonomous Rendezvous in a Circular Orbit," *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 3, 2002, pp. 553–562.
doi:10.2514/2.4916

[4] Gaylor, D. E., and Barbee, B. W., "Algorithms for Safe Spacecraft Proximity Operations," *AAS Space Flight Mechanics Meeting*, Vol. 127, American Astronomical Soc., Sedona, AZ, Jan. 2007, pp. 132–152.

[5] Develle, M., Xu, Y., Pham, K., and Chen, G., "Fast Relative Guidance Approach for Autonomous Rendezvous and Docking Control," *Proceedings of SPIE*, Vol. 8044, Soc. of Photo-Optical Instrumentation Engineers, Orlando, FL, April 2011, Paper 80440F.

[6] Lopez, I., and McInnes, C. R., "Autonomous Rendezvous using Artificial Potential Function Guidance," *Journal of Guidance, Control, and Dynamics*, Vol. 18, No. 2, March 1995, pp. 237–241.
doi:10.2514/3.21375

[7] Muñoz, J. D., and Fitz Coy, N. G., "Rapid Path-Planning Options for Autonomous Proximity Operations of Spacecraft," *AAS Astrodynamics Specialist Conference*, American Astronomical Soc., Toronto, Canada, Aug. 2010, pp. 1–24.

[8] Açıkmeşe, B., and Blackmore, L., "Lossless Convexification of a Class of Optimal Control Problems with Non-Convex Control Constraints," *Automatica*, Vol. 47, No. 2, Feb. 2011, pp. 341–347.
doi:10.1016/j.automatica.2010.10.037

[9] Harris, M. W., and Açıkmeşe, B., "Lossless Convexification of Non-Convex Optimal Control Problems for State Constrained Linear Systems," *Automatica*, Vol. 50, No. 9, 2014, pp. 2304–2311.
doi:10.1016/j.automatica.2014.06.008

[10] Martinson, N., "Obstacle Avoidance Guidance and Control Algorithms for Spacecraft Maneuvers," *AIAA Conference on Guidance, Navigation, and Control*, Vol. 5672, AIAA, Reston, VA, Aug. 2009, pp. 1–9.

[11] Breger, L., and How, J. P., "Safe Trajectories for Autonomous Rendezvous of Spacecraft," *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 5, 2008, pp. 1478–1489.
doi:10.2514/1.29590

[12] Vazquez, R., Gavilan, F., and Camacho, E. F., "Trajectory Planning for Spacecraft Rendezvous with On/Off Thrusters," *International Federation of Automatic Control*, Vol. 18, Elsevier, Milan, Italy, Aug. 2011, pp. 8473–8478.

[13] Lu, P., and Liu, X., "Autonomous Trajectory Planning for Rendezvous and Proximity Operations by Conic Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 36, No. 2, March 2013, pp. 375–389.
doi:10.2514/1.58436

[14] Luo, Y.-Z., Liang, L.-B., Wang, H., and Tang, G.-J., "Quantitative Performance for Spacecraft Rendezvous Trajectory Safety," *Journal of Guidance, Control, and Dynamics*, Vol. 34, No. 4, July 2011, pp. 1264–1269.
doi:10.2514/1.52041

[15] 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.
doi:10.2514/2.4943

[16] LaValle, S. M., and Kuffner, J. J., "Randomized Kinodynamic Planning," *International Journal of Robotics Research*, Vol. 20, No. 5, 2001, pp. 378–400.
doi:10.1177/02783640122067453

[17] Frazzoli, E., "Quasi-Random Algorithms for Real-Time Spacecraft Motion Planning and Coordination," *Acta Astronautica*, Vol. 53, Nos. 4–10, Aug. 2003, pp. 485–495.
doi:10.1016/S0094-5765(03)80009-7

[18] Phillips, J. M., Kavraki, L. E., and Bedrossian, N., "Spacecraft Rendezvous and Docking with Real-Time, Randomized Optimization," *AIAA Conference on Guidance, Navigation and Control*, AIAA, Reston, VA, Aug. 2003, pp. 1–11.

