Stochastic Motion Planning for Hopping Rovers on Small Solar System Bodies

B. Hockman and M. Pavone

Keywords: Space robotics, Planning under uncertainty, Reinforcement learning

Abstract Hopping rovers have emerged as a promising platform for the future surface exploration of small Solar System bodies, such as asteroids and comets. However, hopping dynamics are governed by nonlinear gravity fields and stochastic bouncing on highly irregular surfaces, which pose several challenges for traditional motion planning methods. This paper presents the first ever discussion of motion planning for hopping rovers that explicitly accounts for various sources of uncertainty. We first address the problem of planning a single hopping trajectory by developing (1) an algorithm for robustly solving Lambert's orbital boundary value problems in irregular gravity fields, and (2) a method for computing landing distributions by propagating control and model uncertainties—from which, a time/energy-optimal hop can be selected using a (myopic) policy gradient. We then cast the sequential planning problem as a Markov decision process and apply a sample-efficient, off-line, off-policy reinforcement learning algorithm—namely, a variant of least squares policy iteration (LSPI)—to derive approximately optimal control policies that are safe, efficient, and amenable to real-time implementation on computationally-constrained rover hardware. These policies are demonstrated in simulation to be robust to modelling errors and outperform previous heuristics.

1 Introduction

Small Solar System bodies, such as asteroids, comets, and irregular moons, have become a key target for exploration due to their scientific interest, potential for resource extraction, and for studying impact mitigation strategies. While some information about their chemical and structural properties can be obtained remotely, measurements that constrain composition and physical properties require direct contact with the surface at multiple locations and over extended periods of time [1]. Accordingly, NASA and space agencies worldwide have recognized the need for rovers capable of controlled surface mobility [2, 3, 4].

B. Hockman and M. Pavone are with the Department of Aeronautics and Astronautics, Stanford University, Stanford, CA, e-mail: {bhockman,pavone}@stanford.edu.
This work is supported by NASA under the Innovative Advanced Concepts program.

However, the surface environment of small bodies presents many unique challenges for mobility, including their highly irregular shape and extremely reduced gravity (10–1000 μg), which prohibit conventional wheeled rovers that rely on surface traction. Instead, *hopping* systems are more naturally suited for such environments, as they can traverse large distances over arbitrarily rough terrain with little energy. In fact, four small hoppers are currently en route to asteroid Ryugu aboard JAXA's Hayabusa 2 spacecraft—a MASCOT lander developed by DLR [3] and three MINERVA landers [4], which are equipped with internal momentum devices to provide a "kick" for hopping, albeit with minimal control.

To extend the idea of hopping to an architecture capable of targeted science, NASA has invested in various new technologies to enable *controlled* mobility in microgravity. As part of the Evolvable Mars Campaign, Howe and Gernhardt et. al investigated a pressurized, six-legged hopping excursion vehicle for human exploration of Mars' moon, Phobos [5] (see Fig. 1, right). The authors have been developing a small, internally-actuated hopping rover called "Hedgehog," which can perform various *motion primitives* by applying torques to three orthogonal flywheels, including long-range hops as well as short, precise tumbling (see Fig. 1, left). The dynamics and control of these motion primitives has been studied in detail, from analytical models to high fidelity simulations [6, 7, 8]. Experiments in various test beds have also validated these control laws, including a custom-built 6 DoF gravity-offloading test bed [8] and a parabolic flight campaign [9].

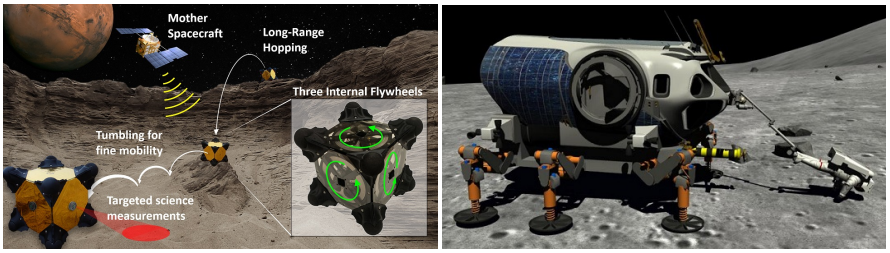


Fig. 1: Rovers designed for precise mobility on small bodies. **Left**: Hedgehog: a small (15-25 cm), internally-actuated rover that uses three internal flywheels to hop and tumble in microgravity. **Right**: ATHLETE hopper: a pressurized excursion vehicle that uses six spring-loaded legs to hop and transport humans on the surface of Phobos.

In addition to controllability, hopping rovers operating on distant bodies require a high degree of *autonomy*, as communication suffers from long light-speed delays and relies on a mothership relay, which may be infrequent. On Mars, wheeled rovers are equipped with visual perception, terrain classification, and path planning algorithms for autonomous mobility [10]. However, in contrast to wheeled rovers, which operate through continuous interaction with the environment, hoppers can only apply forces from rest on the surface and have no control of their trajectory mid-flight. Thus, autonomy for hopping systems requires a more discrete and sequential structure, which can be decomposed into four phases: (1) localization, (2) trajectory planning, (3) hop execution, and (4) ballistic dynamics (see Fig. 2).

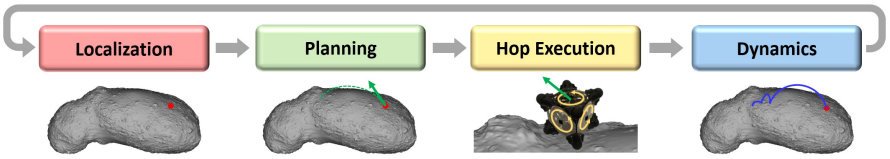


Fig. 2: The high-level autonomy architecture for hopping rovers consists of four phases: (1) localization, (2) trajectory planning, (3) hop execution, and (4) uncontrolled, ballistic dynamics.

Localization on the surface of small bodies is an open area of research, which has been addressed from two perspectives: (1) assuming an "eye-in-the-sky" mothership [11], and (2) mothership-independent, on-board perception, primarily through visual odometry [12]. While each method has its strengths and weaknesses, any approach is likely to produce pose estimates with some level of uncertainty.

Hop execution addresses the problem of pushing off from the surface to achieve some desired velocity vector ($\mathbf{v}^*$) and, as discussed for Hedgehog in [8], is highly dependent on the rover architecture (e.g. actuation mechanisms, surface interaction, etc.). Generally, for a given hopper, $\mathbf{v}^*$ can only be executed approximately—due to environmental uncertainty and control errors—and must obey certain constraints, such as speed and direction limitations.

Dynamics about small bodies has a rich body of literature for orbiting spacecraft [13], and to a lesser extent for surface interaction [14, 15]. The key challenges lie in accurate modeling of the gravity field about irregularly shaped bodies and the physical interactions with the surface. However, even the highest fidelity gravity and contact models rely on information that may not be available a priori via remote observations, such as internal mass distribution and surface structure.

