Robust Entry Vehicle Guidance with Sampling-Based Invariant Funnels

Remy Derollez
Stanford University
Department of Aeronautics and Astronautics
496 Lomita Mall
Stanford CA 94305
derollez@stanford.edu

Simon Le Cleac'h
Stanford University
Department of Mechanical Engineering
496 Lomita Mall
Stanford CA 94305
simonlc@stanford.edu

Zachary Manchester
Carnegie Mellon University
Robotics Institute
5000 Forbes Ave
Pittsburgh, PA 15213
zacm@cmu.edu

Abstract—**Managing uncertainty is a fundamental and critical issue in spacecraft entry guidance. This paper presents a novel approach for uncertainty propagation during entry, descent and landing that relies on a new sum-of-squares robust verification technique. Unlike risk-based and probabilistic approaches, our technique does not rely on any probabilistic assumptions. It uses a set-based description to bound uncertainties and disturbances like vehicle and atmospheric parameters and winds. The approach leverages a recently developed sampling-based version of sum-of-squares programming to compute regions of finite time invariance, commonly refered to as "invariant funnels". We apply this approach to a three-degree-of-freedom entry vehicle model and test it using a Mars Science Laboratory reference trajectory. We compute tight approximations of robust invariant funnels that are guaranteed to reach a goal region with increased landing accuracy while respecting realistic thermal constraints.**

TABLE OF CONTENTS

1. INTRODUCTION

The future ambitions of multiple space agencies, including NASA, target the Moon and Mars as future destinations. Robotic missions such as Mars 2020 extend our knowledge of the red planet incrementally by performing increasingly complex scientific experiments and in-situ exploration tasks. Ultimately, robotic missions also pave the way for crewed missions to Mars that are studied and planned both by national space agencies and private companies. The Entry Descent and Landing (EDL) phase is a crucial part of all missions that require landing on another celestial body.
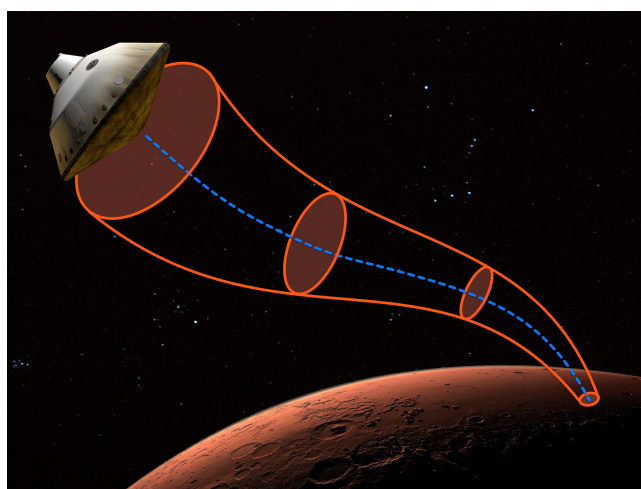
Figure 1. Mars entry funnel concept. The funnel provides conservative bounds on the maximum deviation of the closed-loop system from the nominal trajectory in the presence of disturbances and model uncertainty.

Past missions to Mars were primarily concerned with safe landing, paying little respect to the value of the landing site [1]. As ambitions grow, mission and specifically landing requirements reach new levels. New missions seek to land larger and more complex payloads on the surface of Mars. Improved landing accuracy is also desired, whether it is to achieve pinpoint landing at a scientifically relevant site, land near pre-positioned assets, or ensure the proximity to an established human colony on Mars. It is currently envisioned that future crewed missions to our neighbour planet will likely require safe landing of twenty metric tons mass with an accuracy better than one hundred meters.

Landing on Mars is a hard problem that encompasses major challenges [2]. Mars' atmosphere is thin enough to affect vehicles thermally [3] in a critical way, but does not provide the same braking as Earth's atmosphere. This situation still prevents missions landing at higher altitude points. In addition, knowledge of the planetary environment, atmospheric parameters, and winds are all imperfect. Uncertainties on position, velocity, and aerodynamic properties of the spacecraft also still represent a substantial issue [4].

Many of the entry guidance strategies currently in use still rely heavily on methods developed for the Viking missions in the 70's [2, 5]. Until recently, most landers had flown an unguided ballistic atmospheric entry. An integrated closed-loop guidance and control strategy was used for the Mars Science Laboratory (MSL) and Curiosity missions. Recently algorithms and approaches leveraging optimization, optimal control, and adaptive techniques [6] have emerged to address the entry trajectory optimization, guidance, and control problem. Predictor-corrector methods have been developed and tested for better performance [1]. Li uses direct collocation and nonlinear programming to achieve increased precision during entry [7]. The work of Benito expands on landing footprints [8] using the concepts of reachable and controllable sets [9]. Particle-swarm optimization approaches have been used to tackle the re-entry problem [10]. Steinfeldt's work leverages the concept of a state-dependent Riccati equation and applies it to atmospheric entry guidance [11]. Finally, machine learning and neural networks have also begun to be applied to the topic [12].

On top of traditional entry guidance techniques, uncertainty analysis [13] is usually performed using Monte Carlo methods [14], which are computationally expensive. Polynomial Chaos Expansion (PCE) [15] methods have also been applied to hypersonic flight dynamics [16, 17], but do not scale well to the higher dimensions needed to model realistic problems. More recently, Jin and al. applied linear covariance techniques to the entry problem on a three-degree-of-freedom model and compared the results to a Monte Carlo analysis [18]. Woffinden and al. went further to apply similar linear covariance analysis techniques to atmospheric flight during the EDL phase, but this time on a full six-degree-of-freedom dynamical model to generate navigation system requirements for the Safe and Precise Landing – Integrated Capabilities Evolution (SPLICE) project [19]. While these new approaches are usually faster than traditional Monte Carlo analyses, some require strong assumptions on the probability distributions of the state and/or parameters, which are difficult to infer given our limited knowledge of the Mars environment. Many techniques are also difficult to integrate in an optimization framework. Following those observations, the authors previously investigated a non-probabilistic, scalable alternative using sampling and convex optimization [17].

In parallel, robust motion planning tools and systems verification techniques have been developed in the robotics community with the goal to obtain strong guarantees on performance and safety of robotic systems. In particular, some set-based uncertainty propagation and verification approaches rely on Sum-Of-Squares (SOS) programming for the computation of regions around nominal trajectories in which the system under study is guaranteed to be at all time given some bounded sets for the uncertainties and disturbances. These regions are referred to as regions of finite-time invariance (or "invariant funnels") [20, 21, 22]. Even if we cannot deny the power of these approaches, their tractability has usually been an issue for application to real-life systems.