[19] Kobilarov, M., and Pellegrino, S., "Trajectory Planning for CubeSat Short-Time-Scale Proximity Operations," *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 2, March 2014, pp. 566–579.
doi:10.2514/1.60289

[20] Haught, M., and Duncan, G., "Modeling Common Cause Failures of Thrusters on ISS Visiting Vehicles," NASA JSC-CN-30604, June 2014, http://ntrs.nasa.gov/search.jsp?R=20140004797 [retrieved Dec. 2015].

[21] Weiss, A., Baldwin, M., Petersen, C., Erwin, R. S., and Kolmanovsky, I. V., "Spacecraft Constrained Maneuver Planning for Moving Debris Avoidance Using Positively Invariant Constraint Admissible Sets," *American Control Conference*, IEEE Publ., Piscataway, NJ, June 2013, pp. 4802–4807.

[22] Carson, J. M., Açikmeşe, B., Murray, R. M., and MacMartin, D. G., "A Robust Model Predictive Control Algorithm with a Reactive Safety Mode," *Automatica*, Vol. 49, No. 5, May 2013, pp. 1251–1260.
doi:10.1016/j.automatica.2013.02.025

[23] Lavalle, S., *Planning Algorithms*, Cambridge Univ. Press, Cambridge, England, U.K., 2006, pp. 1–842.

[24] Janson, L., Schmerling, E., Clark, A., and Pavone, M., "Fast Marching Tree: A Fast Marching Sampling-Based Method for Optimal Motion Planning in Many Dimensions," *International Journal of Robotics Research*, Vol. 34, No. 7, 2015, pp. 883–921.
doi:10.1177/0278364915577958

[25] Starek, J. A., Barbee, B. W., and Pavone, M., "A Sampling-Based Approach to Spacecraft Autonomous Maneuvering with Safety Specifications," *American Astronomical Society Guidance Navigation and Control Conference*, Vol. 154, American Astronomical Soc., Breckenridge, CO, Feb. 2015, pp. 1–13.

[26] Starek, J. A., Schmerling, E., Maher, G. D., Barbee, B. W., and Pavone, M., "Real-Time, Propellant-Optimized Spacecraft Motion Planning under Clohessy-Wiltshire-Hill Dynamics," *IEEE Aerospace Conference*, IEEE Publ., Piscataway, NJ, March 2016, pp. 1–16.

[27] Ross, I. M., "How to Find Minimum-Fuel Controllers," *AIAA Conference on Guidance, Navigation and Control*, AAAA, Reston, VA, Aug. 2004, pp. 1–10.

[28] Clohessy, W. H., and Wiltshire, R. S., "Terminal Guidance System for Satellite Rendezvous," *Journal of the Aerospace Sciences*, Vol. 27, No. 9, Sept. 1960, pp. 653–658.
doi:10.2514/8.8704

[29] Hill, G. W., "Researches in the Lunar Theory," *JSTOR American Journal of Mathematics*, Vol. 1, No. 1, 1878, pp. 5–26.
doi:10.2307/2369430

[30] Bodson, M., "Evaluation of Optimization Methods for Control Allocation," *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 4, July 2002, pp. 703–711.
doi:10.2514/2.4937

[31] Dettleff, G., "Plume Flow and Plume Impingement in Space Technology," *Progress in Aerospace Sciences*, Vol. 28, No. 1, 1991, pp. 1–71.
doi:10.1016/0376-0421(91)90008-R

[32] Goodman, J. L., "History of Space Shuttle Rendezvous and Proximity Operations," *Journal of Spacecraft and Rockets*, Vol. 43, No. 5, Sept. 2006, pp. 944–959.
doi:10.2514/1.19653