The focus of this paper is on phase II, *planning*, which is by far the least addressed element in the autonomy stack. Planning seeks to answer the question:

> *"What is the next best hop to perform, given a set of mission objectives, an estimate of the rover's location, an understanding of its capabilities, and a model of the world?"*

In other words, planning represents the rover's decision-making module and is tightly intertwined with all other elements in the mobility pipeline. Previous studies of planning for hopping rovers have assumed highly simplified and deterministic models of the dynamics, localization, and the hopper itself. Bellerose, et. al derive analytical control laws for a spherical hopper on a smooth, spherical asteroid with a coulomb friction contact model, and exact localization [16]. This work is extended to the case of smooth, *ellipsoidal* asteroids in [17], which also derives approximate speed constraints to prevent escape trajectories. However, these results are founded on oversimplifications of the body shape, gravity, and contact models, as evidenced by remote observations of highly uneven and rocky surfaces (see Fig. 3), and the chaotic bouncing pattern of the Philae lander on comet 67P [18].

Statement of Contributions: In contrast to the approaches of [16] and [17], the goal of this paper is *not* to derive closed-form analytical expressions for optimal control laws, which requires a number of unrealistic simplifications. Instead, we

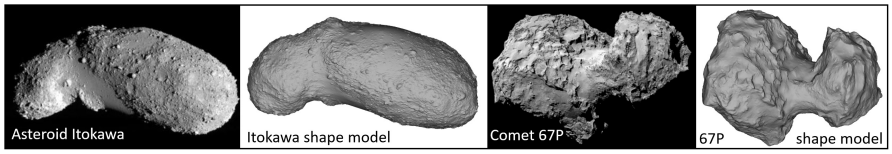


Fig. 3: High-resolution images and shape models of two small bodies. **Left:** Asteroid 25143 Itokawa (535 m, $6-9\,\mu g$), **Right:** Comet 67P Churyumov-Gerasimenko (4.3 km, $140-300\,\mu g$).

propose a planning architecture that directly accounts for various sources of uncertainty, and thus, produces control policies that are more robust to modeling, control, and localization errors. The key idea is to shift from a "first-principles" approach, to a data-driven approach, whereby high-fidelity dynamics models can be used to simulate *instances* of hopping trajectories from various uncertainty distributions.

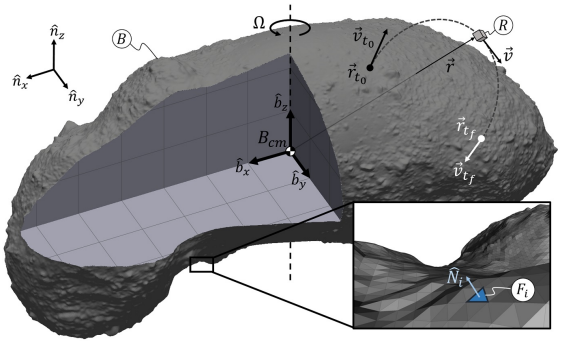
Specifically, the contributions of this paper are twofold: First, we solve the problem of planning a single hop under uncertainty and assuming minimal bouncing (Sect. 3). We develop a robust and efficient algorithm for solving "Lambert's orbital boundary value problem"—the problem of finding an initial velocity to intercept a target—in highly irregular gravity fields (Sect. 3.1). We then forward propagate control and gravity uncertainty (modeled as Gaussian mixtures) through the dynamics to compute trajectory funnels and landing distributions for a given nominal trajectory (Sect. 3.2). Then, we compute myopic policy gradients based on these landing distributions to derive time/energy-optimal single-hop control policies (Sect. 3.3). Second, in Sect. 4 we extend the planning problem to the case where multiple hops are required and structure the problem as an MDP. We apply a variant of Least Squares Policy Iteration (LSPI) to derive approximately optimal control policies that are safe, efficient, and amenable to real-time implementation on computationally-constrained rover hardware. The performance of these policies is evaluated on a high fidelity dynamics simulator and compared to that of a greedy heuristic policy. Collectively, the methods presented in this paper constitute the first ever study of uncertainty-aware motion planning for hopping rovers—a crucial component of autonomy for future exploration missions to small Solar System bodies.

2 Preliminaries

In this section, we present the dynamics models used to simulate the trajectories of a hopping rover. Section 2.1 details the forces acting on the rover during ballistic flight, and Sect. 2.2 discusses the model used for the contact dynamics of bouncing on the surface.

2.1 Ballistic Dynamics

A rover, R, at position $\mathbf{r}$ and velocity $\mathbf{v}$ relative to the asteroid body, B, hops from rest at $\mathbf{r}_{t_0}$ with velocity $\mathbf{v}_{t_0}$ and impacts at $\mathbf{r}_{t_f}$ with velocity $\mathbf{v}_{t_f}$. The body is represented as a closed polygonal mesh with k triangular facets, where facet F_i has outward normal $\mathbf{N}_i$. The asteroid rotates at a constant angular velocity $\omega_B = \Omega \hat{b}_z$.



	Definition
B	small body
R	hopping rover
B_{cm}	body's center of mass
$\hat{n}$	inertial frame
$\hat{b}$	frame fixed to B at B_{cm}
$\mathbf{r}$	rover position from B_{cm}
$\mathbf{v}$	rover velocity relative to $\hat{b}$
Ω	ang. velocity of B about $\hat{b}_z$
F_i	i^{th} triangular surface facet
$\hat{N}_i$	outward normal of facet F_i

Fig. 4: Dynamic model of rover R hopping on body B, which is rotating at $\omega_B = \Omega\hat{b}_z$ and is represented by a closed surface mesh consisting of k triangular facets, F_i.

The external forces ($\mathbf{F}_e$) acting on a hopping rover include gravitational forces (of the primary and possibly tertiary bodies), solar radiation pressure (SRP), electrostatic forces, and contact forces. It will also be convenient to represent the rover's dynamics in the rotating body frame, thus introducing effective centrifugal and Coriolis forces. The total effective force is expressed as,

$$\mathbf{F} = \mathbf{F}_e - m\omega_B \times (\omega_B \times \mathbf{r}) - 2m\omega_B \times \mathbf{v}. \tag{1}$$

In general, $\mathbf{F}_e$ (expressed in $\hat{b}$) is a function of not only position and velocity, but also time. In this paper, we focus on the *stationary* case to derive time-invariant control policies, which excludes forces that are periodic when expressed in the rotating body frame such as SRP and third body perturbations. This assumption is not overly restrictive since SRP is typically three to six orders of magnitude weaker than gravity in close proximity to small bodies, and most small bodies of interest for exploration are either gravitationally isolated (e.g., Itokawa, Bennu, and Psyche) or tidally locked secondaries (e.g., Phobos and Deimos). However, if periodic force models are important, the methods in this paper can be generalized by augmenting the rover's "state" with a temporal state (e.g., body phase). Nutation of the body's spin axis can also be accounted for in this way, although most asteroids are believed to be in stable spin about their major axis [19].