It is the authors' belief that Mars entry trajectory planning and robust uncertainty propagation are important prerequisites for the implementation and integration of relevant guidance and control strategies. Improvement in these domains should allow us to reach the safety and accuracy goals driven by future missions' requirements. In that context, this paper presents a novel approach for uncertainty propagation during EDL that relies on a recently developed sum-of-squares robust verification technique. Combining the recent work from Shen and Tedrake [23] and the concept of invariant funnels, a sampling-based SOS approach is leveraged to compute regions of finite-time invariance for the Mars entry guidance problem. The resulting technique does not rely on any probabilistic assumptions. The funnels we compute provide strong guarantees on safety and accuracy, while substantially reducing computational burden. Figure 1 shows a conceptual view of the entry funnel.

Our primary contributions include:

- The introduction of sum-of-squares verification techniques for robust motion planning and uncertainty propagation to EDL applications.

- The presentation of a novel sampling-based SOS funnel computation approach providing safety and performance guarantees.

- The application of this new technique to the Mars entry guidance problem and the demonstration of its performance compared to traditional methods.

- The development of the open-source SystemVerification.jl software toolbox to compute invariant funnels for dynamical systems using both traditional and sampling-based approaches.

This paper is organized as follows: Section 2 provides some background on sum-of-squares programming and regions of finite-time invariance. Using these concepts, Section 3 details the sampling-based SOS approach for funnel computation at the core of this work. Section 4 presents the results of the approach applied to the Mars entry guidance problem. Finally Section 5 provides some analysis on the results and ends with some ways forward.

2. BACKGROUND

Sum-of-Squares Programming

Sum-of-squares methods have been developed in the robotics community over the past decade to enable control of complex mechanical systems with guarantees on safety, robustness, and performance in the presence of disturbances and uncertainty. SOS programming relies on the notion of sum-of-squares polynomials. A polynomial is said to be a sum-of-squares if it can be written as a sum of squared polynomial terms. The set of SOS polynomials up to a certain degree k is denoted $\Sigma_k[x]$. By definition,

$$P \in \Sigma_k[x] \iff P(x) = \sum_{j=1}^{n} g_j^2(x), \qquad (1)$$

for some polynomials $g_j(x)$. If a polynomial P is SOS, its expression in terms of its associated $g_i(x)$ polynomials is called its SOS decomposition. Note that this decomposition is not unique in general.

The set $\Sigma_k[x]$ is a proper convex cone. As a consequence, general tools from convex optimization and conic programming can be directly applied to verify SOS conditions and decomposition [24, 25]. Recently, the authors in [26] have suggested the use of a primal-dual interior point method to solve SOS programs in a direct manner. Noting that the sum-of-squares cone is not symmetric and that currently no logarithmically homogeneous self-concordant barrier function is known for the primal cone, the authors leverage a non-

symmetric conic solver. Ahmadi et. al. have also introduced the concepts of DSOS and SDSOS [27] that rely on linear and second-order-cone programming (LP & SOCP) as approximations to sum-of-squares and semi-definite optimization. Up to now though, the widely-used alternative for dealing with SOS programs remains the exploitation of a simple, yet powerful link between SOS and Semi-Definite Programming (SDP) that can be summarized as,

$$P \in \Sigma_{2k}[x] \iff P(x) = Z(x)^T Q Z(x), \ Q \in S_k^+, \quad (2)$$

where $Z(x)$ is a vector of monomials in variable x up to degree k and S_k^+ is the set of positive semi-definite matrices of dimension k. Using this equivalence, certifying that a polynomial is SOS amounts to solving a positive-semi-definite feasibility program (i.e. finding a positive-semi-definite matrix, Q, satisfying (2)).

Sum-of-squares conditions are usually used as substitute conditions for non-negativity. While certifying non-negativity of a polynomial is NP-hard, sum-of-squares certificates can be obtained much more easily using semi-definite programming. Consequently, if a SOS certificate is found, non-negativity is obtained automatically. Note that the converse is not true in general. In addition, these techniques can be used even in the case of non-polynomial dynamical systems by performing a Taylor expansion on the dynamics function up to the desired degree.

Note that we can extend the previous definitions and results to consider the general case of SOS certificates on semi-algebraic sets. For a more in depth treatment of SOS programming and its application to dynamics and control [28] we refer the interested reader to the work of Parrilo [29].

Lyapunov Stability Analysis

Lyapunov analysis is concerned with showing convergence or stability properties of dynamical systems. Usually, the objective is to demonstrate the ability of a given controller to drive the system to a goal state, x_g, which can be assumed to be the origin of the system without loss of generality. The system is governed by a dynamical equation and a feedback controller,

$$\dot{x} = g(x, u), \quad (3)$$
$$u = h(x). \quad (4)$$

Where $x \in R^n$, is the state of the system and $u \in R^m$, is the control input. The composition of the controller with the system dynamics provides an autonomous closed-loop dynamics equation dictating the evolution of the system in time,

$$\dot{x} = f(x). \quad (5)$$

A Lyapunov function, $V(x)$ is a positive function of the state of the system. This function is continuously differentiable over, D, an open subset of R^n containing the origin. Moreover, it respects the following global properties,

$$V(x) > 0, \qquad \forall x \in D \setminus \{0\} \quad (6)$$
$$V(0) = 0, \quad (7)$$
$$\dot{V}(x) = \frac{\partial V}{\partial x} f(x) < 0, \qquad \forall x \in D \setminus \{0\} \quad (8)$$
$$\dot{V}(0) = 0. \quad (9)$$

We encapsulate the Equations 6 - 9 with the following notations; $V \succ 0$ and $\dot{V} \prec 0$.

V is typically an energy-like function decreasing over time as the system evolves under its controlled dynamics. This Lyapunov function is used to demonstrate long-term behavior of the controlled system. In the literature, these techniques have been extended to non-autonomous dynamical systems as well.

Region of Attraction

A region-of-attraction, G, is a subset of R^n for which the following property holds,

$$x(0) \in G \implies \lim_{t \to \infty} ||x(t) - x_g|| = 0. \quad (10)$$

Intuitively, this means that if the dynamical system starts inside G, it will converge to the goal state, x_g, as time t goes to infinity. Sum-of-squares techniques are the basis for a large number of algorithms estimating regions of attractions for non-linear systems [30].