[33] Starek, J. A., Açıkmeşe, B., Nesnas, I. A. D., and Pavone, M., "Spacecraft Autonomy Challenges for Next Generation Space Missions," *Advances in Control System Technology for Aerospace Applications*, edited by Feron, E., Vol. 460, Lecture Notes in Control and Information Sciences, Springer–Verlag, New York, Sept. 2015, pp. 1–48, Chap. 1.
doi:10.1007/978-3-662-47694-9_1

[34] Barbee, B. W., Carpenter, J. R., Heatwole, S., Markley, F. L., Moreau, M., Naasz, B. J., and Van Eepoel, J., "A Guidance and Navigation Strategy for Rendezvous and Proximity Operations with a Noncooperative Spacecraft in Geosynchronous Orbit," *Journal of the Aerospace Sciences*, Vol. 58, No. 3, July 2011, pp. 389–408.

[35] Schouwenaars, T., How, J. P., and Feron, E., "Decentralized Cooperative Trajectory Planning of Multiple Aircraft with Hard Safety Guarantees," *AIAA Conference on Guidance, Navigation and Control*, AIAA, Reston, VA, Aug. 2004, pp. 1–14.

[36] Fehse, W., *Automated Rendezvous and Docking of Spacecraft*, Vol. 16, Cambridge Univ. Press, Cambridge, England, U.K., 2003, pp. 1–494.

[37] Fraichard, T., "A Short Paper About Motion Safety," *Proceedings of IEEE Conference on Robotics and Automation*, IEEE Publ., Piscataway, NJ, April 2007, pp. 1140–1145.

[38] Alfriend, K., Vadali, S. R., Gurfil, P., How, J., and Breger, L., *Spacecraft Formation Flying: Dynamics, Control and Navigation*, Vol. 2, Butterworth-Heinemann, Burlington, MA, Nov. 2009, pp. 1–402.

[39] Janson, L., Ichter, B., and Pavone, M., "Deterministic Sampling-Based Motion Planning: Optimality, Complexity, and Performance," *17th International Symposium of Robotic Research (ISRR)*, Springer, Sept. 2015, p. 16, http://web.stanford.edu/~pavone/papers/Janson.Ichter.ea.ISRR15.pdf.

[40] Halton, J. H., "On the Efficiency of Certain Quasirandom Sequences of Points in Evaluating Multidimensional Integrals," *Numerische Mathematik*, Vol. 2, No. 1, 1960, pp. 84–90.
doi:10.1007/BF01386213

[41] Allen, R., and Pavone, M., "A Real-Time Framework for Kinodynamic Planning with Application to Quadrotor Obstacle Avoidance," *AIAA Conference on Guidance, Navigation and Control*, AIAA, Reston, VA, Jan. 2016, pp. 1–18.

[42] Schmerling, E., Janson, L., and Pavone, M., "Optimal Sampling-Based Motion Planning under Differential Constraints: the Drift Case with Linear Affine Dynamics," *Proceedings of IEEE Conference on Decision and Control*, IEEE Publ., Piscataway, NJ, Dec. 2015, pp. 2574–2581.
doi:10.1109/CDC.2015.7402604

[43] Schmerling, E., Janson, L., and Pavone, M., "Optimal Sampling-Based Motion Planning Under Differential Constraints: The Driftless Case," *Proceedings of IEEE Conference on Robotics and Automation*, IEEE Publ., Piscataway, NJ, May 2015, pp. 2368–2375.

[44] *Landsat 7 Science Data Users Handbook*, NASA, March 2011, http://landsathandbook.gsfc.nasa. gov/pdfs/LandSat7_Handbook.html.

[45] Goward, S. N., Masek, J. G., Williams, D. L., Irons, J. R., and Thompson, R. J., "The Landsat 7 Mission: Terrestrial Research and Applications for the 21st Century," *Remote Sensing of Environment*, Vol. 78, Nos. 1–2, 2001, pp. 3–12.
doi:10.1016/S0034-4257(01)00262-0

[46] Grant, M., and Boyd, S., "CVX: Matlab Software for Disciplined Convex Programming," Ver. 2.1, March 2014, http://cvxr.com/cvx/citing/ [retrieved June 2015].