Gravity on small bodies may be orders of magnitude weaker than on Earth, but it still represents the dominant force on rovers, so accurate modeling is essential. With only shape information, the most accurate gravity model is a polyhedral model [20], which leverages the divergence theorem to exactly model the gravitational potential (U), attraction ($\mathbf{g} = \nabla U$), gradient ($\nabla\nabla U$), and Laplacian ($\nabla^2 U$) of a constant density polyhedron as a summation over all facets and edges of the surface mesh. This representation, while highly computational, is especially critical in close proximity to irregular bodies (see, e.g., Fig 3), where conventional models such as harmonic expansions and mascons yield large errors [20]. We alleviate the computational burden of evaluating the surface integral at each time step by precomputing the gravity field at regular grid points within the vicinity of the body *offline*, and then interpo-

lating within this precomputed field online. Validation tests comparing this approximate interpolation to exact evaluation suggest errors on the order of 0.01% for a 5m field discretization on Itokawa using a 5000-facet shape model.

However, most small bodies are likely *not* constant density, and internal density variations are, in general, not know a priori. The polyhedral model may be superimposed with harmonics or mascons if in situ measurements are taken, but a model of *uncertainty* is perhaps more import for robust policy generation. We consider a parametric uncertainty model,

$$\mathbf{g} = \bar{\mathbf{g}} + \delta\mathbf{g}, \quad \delta\mathbf{g} \sim P_\theta, \quad \theta \in \mathbb{R}^k, \tag{2}$$

where $\bar{\mathbf{g}} \in \mathbb{R}^3$ is the nominal modeled gravity vector, and $\delta\mathbf{g}$ is a random perturbation from distribution P_θ. As a simple example, Gaussian uncertainty on the *mass* of the body could be encoded with a single parameter, $\delta\mathbf{g} \sim \mathcal{N}(0, \sigma_M^2)$.

2.2 Contact Model

Finally, we require a model for the dynamics of the rover upon impact with the surface. In some cases, it may be sufficient to assume that the rover can achieve a dead-stick landing (e.g., if it is equipped with a damping mechanism), but in general, uncontrolled impact will cause the rover to rebound somewhat randomly. The dynamic response of an impact event is a complex function of the physical properties of both the surface and rover, and the speed and orientation of the rover upon contact. Tardivel and Van wal, et al. developed high-fidelity small body lander simulations to model the rebound and settling behavior of landers using rigid-body contact models [14, 15], and Murdock, et al. have studied low-velocity impact dynamics in granular media [21]. However, for the motion planning problem, we only need a model of the rebound *distribution*,

$$\mathbf{v}_{t_f^+} \sim P_\zeta(\mathbf{v}_{t_f^-}, \zeta(\mathbf{r}_{t_f})), \tag{3}$$

where the rebound velocity, $\mathbf{v}_{t_f^+}$, is a random variable dependent on the pre-impact velocity, $\mathbf{v}_{t_f^-}$, and a parametrized description of the surface properties, $\zeta(\mathbf{r}_{t_f})$ (e.g. surface friction and elasticity). This allows us to abstract away the detailed contact physics in a general way, whereby the rover can be modeled as a dimensionless particle. In practice, P_ζ should be fit to the rebound dynamics observed on a higher fidelity model. For the data used in Sect. 4.2, we use a kernel density estimator to fit P_ζ to the rebound dynamics of a cube impacting a flat surface with some friction and elasticity. The simulation is stopped when $||\mathbf{v}_{t_f^-}|| < v_{\min}$. An example of 20 Monte Carlo simulations is shown in Fig. 5.

3 Single-Hop Planning

Before addressing the inherently sequential planning problem, we first consider the simpler problem of planning a single hop. However, even this problem is far from trivial, as the rover must contend with highly nonlinear dynamics (discussed

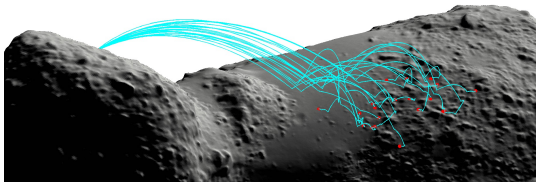


Fig. 5: Monte Carlo simulation of a single hopping trajectory subject to control, gravity, and rebound uncertainty. The rover is modeled as a particle and is subject to forces based on Eq. (1).

in Sect. 2) and many sources of uncertainty. In this section, we develop a framework for studying stochastic hopping trajectories and extracting approximately optimal (myopic) control policies.

3.1 Impact Targeting

In some cases, it may be desirable for a hopper to target a specific touch down location in a single hop (e.g., if the impact rebounds are expected to be minimal). The problem of computing the launch velocity ($\mathbf{v}_{t_0}$) to intercept a target location ($\mathbf{r}_{t_f}$) at time $\tau = t_f - t_0$ is the well-known "Lambert orbital boundary-value problem," and has efficient numerical solutions for spherical [22] and perturbed [23] gravity fields. However, for a polyhedral gravity model, a shooting method is required, which relies on good initial guesses for convergence.

Accordingly, we propose an algorithm that procedurally solves for the set of initial velocities ($\mathbf{v}_{t_0}$) that correspond to a *range* of flight times $\tau = [\tau_1, ..., \tau_n]^T$. Algorithm 1 leverages three key insights to robustly and efficiently compute the solution set. First, the dynamics model in the shooting solver (line 3) *ignores collisions* with the surface, which leverages the fact that the polyhedral gravity model is also valid inside the body [20]. Avoiding collision checks makes the dynamics continuous and differentiable and drastically speeds up integration. Surface penetration is checked for feasibility only after the solution has converged (line 6). The second key insight is that gravity has a second order effect on position, and thus has more influence the longer it is integrated. Thus, by setting τ_1 sufficiently small, the solution is close to a straight line between $\mathbf{r}_{t_0}$ and $\mathbf{r}_{t_f}$, and, although it may often be infeasible (or impractical), this serves as a robust initialization for subsequent solutions for larger τ. Finally, we leverage the differentiability of the dynamics to make a good initial guess of $\mathbf{v}_{t_0}(\tau_{i+1})$ given the solution for $\mathbf{v}_{t_0}(\tau_i)$ and the Jacobian, $J(\tau_i)$ (lines 4-5).

Lambert's problem is known to have multiple solutions for a given τ: two for each integer number of orbits. Although Alg. 1 will always return the most direct family of solutions (i.e., shortest path with zero orbits), other families of solutions may be found through a bisection search on $\mathbf{v}_{t_0}$. Figure 6 illustrates an example of three such families of solutions for a given $(\mathbf{r}_{t_0}, \mathbf{r}_{t_f})$ pair, which vary in duration from 25 minutes to 4 hours (blue trajectories represent the nominal, most direct solution family). Interestingly, the fact that *three* families of solutions exist within a single orbit—albeit, not all feasible—contradicts the Lambert solution for spherical gravity, thus illustrating the importance of high-fidelity gravity models.