Region of Finite-Time Invariance

It is possible to extend the notion of region of attraction in a timely manner to deal with regions of finite-time invariance, also known as invariant "funnels". As described in the work of Majumdar [31, 21], a funnel can be seen as a time-varying generalization of a region of attraction. Loosely, an invariant funnel can be thought of as a tube around a reference trajectory within which a tracking controller is guaranteed to stabilize the system. Funnels have attracted widespread interest in the robotic motion planning community in recent years and have been used to solve a variety of challenging control problems.
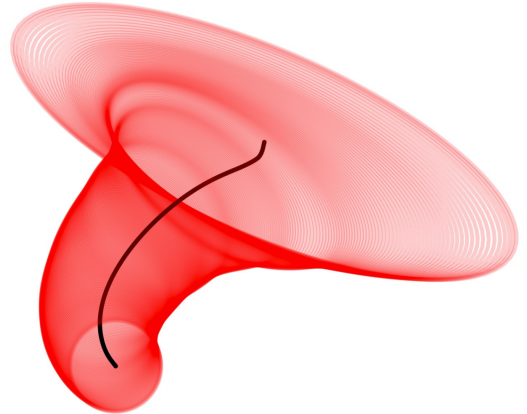

Figure 2. Three-dimensional invariant funnel computed around a nominal reference trajectory for a Dubins car.

Let's assume we are given a set of initial conditions S_1. We are interested in the set of goals that the system can evolve to at time $t \in [0, T]$. Mathematically we can define a funnel as a map F,

$$F : [0, T] \to P(R^n) \quad (11)$$
$$F : t \mapsto F(t) \quad (12)$$

3

where the set $F(t)$ satisfies

$$\bar{x}(0) \in M_1 \implies \bar{x}(t) \in F(t) \; \forall \, t \in [0, T] \qquad (13)$$

where $\bar{x}(t) = x(t) - x^*(t)$ is the difference between the current state and the reference state at time t. Just like SOS tools can be applied to compute approximations of regions of attraction for dynamical systems and specific controllers, they can also be used to compute inner and outer approximations of invariant funnels.

In [20] the authors compute approximate funnels around a nominal trajectory by defining $F(t)$ as,

$$F(t) = \{\bar{x}(t) | V(t, \bar{x}(t)) \leq \rho(t)\}, \qquad (14)$$

where ρ is a strictly positive function of time defining the boundary of the funnel and $V(t, \bar{x}(t))$ is a time-dependent Lyapunov function of the system under study. Figure 2 displays such a funnel computed around a nominal trajectory for a Dubins car system computed using the Julia package SystemVerification.jl developed by the authors.

This framework coupled with the SOS approach is a generic environment for computing invariant funnels in the literature [22], and constitute the basis for the work developed in this paper. Other approaches exist that are related to the concept of invariant funnels, but rely on different tools. Tube Model Predictive Control (TMPC) allows for the design of robust controllers maintaining the system in a tube around the nominal trajectory [32, 33]. Reachability analysis is yet another framework following similar goals [34, 35] with a different approach.

3. SAMPLING-BASED SOS FUNNEL

Approach

The authors in [36] develop the theory for using sampling on algebraic varieties to solve SOS programs while reducing computational effort. Building upon this theoretical foundation, Shen and Tedrake apply it for the computation of regions of attraction [23]. The approach presented in this paper is inspired by their work and extends it to the computation of invariant funnels.

Following the traditional funnel computation described in [21], we use $(x^*(t))_t$ to represent the reference trajectory followed by the system and $V(x, t)$ to represent a Lyapunov function of the system evaluated at state x and time t. In the following, we consider a discrete-time version of the funnel. In this sense, we compute an approximation of the funnel by only focusing on "slices" at given time steps. We then reconnect the pieces together using interpolation techniques to obtain the complete funnel. In addition, in this work we focus on ellipsoidal-shaped funnels. In this configuration the Lyapunov functions are quadratic in $\bar{x}$ and we write,

$$V(\bar{x}, t_k) = \frac{1}{2} \bar{x}^T P(t_k) \bar{x}, \qquad (15)$$

to refer to the Lyapunov function V computed at time t_k (corresponding to time step k) and evaluated at $\bar{x}$.

To simplify notation we will write,

$$V_k(\bar{x}) = V(\bar{x}, t_k) = V(x - x_k^*, t_k) \qquad (16)$$
$$x_k = x(t_k) \qquad (17)$$

A slice of the funnel at a given time step t_k is a mathematical set that can be written as,

$$F_k = F(t_k) = \{x | \frac{1}{2}(x - x_k^*)^T P_k (x - x_k^*) \leq \rho_k\} \qquad (18)$$

where $\rho_k = \rho(t_k)$ is a scalar that concretely represents the limit of the funnel around the nominal trajectory at time step k. Equation (18) is the discrete-time equivalent of Equation (14). Note that, at this, stage P_k and x_k^* are known. Therefore the funnel is parameterized only by ρ.

We consider computing funnels in a forward manner. Using our discrete definition, we can compute the volume of the funnel with,

$$S = \sum_{k=1}^{N} \rho_k^n \det(P_k)^{-1}, \qquad (19)$$

which basically corresponds to the sum of the volumes of the ellipsoidal slices in time. The quantity S represents the objective of our optimization problem.

Let's now focus on the definition of the constraints of the optimization problem that we are trying to solve. We can distinguish two different types of constraints: The first one corresponds to the constraint at the initial time step that is needed to start the computation of the funnel. We require that the initial region of the state-space in which the system can be found be included in the first slice of the funnel parameterized by the positive semi-definite matrix P_1. By considering an ellipsoidal initial region S_1 parameterized by the matrix M_1, a simple derivation shows that this constraint can be written,

$$M_1 - \frac{P_1}{\rho_1} \succeq 0, \qquad (20)$$

which is a simple semi-definite constraint allowing us to find ρ_1.

The second type of constraint is the stage constraint that allows us to compute the value of ρ_{k+1} given the value of ρ_k. On the boundary of the funnel at t_k, we require $\dot{\rho}(t_k) > \dot{V}(t_k, \bar{x})$. We denote the boundary of the funnel by B_k where

$$B_k = \{\bar{x} | V(t_k, \bar{x}) = \rho_k\}. \qquad (21)$$

This condition can be expanded to obtain a relation between ρ_{k+1} and ρ_k such that,

$$\frac{\rho_{k+1}}{t_{k+1} - t_k} > \frac{1}{2} \frac{\bar{x}^T P_{k+1} \bar{x}}{t_{k+1} - t_k} + \cdots$$
$$[f_k(\bar{x} + x_k^*) - f_k(x_k^*)]^T P_k \bar{x}, \qquad (22)$$

where f_k is the closed-loop dynamics function of the system evaluated with control inputs from time step k leveraging the linear controller gain K_k obtained with iLQR. The derivation can be found in Appendix A.

