Stochastic Motion Planning Using Successive Convexification and Probabilistic Occupancy Functions

Abraham P. Vinod[†1], Sean Rice[†2], Yuanqi Mao[2], Meeko M. K. Oishi[1], and Behçet Açıkmeşe[2]

Abstract— **We propose a method for real-time motion planning in stochastic, dynamic environments via a receding horizon framework that exploits computationally efficient algorithms for forward stochastic reachability analysis and non-convex optimization. Our method constructs a dynamically feasible trajectory for a robot, modeled as an LTI dynamical system, while ensuring 1) a desired probabilistic collision-avoidance guarantee is achieved, 2) state and control constraints are satisfied, and 3) a convex performance objective is minimized. We first compute "keep-out" regions at each time instant to assure a probabilistic collision avoidance guarantee. These keep-out regions are convex and compact, and can be tightly overapproximated by ellipsoids which may be computed in a grid-free, recursion-free, and sample-free manner. The regions are constraints in a non-convex optimization problem, solved via successive convexification. This algorithm uses interior point methods for real-time implementation. We present numerical simulations to demonstrate the efficacy of the approach.**

I. Introduction

Motion planning in the presence of dynamic obstacles has been an active area of research for the past few decades, with applications in robotics, autonomous transportation, and space systems [1]–[5]. In such applications, stochasticity may arise due to modeling uncertainties as well as from sensing inaccuracies. Compared to planning with non-stochastic uncertain obstacle models [6]–[8], motion planning formulations with stochastic obstacle models require that trajectories satisfy probabilistic safety requirements [9]. In this paper, we propose a stochastic motion planner that is grid-free and sample-free, and plans trajectories that are probabilistically safe from rigid-body obstacles with stochastic dynamics. The trajectory also minimizes a cost function and satisfies additional hard (deterministic) state and input constraints.

The existing literature takes two main approaches to the stochastic motion planning problem — 1) via stochastic optimal control theory, or 2) via sampling-based approaches. In the stochastic optimal control approach, a constrained finite-horizon stochastic optimal control problem is formulated with constraints arising due to robot dynamics, bounded robot control authority, and a collection of unsafe sets to be avoided. The resulting optimization problems may be solved either via stochastic reachability techniques [10], [11] or via chance-constrained approaches [4], [12]. The stochastic reachability approach uses dynamic programming to solve the corresponding Markov decision process problem to obtain a trajectory with desired probabilistic safety guarantees [10], [11], [13]. However, such an approach scales poorly with system dimension [13] and is computationally expensive, limiting its application to simple system models and small planning areas. Chance-constrained approaches have been explored for rigid body robots with stochastic dynamics that need to avoid static polytopic obstacles [4]. However, this relies on a mixed-integer programming formulation which scales poorly with increasing number of obstacles and shape complexities.

The alternative is sampling-based approaches [1], [2], [14]–[18]. These methods account for the stochastic behavior of the obstacles by sampling their trajectories. Popular planning techniques using sampling methods include RRT* [16], PRM* [16], and FMT* [17]. Recently, sampling-based planners that use stochastic reachability have also been proposed [15], [18]. Sampling-based methods are highly parallelizable and are applicable to high-dimensional nonlinear dynamics [19]; however, the number of particles required for the desired quality of approximation results in high computational costs [20].

The main contribution of this paper is a stochastic motion planner which is built using: 1) an algorithm to compute an overapproximation of the probability-weighted deterministic keep-out region, 2) a receding horizon control-based solution to the stochastic motion planning problem, and 3) a receding horizon-based optimal controller designed using the successive convexification method. Each keep-out region is associated with a desired likelihood of safety, that is, probabilistic collision avoidance. We provide a computationally efficient algorithm to compute ellipsoidal overapproximations of the keep-out regions using recent results in forward stochastic reachability [21] and probabilistic occupancy functions [22]. We then use known results in risk allocation [12] to convert the given stochastic optimal control problem into a formulation amenable to successive convexification [23]–[25]. Successive convexification allows us to solve the motion planning problem efficiently via convex optimization.