Algorithm 1 Procedural Lambert Solver for Irregular Gravity Fields

Input: $\mathbf{r}_{t_0}, \mathbf{r}_{t_f} \in \mathbb{R}^3$, $\tau = [\tau_1, ..., \tau_n]^T \in \mathbb{R}^n$ s.t. $\tau_{i+1} > \tau_i$, gravity field $\bar{\mathbf{g}}(\mathbf{r})$

1: initialize guess for $\mathbf{v}_{t_0}(\tau_1)$ (e.g., as Lambert solution in spherical gravity field)
2: **for** $i = 1, ..., n$ **do**
3: Use shooting solver to solve for $\mathbf{v}_{t_0}(\tau_i)$ and Jacobian, $J(\tau_i)$:
$\mathbf{v}_{t_0}(\tau_i),\ \mathbf{v}_{t_f}(\tau_i),\ J(\tau_i) \leftarrow \text{Solve}\big(\mathbf{r}_{t_0}, \mathbf{r}_{t_f}, \bar{\mathbf{g}}(\mathbf{r}), \mathbf{v}_{t_0}(\tau_i)\big)$
4: Compute first-order estimate of partial $\mathbf{v}_{t_0}$ w.r.t. τ: $\frac{\partial \mathbf{v}_{t_0}}{\partial \tau} \leftarrow J(\tau_i)^{-1}\mathbf{v}_{t_f}(\tau_i)$
5: Initial guess for τ_{i+1}: $\mathbf{v}_{t_0}(\tau_{i+1}) \leftarrow \mathbf{v}_{t_0}(\tau_i) + \frac{\partial \mathbf{v}_{t_0}}{\partial \tau}(\tau_{i+1} - \tau_i)$
6: isValid(i) $\leftarrow$ Check for trajectory surface penetration
7: **end for**

Output: $\mathbf{v}_{t_0}, \mathbf{v}_{t_f} \in \mathbb{R}^{3\times n}$, $J \in \mathbb{R}^{3\times 3\times n}$, isValid $\in \{0,1\}^n$

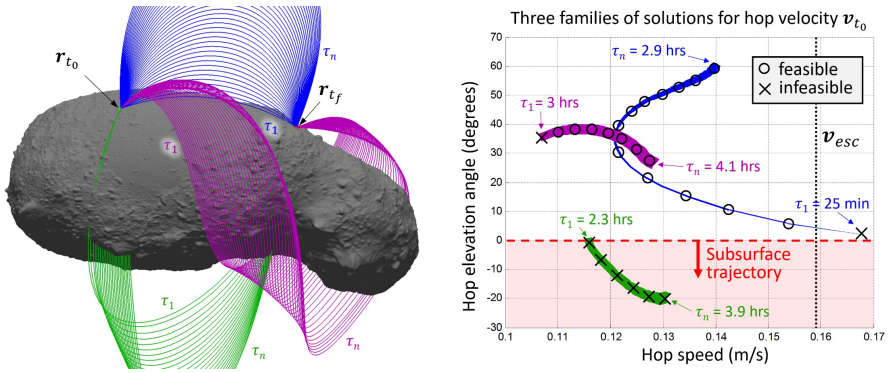


Fig. 6: **Left:** Three families of hopping solutions computed by Alg. 1, ranging from short and direct to long and winding. **Right:** Plot of hopping speed, $\|\mathbf{v}_{t_0}\|$, vs. elevation angle (with respect to the local surface plane). Line thickness is proportional to τ for that solution.

3.2 Uncertainty Propagation

Computing the nominal Lambert solutions for planning a single hop helps to inform what trajectories may be beneficial for targeting a specific impact location, but it assumes a perfect gravity model, perfect control accuracy, and perfect state information. It is also important to understand how sensitive these solutions are to various sources of uncertainty. In other words, we would like to predict the impact *distribution* by treating the gravity field, control accuracy, and state estimate as random variables rather than known quantities.

In general, there are two approaches to density estimation for nonlinear functions of random variables: (1) analytical propagation of simplified uncertainty models through linearized dynamics, and (2) sampling-based techniques. While sampling-based techniques (e.g., kernel methods) are amenable to arbitrarily complex dynamics and uncertainty models, they typically assume some measure of "local smoothness" and do not scale well to higher dimensions. On the other hand, analytical methods can be much more sample efficient for high-dimensional uncertainty models (e.g., by approximating gradients), but are often restricted to simple uncertainty

models and locally linear dynamics. The ballistic dynamics of hopping are indeed smooth and linearizable, but also depend on the collision with a highly irregular surface. Accordingly, we decompose the density estimation problem into two phases: (1) Gaussian error propagation through the linearized ballistic dynamics, and (2) projection onto the irregular surface mesh.

For error propagation through the ballistic dynamics, we assume Gaussian uncertainty on the gravity, $\mathbf{g}$, according to Eq. (2) and Gaussian uncertainty on the control ($\mathbf{v}_0$) and initial state ($\mathbf{r}_0$) according to $\mathbf{v}_0 \sim \mathcal{N}(\mu_{\mathbf{v}_0}, \Sigma_{\mathbf{v}_0})$, and $\mathbf{r}_0 \sim \mathcal{N}(\mu_{\mathbf{r}_0}, \Sigma_{\mathbf{r}_0})$. More generally, the joint uncertainty of $\mathbf{g}$, $\mathbf{v}_0$, and $\mathbf{r}_0$ may be expressed as,

$$\begin{bmatrix} \delta\mathbf{g} \\ \mathbf{v}_0 \\ \mathbf{r}_0 \end{bmatrix} \sim \mathcal{N}(\mu_{\delta\mathbf{g},\mathbf{v}_0,\mathbf{r}_0}, \Sigma_{\delta\mathbf{g},\mathbf{v}_0,\mathbf{r}_0}),\ \mu_{\delta\mathbf{g},\mathbf{v}_0,\mathbf{r}_0} = \begin{bmatrix} \mathbf{0} \\ \mu_{\mathbf{v}_0} \\ \mu_{\mathbf{r}_0} \end{bmatrix},\ \Sigma_{\delta\mathbf{g},\mathbf{v}_0,\mathbf{r}_0} = \begin{bmatrix} \Sigma_{\delta\mathbf{g}} & \mathbf{0} \\ \mathbf{0} & \Sigma_{\mathbf{v}_0,\mathbf{r}_0} \end{bmatrix}, \quad (4)$$

where $\Sigma_{\delta\mathbf{g}} \in \mathbb{S}^k_+$ is the covariance of the gravity model, and $\Sigma_{\mathbf{v}_0,\mathbf{r}_0} \in \mathbb{S}^6_+$ is the joint state-control covariance. A two-sided finite difference approximation of the Jacobian, $J_{\mathbf{v}_0,\mathbf{r}_0} = [\partial\mathbf{r}_f/\partial\theta \ \ \partial\mathbf{r}_f/\partial\mathbf{v}_0 \ \ \partial\mathbf{r}_f/\partial\mathbf{r}_0] \in \mathbb{R}^{3\times(k+6)}$, can be approximated with $2(k+6)$ simulations (note that $\partial\mathbf{r}_f/\partial\mathbf{v}_0$ is obtained for free from Alg. 1). With this linear approximation, the impact covariance about $\mathbf{r}_f$ can be computed as $\Sigma_{\mathbf{r}_f} = J_{\mathbf{v}_0,\mathbf{r}_0}\Sigma_{\delta\mathbf{g},\mathbf{v}_0,\mathbf{r}_0}J^T_{\mathbf{v}_0,\mathbf{r}_0}$.