We rewrite Equation (22) in the following way for simplicity:

$$D_k(\bar{x}; \rho_{k+1}) > 0. \qquad \forall \, \bar{x} \in B_k \qquad (23)$$

This notation shows that at each time step $k \geq 1$, we can determine ρ_{k+1} by using P_k, P_{k+1} and f_k. It also shows that at each time step, the condition must hold for all values of $\bar{x}$ in B_k.

4

Assuming that the quantity D_k in Equation (23) can be approximated by a polynomial function (using a Taylor expansion to the appropriate order, for instance), and that the set B_k is an algebraic variety (which is always the case given our assumptions), we can relax the non-negativity constraint in Equation (23) to a sum-of-squares constraint. Traditionally, the so-called "multiplier approach" is used to incorporate the constraint on the set in the inequality and solve the problem in a global manner.

This is where our approach diverges: We use the quotient-ring sampling strategy [23, 36] to rewrite the constraint in Equation (23). Indeed, following the theoretical results provided in the references, we can rewrite the SOS relaxed constraint in the following *equivalent* way (provided that the proper assumptions hold),

$$(\bar{x}_k^{(i)T}\bar{x}_k^{(i)})^d D_k(\bar{x}_k^{(i)}) = \tilde{n}(\bar{x}_k^{(i)})^T Q_k \tilde{n}(\bar{x}_k^{(i)}), \qquad (24)$$

where Q_k is a new positive semi-definite variable added to the problem, d is an arbitrary integer exponent and $(\bar{x}_k^{(i)})$ is a family of M_k well chosen samples drawn from B_k and $\tilde{n}$ is a vector containing a polynomial basis. We end up with a semi-definite feasibility constraint (on the variable Q_k) to be verified on a specific set of samples.

To summarize, the optimization problem that we solve can be written,

$$\underset{\rho_1, \ldots, \rho_N, Q_1, \ldots, Q_{N-1}}{\text{minimize}} \quad \sum_{k=1}^{N} \rho_k^n \det(P_k)^{-1}$$

$$\text{subject to} \quad M_1 - \frac{P_1}{\rho_1} \succeq 0,$$

$$\bar{x}_k^{(i)} \in B_k, \qquad \forall k \in 1, N-1,$$

$$(\bar{x}_k^{(i)T}\bar{x}_k^{(i)})^d D_k(\bar{x}_k^{(i)}; \rho_{k+1}),$$

$$= \tilde{n}(\bar{x}_k^{(i)})^T Q_k \tilde{n}(\bar{x}_k^{(i)}),$$

$$\forall \bar{x}_k^{(i)}, \; \forall k \in 1, N-1, \qquad (25)$$

where N is the total number of steps.

One important difference compared to the multiplier approach is that the algebraic variety, B_k, that needs to be sampled depends on the value of ρ_k. Therefore, we need to know this value in order to be able to sample the $\bar{x}_k^{(i)}$ values and obtain a constraint on ρ_{k+1}. This prevents us from a global approach and brings us to consider a sequential "greedy" strategy at each time step to obtain the successive values of ρ_k with ρ_1 given by the initial constraint in Equation (20).

Sampling Strategy

An important step in the approach described above is the sampling of the points on the variety B_k at each time step. Different approaches are described for sampling the points $\bar{x}_k^{(i)}$ on the variety B_k and we refer the reader to the work of Cifuentes [36] for a detailed and rigorous treatment of the topic.

In this paper, we note that we can choose a random point $\tilde{x}^{(i)}$ in R^n, then rescale it to obtain our sample $\bar{x}^{(i)}$, which

belongs to the variety automatically:

$$\bar{x}^{(i)} = \frac{\tilde{x}^{(i)}}{\sqrt{\frac{1}{2}(\tilde{x}^{(i)})^T \frac{P_k}{\rho_k} \tilde{x}^{(i)}}}, \qquad (26)$$

$$1 = \frac{1}{2}(\tilde{x}^{(i)})^T \frac{P_k}{\rho_k} \tilde{x}^{(i)} \implies \bar{x}^{(i)} \in B_k. \qquad (27)$$

Interestingly, the minimum number of sample points necessary to guarantee equivalence between inequalities (23) and (24) can be computed. We implement the technique described in [36] to find that minimum number of sample points and apply it for real vectors in our case. This indicator gives us a threshold on the number of samples to gather on the variety.

When analytical sampling methods are not applicable, we notice that the algebraic varieties $(B_k)_k$ are defined by polynomial equalities only. This amounts to finding a subset of the roots of multivariate polynomials. Root-finding techniques like Newton's method can therefore be applied to sample these sets.

Disturbances

In the formulation described above, we have assumed that an initial state-space region S_1 was given to represent the uncertainty on the state of the system. In a similar way, we would like to be able to handle disturbances and uncertainty on parameters of the dynamical model. Following the same robust approach, we define a vector w to represent the uncertain parameters in the model. We also assume that these uncertain parameters live in an algebraic set given by a quadratic function and parameterized by a matrix U:

$$W = \left\{ w \mid \frac{1}{2}w^T U w \leq 1 \right\}. \qquad (28)$$

To extend the computation of the invariant funnel under the disturbance w, we refer again to the work of Majumdar [21]. The invariant funnel condition can be expressed in continuous time as follows,

$$\dot{\rho}(t) > \dot{V}(t, \bar{x}, w) \qquad \forall \bar{x} \in B(t) \quad \forall t \in [0, T] \quad \forall w \in W, \qquad (29)$$

$$V(t) = \{\bar{x} | V(t, \bar{x}) = \rho(t)\}. \qquad (30)$$

Verifying the condition in Equation (29) on the discretized time steps t_k's, would require verifying the inequality for any $w \in W$ at each step. Again, instead of making use of the multiplier approach, we can adapt the sampling technique to draw samples from the relevant variety.