To get an impact distribution *over the surface*, we then project this covariance along $\mathbf{v}_f$, whereby the probability of impact on any facet can be computed as,

$$P(\mathbf{r}_{t_f} \in F_i) \ \approx \ a_i \left(\frac{e^{-\frac{1}{2}\mathbf{r}^T_{F_i}\Sigma^{-1}_{uw}\mathbf{r}_{F_i}}}{-2\pi\sqrt{|\Sigma_{uw}|}} \right) \max\left\{\hat{N}_i \cdot \hat{c}_v, \ \ 0\right\}, \quad (5)$$

$$\text{where} \qquad \Sigma_{uwv} = {}^cR^b\,\Sigma_{\mathbf{r}_f}\,{}^cR^{bT} = \begin{bmatrix} \Sigma_{uw} & \begin{matrix}\sigma_u\sigma_v \\ \sigma_w\sigma_v\end{matrix} \\ \begin{matrix}\sigma_v\sigma_u & \sigma_v\sigma_w\end{matrix} & \sigma_v^2 \end{bmatrix}. \quad (6)$$

Here, ${}^cR^b$ is the rotation matrix from $\hat{b}$ to $\hat{c}$, $\mathbf{r}_{F_i} \in \mathbb{R}^2$ is the vector from $\mathbf{r}_{t_f}$ to the center of facet F_i (in $\langle \hat{c}_u, \hat{c}_w \rangle$), and a_i is the area of facet F_i (see Fig. 7, left).

Figure 7 (right) illustrates this two-phase error propagation method on three trajectories from the example in Fig. 6, where trajectory funnels bound a 90% confidence envelope, and surface color denotes projected uncertainty distribution, $\Sigma_{\mathbf{r}_f}$. Interestingly, these funnels are not always divergent along the trajectory as one might expect (e.g. the blue trajectory). While the total magnitude of the error ($|\Sigma_{uwv}|$) does generally grow, its projection along the trajectory (i.e., $|\Sigma_{uw}|$) can shrink, indicating that errors can grow in *time* while shrinking in spatial dispersion. Also, while this analysis assumed Gaussian uncertainty, it can be trivially extended to Gaussian mixture models by simply taking a weighted sum of the components.

One drawback of the projection in Eq. 5 is that it does not capture "shadowing" effects, which is the case whenever $\hat{N}_i \cdot \hat{c}_v > 0$. For surfaces that are highly irregular in the vicinity of $\mathbf{r}_{t_f}$ or for shallow, glancing impacts, Eq. 5 may be augmented with a "ray-tracing" collision checker to accurately project shadows—an established, albeit more computational technique commonly used for graphics rendering.

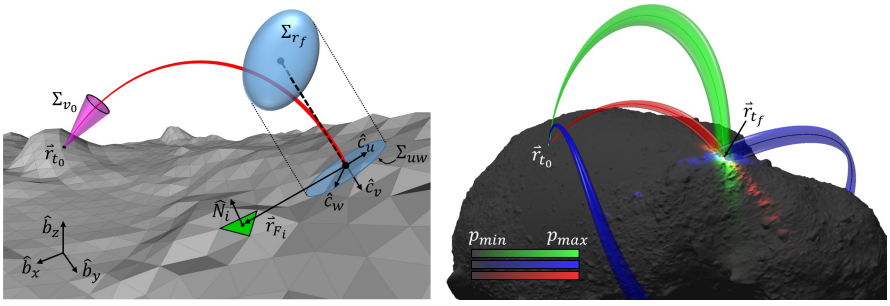


Fig. 7: **Left:** Landing distributions are computed using covariance propagation and projection. **Right:** Error propagation for three trajectories on Itokawa, corresponding to position uncertainty, $\Sigma_{r_0} = I\sigma_{r_0}^2, \sigma_{r_0} = 1\,m$, velocity error $\Sigma_{v_0} = I\sigma_{v_0}^2, \sigma_{v_0} = 0.03||\mathbf{v}_0||$, and gravity error $\sigma_g = 0.03||\tilde{\mathbf{g}}||$.

3.3 Optimal Hop Selection

Equipped with an algorithm to compute exact solutions for Lambert's two-point boundary value problem (Sect. 3.1) and a method for propagating uncertainty to compute approximate landing distributions (Sect. 3.2), we can now formulate an optimization problem for selecting the best (nominal) hop velocity, $\mu_{\mathbf{v}_0}^*$. We consider the following optimization problem:

$$\begin{aligned} \underset{\mu_{\mathbf{v}_0}}{\text{minimize}} \quad & J(\mu_{\mathbf{v}_0}) = \mathbb{E}\left[\lambda_u \mathbf{v}_0^T S_u \mathbf{v}_0 + \lambda_T \tau(\mathbf{r}_0, \mathbf{v}_0) - V(\mathbf{r}_f | \mathbf{r}_0, \mathbf{v}_0)\right] \\ \text{subject to} \quad & \mu_{\mathbf{v}_0} \in \mathcal{A}(\mathbf{r}_0) \\ \text{where} \quad & \mathbf{v}_0 \sim \mathcal{N}(\mu_{\mathbf{v}_0}, \Sigma_{\mathbf{v}_0}), \quad \mathbf{r}_0 \sim \mathcal{N}(\mu_{\mathbf{r}_0}, \Sigma_{\mathbf{r}_0}) \end{aligned} \tag{7}$$

where the additive cost function, J, represents an expectation of the control effort (weighted by λ_u), flight time, τ (weighted by λ_τ), and the negative "value," V, of impacting at location $\mathbf{r}_f$. The surface value map, V, may encode various mission objectives including the distance to the goal, possible hazards, or even the expected future rewards in the context of sequential hopping (to be addressed in Sect. 4). The constraint on $\mu_{\mathbf{v}_0}$ belonging to the action space of the rover, $\mathcal{A}$, at position estimate, $(\mu_{\mathbf{r}_0}, \Sigma_{\mathbf{v}_0})$, can be quite naturally constructed as the intersection of two convex sets representing the *speed*, $||\mu_{\mathbf{v}_0}|| \leq v_{\max}$, and *direction*, $\cos^{-1}(\mu_{\mathbf{v}_0} \cdot \hat{N}(\mathbf{r}_0)/||\mu_{\mathbf{v}_0}||) \leq \pi/2 - \theta_{\min}$, which is a cone about the local surface normal vector, $\hat{N}(\mathbf{r}_0)$, bounded by the minimum elevation angle, $\theta_{\min}$.

Assuming that V primarily encodes a "distance-to-goal" metric, choosing one solution from each homotopy class of trajectories (e.g., the blue and pink solution families in Fig. 6) can serve as good initializations for a gradient descent solver. An estimate for the global minimum can then be obtained by comparing the local minimum obtained from each homotopy class. For example, in the case where $\lambda_\tau \gg \lambda_u$, $\mu_{\mathbf{v}_0}^*$ belongs to the most direct class (blue trajectories in Fig. 6), whereas for $\lambda_u \gg \lambda_\tau$, $\mu_{\mathbf{v}_0}^*$ belongs to the longer, albeit less energy intensive solution class (pink trajectories in Fig. 6). The details of the gradient descent solver are omitted for

brevity, but at a high level, $\mathbb{E}(V)$ can be derived from Eq. (5) and finite difference estimates of the cost gradient, $\nabla_{\mu_{v_0}} J$, can be obtained in a similar fashion to the Jacobian in Sect. 3.2.

4 Sequential Hop Planning

Recall the sequential autonomy architecture illustrated in Fig. 2. For mobility tasks that require multiple hops (e.g., traversing long distances or correcting for unfavorable bouncing), we must extend the myopic strategies developed in Sect. 3 for planning over a longer horizon. This requires some notion of how immediate actions facilitate future actions and how this sequence of actions achieves certain mission objectives. A natural framework for modeling this inherently discrete and stochastic planning problem is a Markov Decision Process (MDP). In contrast to "classical" open-loop motion planning algorithms (e.g., combinatorial and sampling-based) that search for feasible (or even "optimal") reference trajectories, MDPs provide a more explicit representation of uncertainty and a powerful reward structure for encoding more complex mission objectives (i.e., not just "steering towards the goal"). Section 4.1 outlines how the planning problem can be structured as an MDP, Sect. 4.2 discusses a sample-efficient reinforcement learning method for learning approximate state-action value (Q-) functions and implicitly, approximately optimal policies. Finally, Sect. 4.4 compares the performance of learned control policies to heuristics proposed in [6], and evaluates performance robustness to modeling errors.

4.1 Hopping as an MDP

In accordance with the "classic" infinite horizon MDP formulation, we cast the sequential hop planning problem as the five-tuple, (S, A, T, R, γ)—the state space, action space, transition model, reward model, and discount factor. However, unlike most planning problems in robotics that force an MDP structure by temporally discretizing an inherently continuous-in-time process, hopping has a natural sequential decomposition, where transitions are marked by the eventual settling of each hop.

State space: At rest on the surface, the rover's state is simply its position and orientation. However, assuming that a lower level controller can reorient the rover as needed (as is the case for Hedgehog), we can collapse the state to just the surface position—an irregular manifold in $\mathbb{R}^3$. For nearly-spherical bodies, spherical coordinates (i.e., latitude/longitude) may be sufficient to uniquely parametrize the surface, but for highly irregular bodies such as those in Fig. 3, more elaborate map projections may be required. More generally, other state formulations might also include the rover's internal state (e.g., battery charge), its state history (e.g., in the context of a coverage problem), or its *belief* state (in the case of partial observability).

Action space: In its most fundamental form, the action space of a hopping rover can be described by raw actuators (e.g., three motors and brakes for Hedgehog). However, for motion planning it is more convenient to consider the rover's action as its velocity immediately after hopping—a higher level abstraction of the action space that leverages a lower level hopping controller (e.g., the controller discussed in [8]) and is amenable to the ballistic particle simulator presented in Sect. 2.

Moreover, it is critical that this velocity vector be expressed in a *global* reference frame—rather than a local surface frame—to maintain continuity in the transition dynamics; informally, $T(\cdot|s,a) \approx T(\cdot|s+\delta s,a)$. In other words, action descriptions that depend on the local surface slope (e.g., "spin flywheel number 2" or "hop left") can yield sharp changes in T for small changes in state on irregular terrain, whereas global descriptions (e.g, "hop north") are unaffected by local changes in topography.

However, local surface properties impose critical constraints on the feasible action space of the rover, $\mathcal{A}(s)$ (e.g., that the velocity vector must lie within some "friction cone" about the local surface normal). Thus, expressing actions in a global frame helps to "smooth" the dynamics (and consequently, the Q-function) but comes at the cost of requiring sharp discontinuities in $\mathcal{A}(s)$, suggesting that it may be advantageous to store a policy *implicitly* through a Q-function approximator (i.e. $\pi(s) = \arg\max_{a\in\mathcal{A}(s)} \hat{Q}(s,a)$) rather than explicit function approximators on π.

Reward model: The reward model is a mission designer's tool for encoding various mission objectives, such as "visit sites A, B, and C under time and energy constraints while avoiding hazards D and E." Thus, a reward function can take many forms and, in general, may be updated based on new information gathered or new objectives. We consider a general formulation that penalizes the time and energy required for each hop, incentivizes a set of n_g goal regions, and penalizes a set of n_h hazardous regions. In summary, the state, action and reward models we consider here are:

$$\begin{aligned} &s \in S^2, \quad a \in \mathcal{A}(s) \subset \mathbb{R}^3, \\ &R(s,a) = -\mathbb{E}[\tau(s,a)]/\tau_{\max} - \lambda_u \mathbb{E}[u(s,a)], \quad R(s_{g_i}) = r_i, \quad R(s_{h_j}) = r_j, \end{aligned} \tag{8}$$

where $\mathbb{E}[\tau(s,a)]$ and $\mathbb{E}[u(s,a)]$ are the expected time and energy required to execute action a at state s. States $s_{g_i} \in S_{g_i}$ and $s_{h_j} \in S_{h_j}$ are states within the goal and hazard regions, with associated rewards $r_i > 0$ and $r_j < 0$, respectively. $\tau_{\max}$ is a maximum travel time, and λ_u weights the control effort.

4.2 Reinforcement Learning Method

The transition model, T, is unknown. In the case of minimal bouncing, approximations of the dynamics such as those discussed in Sect. 3.2 may work, but in general, a series of elastic bounces (e.g., Fig 5) induces chaotic, highly non-Gaussian, multimodal transition dynamics. Without discretization of the state or action spaces, T is extremely difficult to approximate. Accordingly, model-free methods are better suited for this domain, whereby simulations (discussed in Sect. 2) can be used to generate large sets of transition data offline.

One popular technique for such batch, off-line, off-policy, model-free RL is Least Squares Policy Iteration (LSPI), which, as originally described in [24], uses a linear function approximator for the Q-function, and an exact, fixed-point projection for policy evaluation (LSTD-Q):

$$\begin{aligned} &\text{solve: } (\hat{A}_n - \gamma\hat{B}_n)\theta = \hat{b}_n, \quad \text{where} \qquad \hat{b}_n = \frac{1}{n}\sum_{i=1}^{n} \rho_i \left[\phi(s_i,a_i) r_i\right], \\ &\hat{A}_n - \gamma\hat{B}_n = \frac{1}{n}\sum_{i=1}^{n} \rho_i \phi(s_i,a_i) \left[\phi^T(s_i,a_i) - \gamma\phi^T(s_i', \pi(s_i'))\right], \end{aligned} \tag{9}$$

$$\pi(s) = \underset{a\in\mathcal{A}(s)}{\operatorname{argmax}} \left[\phi^T(s,a)\theta\right]. \tag{10}$$

Here, $\phi(s,a)$ is the state-action feature vector, and ρ is a weight vector that sums to one. Like most approximate RL methods, LSPI is not guaranteed to yield optimal policies, but it is a stable algorithm. That is, it will either converge or it will oscillate in an area of the policy space where policies have suboptimality bounded by some approximation error, ε, which is highly dependent on the richness of the feature space and the coverage of the sampling distribution [24]. In practice, this bound is often quite conservative and LSPI typically converges in very few iterations.