Intuitively, we want to make sure that, by propagating the dynamics under any feasible disturbance for one time step, we end up below the threshold $\rho(t_{k+1})$. Since the dynamics $f_k(x, w)$ is a polynomial function of all its entries, it is continuous and differentiable with respect to w for any x. The same is true for $\dot{V}_k$. Therefore, using Fermat's theorem, we know that the extremal values obtained for $\dot{V}_k$ are located either on the boundary of W, denoted ∂W, or on points $(\bar{x}, w)$ such that $\frac{\partial \dot{V}_k}{\partial w}(\bar{x}, w) = 0$. More formally, the relevant

disturbances are located in the union of two sets,

$$w^* \in \operatorname{argmax}_w \dot{V}_k(\bar{x}, w) \implies w \in \partial W \cup O(\bar{x}), \quad (31)$$

where we define

$$\partial W = \{w \mid g(w) = 0\}, \quad (32)$$

$$O(\bar{x}) = \left\{w \mid \frac{\partial \dot{V}}{\partial w}(\bar{x}, w) = 0\right\}. \quad (33)$$

We remark that both ∂W and $O(\bar{x})$ are algebraic varieties and, therefore, the sampling procedure for SOS constraints can be applied to these sets as well.

We now describe the sampling strategy taking into account uncertainty parameters and disturbances. First, we sample $\bar{x}$ using the same sampling strategy as in the disturbance-free case. Given this sampled $\bar{x}$, we then chose to sample w either from ∂W or $O(\bar{x})$. We can proceed as follows for the first set:

$$w = \frac{\tilde{w}}{\sqrt{\frac{1}{2}\tilde{w}^T W \tilde{w}}} \implies w \in \partial W, \quad \forall \tilde{w} \in R^p. \quad (34)$$

For the second set, we can derive (see Appendix B):

$$\frac{\partial \dot{V}}{\partial w}(\bar{x}, w) = \frac{\partial}{\partial w}[f_k(\bar{x} + x_k^*, w)]^T P_k \bar{x}. \quad (35)$$

We have access to the polynomial expression of the feedback dynamics f_k and to its derivative with respect to w. Given $\bar{x}$, we can apply a root-finding algorithm, like Newton's method, to find a suitable $w \in O(\bar{x})$. Similarly to what has been described in the Approach Section, we simply need to verify the SOS constraints on a small number of samples, allowing us to drastically reduce the computational burden compared to traditional SOS approaches.

The full procedure is summarized the procedure in Algorithm 1, where **SDP** refers to solving a semi-definite program defined as,

$$\begin{aligned}
\underset{\rho_{k+1}, Q_k}{\text{minimize}} \quad & f(\rho_{k+1}) \\
\text{subject to} \quad & \bar{x}_k^{(i)} \in B_k, \qquad \forall i \in 1, M_k, \\
& \left(\bar{x}_k^{(i)T}\bar{x}_k^{(i)}\right)^d D_k(\bar{x}_k^{(i)}; \rho_{k+1}), \\
& = \tilde{n}(\bar{x}_k^{(i)})^T Q_k \tilde{n}(\bar{x}_k^{(i)}), \\
& \forall \bar{x}_k^{(i)},
\end{aligned} \quad (36)$$

where Q_k is a positive semi-definite matrix and f a simple positive and increasing function. Note that the minimization is related to the forward funnel computation. A backward approach would feature maximization.

4. SIMULATION RESULTS

In this section, we present the results of the sampling funnel approach applied to the entry guidance problem. There are several ways to derive the required inputs (i.e. nominal trajectory, x_k^* and time-dependent Lyapunov function V_k) to the process described in Algorithm 1. Our simulations use

---

Algorithm 1: Sampling SOS Funnel Computation

Input : $(t_k, P_k, x_k^*, u_k^*)_k, M_1, U$
Procedure
Obtain minimum ρ_1 satisfying (20);
for $k = 1$ *to* $N - 1$ **do**
  $(\tilde{x}_k^{(i)})_i \leftarrow$ **SAMPLE STATE**(P_k, ρ_k, x_k^*);
  $(\tilde{w}_k^{(j)})_j \leftarrow$ **SAMPLE DIST**$(P_k, \rho_k, x_k^*, (\tilde{x}_k^{(i)})_i)$;
  $\rho_{k+1} \leftarrow$ **SDP**$(\rho_k, P_k, P_{k+1}, x_k^*, (\tilde{x}_k^{(i)})_i, (\tilde{w}_k^{(j)})_j)$;
end
return $(\rho_k)_k$
End Procedure

ALTRO [37], a fast trajectory optimization solver, to find an optimal reference trajectory and the optimal control sequence to reach a specific target. A time-varying linear-quadratic regulator (TVLQR) is then used to obtain a quadratic value function approximation. The positive semi-definite matrix parameterizing this value function is then used in place of the Lyapunov functions for the funnel computation. The trajectory optimization ensures that thermal constraints are satisfied [4].

We consider a three-degree-of-freedom dynamical model for the entry. We recall Vinh's model below [38, 39]. This representation uses the position vector of the center of mass of the vehicle in spherical coordinates and flight path and heading angles for the direction of the velocity vector,

$$\begin{aligned}
\dot{r} &= V \sin \gamma \\
\dot{\theta} &= \frac{V}{r} \frac{\cos \gamma \cos \psi}{\cos \phi} \\
\dot{\phi} &= \frac{V}{r} \cos \gamma \sin \psi \\
\dot{V} &= -D - g \sin \gamma \qquad\qquad (37) \\
\dot{\gamma} &= \frac{1}{V}[L \cos \sigma - (g - \frac{V^2}{r}) \cos \gamma] + C_\gamma \\
\dot{\psi} &= -\frac{1}{V \cos \gamma}(L \sin \sigma + \frac{V^2}{r} \cos^2 \gamma \cos \psi \tan \phi) + C_\psi,
\end{aligned}$$

where C_γ, C_ψ are the Coriolis terms and L and D are lift and drag forces, respectively.

Using this representation, the state space has dimension 6 and the state vector is written $X = [r, \theta, \phi, V, \gamma, \psi]$ where r, θ, ϕ is the position vector expressed in spherical coordinates with origin at the center of the planet, V is the magnitude of the velocity vector, while γ and ψ are respectively the flight path and heading angle giving the direction of the velocity vector. Vehicle parameters and a nominal trajectory approximately matching the Mars Science Laboratory (MSL) mission were used for this study. The simulation was stopped 10 km above the surface corresponding to parachute deployment.