One of the most important features of LSPI for deriving hopping policies is its amenability to off-policy exploration strategies, which provides the ability to reuse large data sets, and thus, relearn a policy for a new reward structure on the fly. The weight vector, ρ, provides a convenient way to preferentially bias previously collected samples via *importance weighting* (i.e., $\rho_i = p(s_i,a_i)/q(s_i,a_i)$, where p is the desired distribution and q is the sample distribution); q may be approximated directly from samples via kernel density estimation, and p may be chosen arbitrarily (e.g., a uniform distribution).

4.3 Practical Considerations

Data Collection: An off-line simulator provides large flexibility for data collection. For a given initial state, s_0, and policy, π, state-action samples can be biased towards more likely regions (e.g., through direct Monte Carlo sampling, or importance sampling/variance reduction techniques). However, in the more general case when s_0 is uncertain, or the reward structure may change (and thus, π), we would like a good fit of $\hat{Q}$ over a much *broader* range of the state-action space. Thus, a mostly "pure exploration" strategy is preferred, with a combination of full episode rollouts and periodic restarts, perhaps with some bias towards "hard-to-reach" regions.

Feature Engineering: Linear function approximation relies on a rich set of features over the state-action space to produce good estimates for the Q-function (i.e., small $||\phi^T(s,a)\theta^* - Q^*||_2$). At the same time, the feature set must be amenable for computing the argmax in Eq. (10) for policy extraction. Accordingly, we decouple state and action features such that $\phi(s,a) = \phi_s(s)\phi_a(a)$, where ϕ_s can be arbitrarily complex in the state while ϕ_a remains "simple" enough for optimization. We construct ϕ_s from a set of "hand-crafted" features that leverage domain knowledge (e.g. local geopotential and surface slope) and a set of distributed basis functions—namely, a Fourier basis (similar to [25]), which is more naturally suited to spherical state domains than say, polynomials or RBFs. The action features are simple monomials of the form $\phi_a(\mathbf{v}) = v_x^i v_y^j v_z^k$, where $i+j+k \leq m$, such that a polynomial root solver can compute all local minima exactly.

4.4 Evaluation of Learned Policies

As a preliminary case study, we consider a notional mission scenario on Asteroid Itokawa in which the hopper must reach a target location in minimum time. The reward model (Eq. (8)) is defined as $R(s_g) = +1$ and $R(s,a,s') = -\tau/\tau_{\max}$, where $\tau_{\max} = 10$ hrs, and $\gamma = 1$[1]. Approximately five million trajectories were simulated over a broad range of the state-action space, sampling from a Gaussian distribution on the gravity field ($\sigma_g \sim 5\%$) and a rebound distribution as discussed in Sect. 2.2, with a mean restitution of 0.65. Uniformly random mini-batches of size 100,000 were used in each iteration of LSPI. The action space, $\mathcal{A}(s)$, is speed constrained by $v_{\max} = 10$ cm/s and direction constrained by $40^o \leq \beta \leq 50^o$, where the elevation angle, β, must lie within an annular cone about the local surface normal (in accordance with the hopping constraints for Hedgehog, derived in [8]). With this problem definition, LSPI was able to converge to a small error, $||\theta_i - \theta_{i-1}|| < \varepsilon$, in only a few tens of iterations and within tens of minutes on a laptop (though, significant speedups may be achieved with a more efficient implementation).

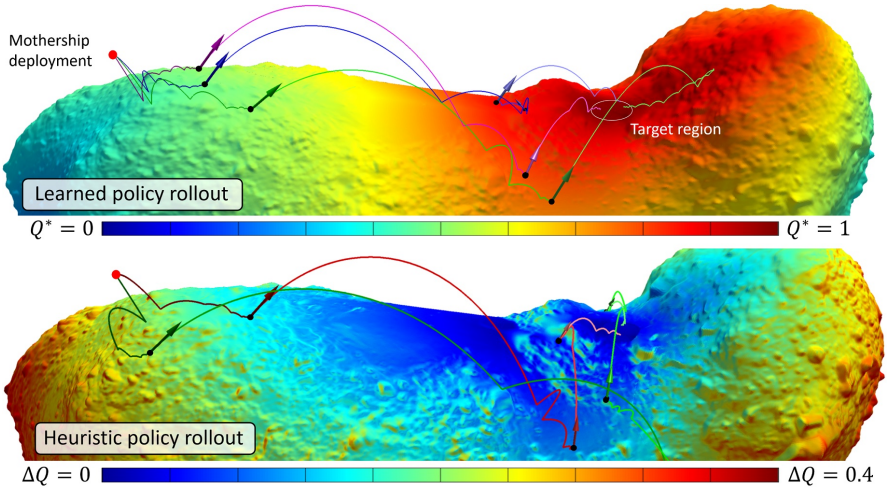


Fig. 8: **Top:** Three rollouts of the learned policy. The surface color map shows the optimal value function under π^*. **Bottom:** Two rollouts of the "hop-to-the-goal" heuristic policy, where the color map shows the *difference* between the learned and heuristic value functions ($\Delta Q = Q^{\pi^*} - Q^{\pi_h}$).

Figure 8 shows a few example trajectories comparing the performance of the learned policy, π^*, with a "hop-towards-the-goal" heuristic policy, π_h, that attempts to take the most direct path to the goal. From 1000 policy rollouts, the mean time to reach the goal from deployment was 5.1 hours for the learned policy, and 7.6 hours for the heuristic policy. The color map in the top figure shows the optimal value, Q^*, at each point on the surface, and due to the reward structure defined above, it also represents the expected time to reach the goal (as a function of $\tau_{\max}$). Not surprisingly, Q^* decays away from the goal region. The color map in the bottom figure

[1] Non-discounting is stable since the reward is always negative and each episode must terminate.

shows the value *margin* of the optimal policy over the heuristic policy, suggesting that the heuristic policy has difficulties on sloped surfaces and states farther from the goal. The second hop of the blue trajectory in the top figure illustrates how the learned policy enables the hopper to perform local adjustments to better position itself for future hops—in this case, by performing a small backwards hop off of a sloped region. The last hop of the green trajectory highlights another interesting learned behavior: the rover hops uphill from the goal region, "understanding" that it is likely to tumble downhill, thereby *leveraging* the dynamics of the environment without ever having explicitly learned a model.

An important consideration when learning in simulation and executing in the real world is robustness to modeling errors. This "transfer learning" problem for asteroid environments may have three major types of modeling errors: (1) the asteroid's shape, (2) its surface properties, and (3) its density/gravity. While this learning method does require an accurate shape model at a global scale, it is insensitive to smaller scale variations due to the global-frame representation of the action space. That is, unanticipated deviations in local topography only affect the constraints for policy extraction, not the optimal value function itself. As a preliminary study of robustness to contact modeling errors, we rolled out the learned policy in an environment with a different contact model—specifically, one with higher surface elasticity and one with lower elasticity. For the *more* elastic case, 1000 policy rollouts exhibited significantly longer traverse times, with a mean of 8.2 hours. These trajectories often bounce off course or overshoot the goal, requiring major corrections. On the other hand, policy rollouts in an environment with *reduced* elasticity actually exhibit better performance, with a mean traverse time of only 4.7 hours. This result suggests that one should err on the side of overestimating the surface elasticity for simulations. Finally, although density/gravity models are likely to be fairly accurate from preliminary surveying by the mothership, simulations can use a conservative *under*approximation of gravity for safe policy transfer (i.e., so that the hopper does not overshoot it's target or reach escape velocity).

5 Conclusions

In this paper, we presented an uncertainty-aware approach to motion planning for hopping rovers on small Solar System bodies. We first examined the problem of planning a single hopping trajectory using a model-based approach, computing exact solutions for impact targeting, propagating uncertainty, and deriving optimal hops from a myopic policy gradient. We then cast the sequential planning problem as an MDP, proposed an offline, off-policy, model-free RL method, and evaluated learned policies against heuristic policies and robustness to modeling errors.

This paper leaves numerous important extensions open for further study. First, we would like to explore various additional mission scenarios that include expert-defined hazards (e.g. pits and caves), multiple target sites, and constraints on mothership communication and solar recharge. Localization errors, or state uncertainty, is another important aspect to consider. Future work will consider extensions of this approach that are amenable to partial observability (POMDPs). Finally, learned policies should be validated in a higher fidelity simulation environment that captures cm-scale contact interactions as well as simulated visual odometry.

References

1. J. Castillo, M. Pavone, I. Nesnas, and J. A. Hoffman. Expected science return of spatially-extended in-situ exploration at small Solar System bodies. In *IEEE Aerospace Conference*, 2012.
2. R. Ambrose, I. A. D. Nesnas, F. Chandler, B. D. Allen, T. Fong, L. Matthies, and R. Mueller. 2015 NASA technology roadmaps: TA 4: Robotics and autonomous systems. Technical report, NASA, 2015.
3. C. Dietze, F. Herrmann, S. Kuß, C. Lange, M. Scharringhausen, L. Witte, T. van Zoest, and H. Yano. Landing and mobility concept for the small asteroid lander MASCOT on asteroid 1999 JU3. In *Int. Astronautical Congress*, 2010.
4. T. Yoshimitsu, T. Kubota, I. Nakatani, T. Adachi, and H. Saito. Micro-hopping robot for asteroid exploration. *Acta Astronautica*, 52(2–6):441–446, 2003.
5. A. F. J. Abercromby, M. L. Gernhardt, S. P. Chappell, D. E. Lee, and A. S. Howe. Human exploration of phobos. In *IEEE Aerospace Conference*, 2015.
6. R. Allen, M. Pavone, C. McQuin, Issa Nesnas, Julie C. Castillo-Rogez, Tam-Nguyen Nguyen, and Jeffrey A. Hoffman. Internally-actuated rovers for all-access surface mobility: Theory and experimentation. In *Proc. IEEE Conf. on Robotics and Automation*, 2013.
7. R. G. Reid, L. Roveda, I. A. D. Nesnas, and M. Pavone. Contact dynamics of internally-actuated platforms for the exploration of small Solar System bodies. In *i-SAIRAS*, 2014.
8. B. Hockman, A. Frick, I. A. D. Nesnas, and M. Pavone. Design, control, and experimentation of internally-actuated rovers for the exploration of low-gravity planetary bodies. *Journal of Field Robotics*, 2016.
9. B. Hockman, R. G. Reid, I. A. D. Nesnas, and M. Pavone. Experimental methods for mobility and surface operations of microgravity robots. In *Int. Symp. on Experimental Robotics*, 2016.
10. M. Bajracharya, M. W. Maimone, and D. Helmick. Autonomy for mars rovers: Past, present, and future. *IEEE Computer*, 41(12):44–50, 2008.
11. S. Higo, I. Nakatani, and T. Yoshimitsu. Localization over small body surface by radio ranging. In *Proceedings of Space Sciences and Technology Conferencs*, 2005.
12. E. W. Y. So, T. Yoshimitsu, and T. Kubota. Relative localization of a hopping rover on an asteroid surface using optical flow. In *SICE Anual Conference*, 2008.
13. D. J. Scheeres. Orbit mechanics about asteroids and comets. *AIAA Journal of Guidance, Control, and Dynamics*, 35(3):987–997, 2012.
14. S. Tardivel, D. J. Scheeres, P. Michel, S. Van wal, and P. Sánchez. Contact motion on surface of asteroid. *AIAA Journal of Spacecraft and Rockets*, 51(6):1857–1871, 2014.
15. S. Van wal, S. Tardivel, and D. J. Scheeres. High-fidelity small body lander simulations. In *Int. Conf. on Astrodynamics Tools and Techniques*, 2016.
16. J. Bellerose and D. J. Scheeres. Dynamics and control for surface exploration of small bodies. In *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, 2008.
17. A. Klesh, J. Bellerose, and T. Kubota. Guidance and control of hoppers for small body exploration. In *Int. Astronautical Congress*, 2010.
18. E. Hand. Philae probe makes bumpy touchdown on a comet. *Science*, 346(6212):900–901, 2014.
19. I. Sharma, J. A. Burns, and C.-Y. Hui. Nutational damping times in solids of revolution. *Monthly Notices of the Royal Astronomical Society*, 359(1):79–92, 2005.
20. R. A. Werner and D. J. Scheeres. Exterior gravitation of a polyhedron derived and compared with harmonic and mascon gravitation representations of asteroid 4769 castalia. *Celestial Mechanics and Dynamical Astronomy*, 65(3):313–344, 1996.
21. N. Murdoch, I. A. Martinez, C. Sunday, E. Zenou, O. Cherrier, A. Cadu, and Y. Gourinat. An experimental study of low-velocity impacts into granular material in reduced gravity. *Monthly Notices of the Royal Astronomical Society*, 468(2):1259–1272, 2017.
22. R. H. Gooding. A procedure for the solution of lambert's orbital boundary-value problem. *Celestial Mechanics and Dynamical Astronomy*, 48(2):145–165, 1990.
23. R. M. Woollands, A. B. Younes, and J. L. Junkins. New solutions for the perturbed lambert problem using regularization and picard iteration. *AIAA Journal of Guidance, Control, and Dynamics*, 38(9):1548–1562, 2015.
24. M. G. Lagoudakis and R. Parr. Least-squares policy iteration. *Journal of Machine Learning Research*, 4(Dec):1107–1149, 2003.
25. G. Konidaris, S. Osentoski, and P. Thomas. Value function approximation in reinforcement learning using the Fourier basis. In *Proc. AAAI Conf. on Artificial Intelligence*, 2011.