Uncertainties are taken into account in the state of the spacecraft as well as in the atmospheric parameters. Relevant additive external disturbances are present in all components of the dynamical model modeling position uncertainty and winds. Finally, uncertainty is considered in the atmospheric density parameters. MarsGRAM 2010 is used to infer proper bounds on the parameters of a bilinear exponential density model.
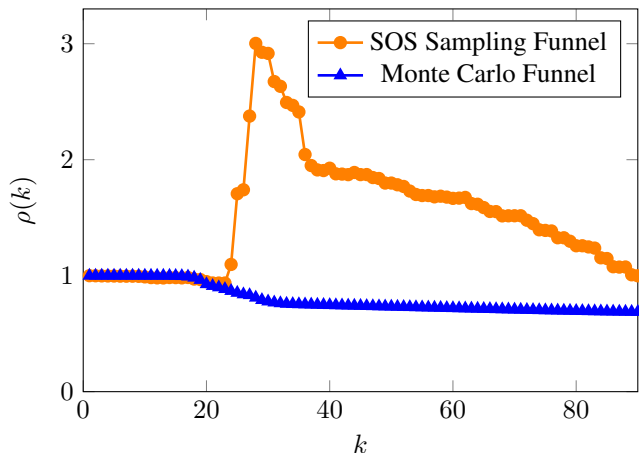
6

Figure 3. Funnel boundary ρ vs time step for both SOS-sampling and Monte Carlo approaches on the entry guidance problem.

Figure 4. Sampled roots location for $k = 7$ on the variety B_k covering the 0-sublevel set of the time derivative of the Lyapunov function V_k.

An open-source Julia package was developed by the authors to perform these funnel computations called SystemVerification.jl. We have made our code available at github.com/RoboticExplorationLab/SystemVerification.jl. Algorithm 1 along with a multiplier-based funnel solver, a Monte Carlo funnel solver and a linear dynamics ellipsoidal funnel solver [40] are implemented in the package. The optimization solver Mosek [41] is used to solve SDPs where needed. At all time steps, the value of the parameter d in Equation (23) is set to 1. Consequently, Q_k is a 45-dimensional square matrix at each time step. When drawing samples on a variety is not straightforward and requires finding roots of a multivariate polynomial (see Section 3), a custom Newton solver is used. For good numerical conditioning, we work with the Hermite polynomial basis for $\tilde{n}$.

Figure 3 presents the value of ρ with respect to time for both the Sampling-SOS approach and a traditional Monte Carlo approach. 10^4 trajectories were generated and propagated for the Monte Carlo run. Figure 4 shows the location of the sampled points (roots of multivariate polynomial) for a 2D slice of the state space along with the 0 sub-level set of the time derivative of the Lyapunov function for a specific time step.

5. DISCUSSION

Figure 3 allows us to validate the approach and to compare it to traditional techniques. A first peak at the beginning of the trajectory is observed corresponding to the vehicle entering the atmosphere, and to the corresponding large deceleration that increases the uncertainty and the volume of the funnel. The invariant funnel also expands towards the end of the trajectory while approaching the ground, where high velocity changes occur. In order to analyze these results properly, it is worth mentioning the major difference that lies in the meaning of these envelopes and that explains the discrepancies observed between our method and Monte Carlo. While traditional approaches provides a result under specific probabilistic assumptions (Gaussian distributions implicitly assumed most of the time), SOS techniques do not assume any particular distribution and instead capture strict bounds on sets. Figure 3 clearly shows that all Monte Carlo sample trajectories are included in the region drawn using
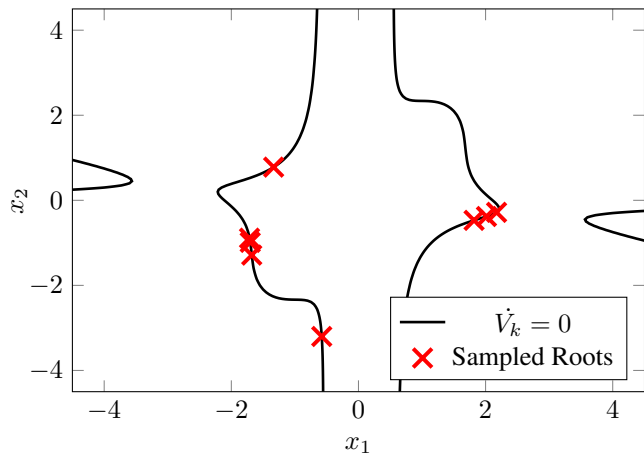
our conservative approach. Monte Carlo solely provides statistical information with no guarantees.

The robustness and non-probabilistic properties also hold in the case of traditional multiplier SOS approaches. However, a multiplier-based SOS approach is prohibitively expensive in the entry vehicle case. Problems with more than 6 or 7 dimensions, including states, controls, and disturbances, are usually intractable with these techniques unless some special separable structure exists in the problem. In comparison, our sampling-based approach verifies equivalent SOS constraints only on a small number of samples. The computation time for our entry funnel example is approximately one minute on a standard laptop computer.

Note that we have assumed quadratic Lyapunov functions in this analysis and have obtained ellipsoidal regions of finite-time invariance in consequence. The theoretical validity of the development presented in Section 3 can be extended to more complex algebraic sets. We have also focused on funnel computation in a forward manner. Backward funnel computation is also possible. A backward version of the sampling-based SOS funnel solver is implemented in SystemVerification.jl.

6. CONCLUSIONS

We have presented a set-based robust approach for uncertainty propagation in entry guidance applications. Our formulation relies on a recent sampling-based sum-of-squares method that is extended to the computation of invariant funnels. The results of this new technique are compared to a classic Monte Carlo approach. The new method does not rely on probabilistic assumptions and provides hard safety and performance guarantees during entry, despite external disturbances and uncertain model parameters, while drastically reducing computational effort. The software developed by the authors has been released in the open-source SystemVerification.jl package for invariant funnel computation.

Directions for future work include extending this analysis to a six-degree-of-freedom entry vehicle model taking into account attitude dynamics. There are also interesting possibilities for combining our invariant funnels with online MPC strategies for improved performance while still maintaining safety guarantees. Finally, given the power and performance

of sampling-based SOS techniques, further developments should also focus on improving computational performance for potential use in online contexts.

APPENDICES

A. STAGE CONSTRAINT DERIVATION

We provide the derivation for the equations used in the sampling version of the funnel:

$$\dot{\rho}(t_k) > \dot{V}(t_k, \bar{x}), \qquad \forall\, \bar{x} \in B_k$$

$$\implies \frac{\rho_{k+1} - \rho_k}{t_{k+1} - t_k} > \frac{V_{k+1}(\bar{x}) - V_k(\bar{x})}{t_{k+1} - t_k} + \left.\frac{\partial \bar{x}}{\partial t}\right|_{t=t_k}^T \frac{\partial V_k}{\partial x}(\bar{x}),$$

$$\implies \frac{\rho_{k+1}}{t_{k+1} - t_k} > \frac{V_{k+1}(\bar{x})}{t_{k+1} - t_k} + \left.\frac{\partial (x - x_k^*)}{\partial t}\right|_{t=t_k}^T \frac{\partial V_k}{\partial x}(\bar{x}),$$

$$\implies \frac{\rho_{k+1}}{t_{k+1} - t_k} > \frac{1}{2}\frac{\bar{x}^T P_{k+1}\bar{x}}{t_{k+1} - t_k} + \ldots$$
$$[f(x, K_k x) - f(x_k^*, K_k x_k^*)]^T P_k \bar{x},$$

$$\implies \frac{\rho_{k+1}}{t_{k+1} - t_k} > \frac{1}{2}\frac{\bar{x}^T P_{k+1}\bar{x}}{t_{k+1} - t_k} + \ldots$$
$$[f_k(\bar{x} + x_k^*) - f_k(x_k^*)]^T P_k \bar{x},$$

$$\implies D_k(\bar{x}; \rho_{k+1}) > 0 \qquad \forall\, \bar{x} \in B_k,$$

where f_k is the closed loop dynamics of the system using the linear control K_k. Note that the inequalities must hold $\forall\, \bar{x} \in B_k$.

B. DISTURBANCES SAMPLING DERIVATION

We present the derivation for the disturbance sampling strategy on the algebraic variety O:

$$\frac{\partial \dot{V}}{\partial w}(\bar{x}, w) = \frac{\partial}{\partial w}\left[\frac{\partial V}{\partial t}(t_k, \bar{x}) + \frac{\partial \bar{x}}{\partial t}(t_k, \bar{x}, w)^T \frac{\partial V}{\partial \bar{x}}(t_k, \bar{x})\right],$$

$$= \frac{\partial}{\partial w}\left[\frac{\partial \bar{x}}{\partial t}(t_k, \bar{x}, w)^T \frac{\partial V}{\partial \bar{x}}(t_k, \bar{x})\right],$$

$$= \frac{\partial}{\partial w}\left[\frac{\partial \bar{x}}{\partial t}(t_k, \bar{x}, w)^T P_k \bar{x}\right],$$

$$= \frac{\partial}{\partial w}\left[\frac{\partial \bar{x}}{\partial t}(t_k, \bar{x}, w)\right]^T P_k \bar{x},$$

$$= \frac{\partial}{\partial w}\left[\frac{\partial x(t)}{\partial t}(t_k, x, w) - \frac{\partial x^*(t)}{\partial t}(t_k, x, 0)\right]^T P_k \bar{x},$$

$$= \frac{\partial}{\partial w}[f_k(x, w) - f_k(x_k^*, 0)]^T P_k \bar{x},$$

$$= \frac{\partial}{\partial w}[f_k(\bar{x} + x_k^*, w) - f_k(x_k^*, 0)]^T P_k \bar{x},$$

$$= \frac{\partial}{\partial w}[f_k(\bar{x} + x_k^*, w)]^T P_k \bar{x}.$$

ACKNOWLEDGMENTS

REFERENCES

[1] S. Li and X. Jiang, "Review and prospect of guidance and control for Mars atmospheric entry," *Progress in Aerospace Sciences*, vol. 69, pp. 40–57, Aug. 2014.

[2] R. Braun, R. Manning, R. Braun, and R. Manning, "Mars Exploration Entry, Descent and Landing Challenges," in *2006 IEEE Aerospace Conference*. Big Sky, MT, USA: IEEE, 2006, pp. 1–18.

[3] M. E. Tauber and K. Sutton, "Stagnation-point radiative heating relations for Earth and Mars entries," *Journal of Spacecraft and Rockets*, vol. 28, no. 1, pp. 40–42, 1991.

[4] D. Bose, J. L. Brown, D. K. Prabhu, P. Gnoffo, C. O. Johnston, and B. Hollis, "Uncertainty Assessment of Hypersonic Aerothermodynamics Prediction Capability," *Journal of Spacecraft and Rockets*, vol. 50, no. 1, pp. 12–18, Jan. 2013.

[5] T. Zang, A. Dwyer Cianciolo, D. Kinney, A. Howard, G. Chen, M. Ivanov, R. Sostaric, and C. Westhelle, "Overview of the nasa entry, descent and landing systems analysis stud," in *AIAA SPACE 2010 Conference & Exposition*, 2010, p. 8649.

[6] A. Roenneke, "Adaptive on-board guidance for entry vehicles," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*. Montreal,Canada: American Institute of Aeronautics and Astronautics, Aug. 2001.

[7] S. Li and Y. Peng, "Mars entry trajectory optimization using DOC and DCNLP," *Advances in Space Research*, vol. 47, no. 3, pp. 440–452, Feb. 2011.

[8] A. Saraf, J. Leavitt, M. Ferch, and K. Mease, "Landing Footprint Computation for Entry Vehicles," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*. Providence, Rhode Island: American Institute of Aeronautics and Astronautics, Aug. 2004.

[9] J. Benito and K. D. Mease, "Reachable and controllable sets for planetary entry and landing," *Journal of guidance, control, and dynamics*, vol. 33, no. 3, pp. 641–654, 2010.

[10] A. Rahimi, K. Dev Kumar, and H. Alighanbari, "Particle Swarm Optimization Applied to Spacecraft Reentry Trajectory," *Journal of Guidance, Control, and Dynamics*, vol. 36, no. 1, pp. 307–310, Jan. 2013.

[11] B. Steinfeldt and P. Tsiotras, "A State-Dependent Riccati Equation Approach to Atmospheric Entry Guidance," in *AIAA Guidance, Navigation, and Control Conference*. Toronto, Ontario, Canada: American Institute of Aeronautics and Astronautics, Aug. 2010.

[12] S. Li and X. Jiang, "RBF neural network based second-order sliding mode guidance for Mars entry under uncertainties," *Aerospace Science and Technology*, vol. 43, pp. 226–235, Jun. 2015.

[13] Y.-z. Luo and Z. Yang, "A review of uncertainty propagation in orbital mechanics," *Progress in Aerospace Sciences*, vol. 89, pp. 23–39, 2017.

[14] M. K. Lockwood, R. W. Powell, C. A. Graves, and G. L. Carman, "Entry system design considerations for Mars landers," 2001.

[15] B. A. Jones, A. Doostan, and G. H. Born, "Nonlinear propagation of orbit uncertainty using non-intrusive polynomial chaos," *Journal of Guidance, Control, and Dynamics*, vol. 36, no. 2, pp. 430–444, 2013.

[16] A. Prabhakar, J. Fisher, and R. Bhattacharya, "Polynomial Chaos-Based Analysis of Probabilistic Uncertainty in Hypersonic Flight Dynamics," *Journal of Guidance, Control, and Dynamics*, vol. 33, no. 1, pp. 222–234, Jan. 2010.

[17] R. Derollez and Z. Manchester, "Sample-based Robust Uncertainty Propagation for Entry Vehicles," 2020.

[18] K. Jin, D. Geller, and J. Luo, "Development and validation of linear covariance analysis tool for atmospheric entry," *Journal of Spacecraft and Rockets*, vol. 56, no. 3, pp. 854–864, 2018.

[19] D. Woffinden, S. Robinson, J. Williams, and Z. R. Putnam, "Linear Covariance Analysis Techniques to Generate Navigation and Sensor Requirements for the Safe and Precise Landing Integrated Capabilities Evolution (SPLICE) Project," in *AIAA Scitech 2019 Forum*. San Diego, California: American Institute of Aeronautics and Astronautics, Jan. 2019.

[20] R. Tedrake, I. R. Manchester, M. Tobenkin, and J. W. Roberts, "LQR-trees: Feedback Motion Planning via Sums-of-Squares Verification," *The International Journal of Robotics Research*, vol. 29, no. 8, pp. 1038–1052, 2010.

[21] A. Majumdar and R. Tedrake, "Funnel Libraries for Real-Time Robust Feedback Motion Planning," *arXiv:1601.04037 [cs, math]*, 2017.

[22] M. M. Tobenkin, I. R. Manchester, and R. Tedrake, "Invariant Funnels around Trajectories using Sum-of-Squares Programming," *arXiv:1010.3013 [math]*, 2010.

[23] S. Shen and R. Tedrake, "Sampling Quotient-Ring Sum-of-Squares Programs for Scalable Verification of Nonlinear Systems," p. 8, 2020.

[24] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed., ser. Springer Series in Operations Research. New York: Springer, 2006.

[25] S. P. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, UK ; New York: Cambridge University Press, 2004.

[26] D. Papp and S. Yıldız, "Sum-of-squares optimization without semidefinite programming," *arXiv:1712.01792 [math]*, 2018.

[27] A. A. Ahmadi and A. Majumdar, "DSOS and SDSOS Optimization: More Tractable Alternatives to Sum of Squares and Semidefinite Optimization," *SIAM Journal on Applied Algebra and Geometry*, vol. 3, no. 2, pp. 193–230, 2019.

[28] S. Singh, M. Chen, S. L. Herbert, C. J. Tomlin, and M. Pavone, "Robust Tracking with Model Mismatch for Fast and Safe Planning: An SOS Optimization Approach," p. 20, 2018.

[29] P. A. Parrilo, "Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization," Ph.D. dissertation, California Institute of Technology, 2000.

[30] F. Meng, D. Wang, P. Yang, G. Xie, and F. Guo, "Application of Sum-of-Squares Method in Estimation of Region of Attraction for Nonlinear Polynomial Systems," *IEEE Access*, vol. 8, pp. 14 234–14 243, 2020.

[31] A. Majumdar, A. A. Ahmadi, and R. Tedrake, "Control design along trajectories with sums of squares programming," in *2013 IEEE International Conference on Robotics and Automation*. Karlsruhe, Germany: IEEE, 2013, pp. 4054–4061.

[32] B. T. Lopez, J.-J. E. Slotine, and J. P. How, "Dynamic Tube MPC for Nonlinear Systems," *arXiv:1907.06553 [cs, eess]*, Jul. 2019.

[33] H. X. Li and B. C. Williams, "Generative Planning for Hybrid Systems based on Flow Tubes," p. 8, 2008.

[34] S. Bansal, M. Chen, S. Herbert, and C. J. Tomlin, "Hamilton-Jacobi Reachability: A Brief Overview and Recent Advances," *arXiv:1709.07523 [cs, math]*, Sep. 2017.

[35] M. Chen, S. Herbert, and C. J. Tomlin, "Exact and Efficient Hamilton-Jacobi-based Guaranteed Safety Analysis via System Decomposition," *arXiv:1609.05248 [math]*, Sep. 2016.

[36] D. Cifuentes and P. A. Parrilo, "Sampling Algebraic Varieties for Sum of Squares Programs," *SIAM Journal on Optimization*, vol. 27, no. 4, pp. 2381–2404, 2017.

[37] T. A. Howell, B. E. Jackson, and Z. Manchester, "ALTRO: A Fast Solver for Constrained Trajectory Optimization," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Macau, China: IEEE, Nov. 2019, pp. 7674–7679.

[38] Vinh, *Hypersonic Planetary Entry Flight Mechanics*, 1980.

[39] J. B. Manrique, "Advances in Spacecraft Atmospheric Reentry Guidance," Ph.D. dissertation, 2010.

[40] Z. Manchester and S. Kuindersma, "DIRTREL: Robust Nonlinear Direct Transcription with Ellipsoidal Disturbances and LQR Feedback," p. 9, 2017.

[41] M. ApS, *MOSEK Optimization Suite. Release 9.1.11.*, 2020.

BIOGRAPHY

Remy Derollez is a Systems Engineer at Loft Orbital and an alumni of the Robotic Exploration Lab. He receives his BS in Engineering from Ecole Centrale Paris in 2017 and his MS in Aerospace Engineering from Stanford University in 2020. His research interests include dynamical systems, robust motion planning, uncertainty propagation, mathematical methods and optimization for space applications and human space exploration.

Simon Le Cleac'h is a graduate student with the Robotic Exploration Lab at Carnegie Mellon University and with the Multi-Robot Systems Lab at Stanford University. He received his BS in Engineering from Ecole Centrale Paris in 2016 and his MS in Mechanical Engineering from Stanford University in 2019. His research interests include real-time motion-planning, game-theoretic optimization, and inverse optimal control.

Zac Manchester is an assistant professor in the Robotics Institute at Carnegie Mellon University and founder of the Robotic Exploration Lab. He received a PhD in aerospace engineering in 2015 and a BS in applied physics in 2009, both from Cornell University. His research interest include control and optimization with application to aerospace and robotic systems with challenging nonlinear dynamics.