This paper is organized as follows: Section II provides the mathematical preliminaries, an introduction to successive convexification, and sets up the stochastic motion planning problem to be solved. The main contribution of this paper is given in Section III. We provide an algorithm to compute

This material is based upon work supported by the National Science Foundation under Grant Number CMMI-1254990 and IIS-1528047 (Oishi), and CNS-1624328 and CMMI-1613235 (Açıkmeşe). Any recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

[1]Abraham Vinod and Meeko Oishi are with Electrical and Computer Engineering, University of New Mexico, Albuquerque, NM; e-mail: aby.vinod@gmail.com and oishi@unm.edu

[2]Sean Rice, Yuanqi Mao, and Behçet Açıkmeşe are with the Department of Aeronautics and Astronautics, University of Washington, Seattle, WA 98105, USA. Emails: {scrice, yqmao, behcet}@uw.edu

† These authors contributed equally to this work.

Stochastic motion planning problem of interest:

$$\begin{aligned}
\underset{\substack{\overline{x}_{\mathrm{robot}}[1],\overline{x}_{\mathrm{robot}}[2],\ldots,\overline{x}_{\mathrm{robot}}[T],\\ \overline{u}_{\mathrm{robot}}[0],\overline{u}_{\mathrm{robot}}[1],\ldots,\overline{u}_{\mathrm{robot}}[T-1]}}{\text{minimize}} \quad & J\big(\overline{u}_{\mathrm{robot}}[0],\overline{u}_{\mathrm{robot}}[1],\ldots,\overline{u}_{\mathrm{robot}}[T-1],\overline{x}_{\mathrm{robot}}[1],\overline{x}_{\mathrm{robot}}[2],\ldots,\overline{x}_{\mathrm{robot}}[T]\big) && \text{(1a)}\\
\text{subject to} \quad & \overline{x}_{\mathrm{robot}}[t+1] = A_{\mathrm{robot}}\overline{x}_{\mathrm{robot}}[t] + B_{\mathrm{robot}}\overline{u}_{\mathrm{robot}}[t] \quad t\in\mathbb{N}_{[0,T-1]}, && \text{(1b)}\\
& \overline{u}_{\mathrm{robot}}[t] \in \mathcal{U} \quad t\in\mathbb{N}_{[0,T-1]}, && \text{(1c)}\\
& \overline{x}_{\mathrm{robot}}[t] \in \mathrm{SafeSet} \quad t\in\mathbb{N}_{[1,T-1]}, && \text{(1d)}\\
& \overline{x}_{\mathrm{robot}}[T] \in \mathrm{GoalSet}, && \text{(1e)}\\
& \phi_{\boldsymbol{X}_{\mathrm{obs}}}(\overline{x}_{\mathrm{robot}}[t]; t, \overline{X}_{\mathrm{obs}}[0], \overline{r}) < \alpha \quad t\in\mathbb{N}_{[1,T]} && \text{(1f)}
\end{aligned}$$

the probability-weighted keep-out region for each obstacle in Section III-A, formulate the receding horizon control problem corresponding to the stochastic motion planning problem of interest in Section III-B, and adapt successive convexification to solve the receding horizon control problem in Section III-C. We demonstrate the efficacy of our approach with a numerical simulation in Section IV and provide conclusions and future research directions in Section V.

II. Preliminaries and Problem formulation

We denote a discrete-time interval by $\mathbb{N}_{[a,b]}$ for $a, b \in \mathbb{N}$ and $a \leq b$, which inclusively enumerates all integers from a to b. Random variables/vectors are bolded, non-random vectors have an overline, and concatenated random variables/vectors are bolded with an overline. The indicator function of a non-empty set $\mathcal{S}$ is denoted by $1_{\mathcal{S}}(\overline{y})$, such that $1_{\mathcal{S}}(\overline{y}) = 1$ if $\overline{y} \in \mathcal{S}$ and is zero otherwise. We denote the Minkowski sum as $\oplus$, the Cartesian product of the set $\mathcal{S}$ with itself $k \in \mathbb{N}$ times as $\mathcal{S}^k$, the cardinality of $\mathcal{S}$ with $|\mathcal{S}|$, and the Lebesgue measure of a measurable set $\mathcal{S}$ by $\mathrm{m}(\mathcal{S})$. Given $r > 0$ and a vector $\overline{c} \in \mathbb{R}^n$, we denote the Euclidean ball by $\mathrm{Ball}(\overline{c}, r) = \{\overline{y} \in \mathbb{R}^n : \|\overline{y} - \overline{c}\|_2 \leq r\}$.

A. Real analysis and probability theory

A set $\mathcal{S} \subseteq \mathbb{R}^n$ is said to be symmetric if $\mathcal{S} = \{-\overline{y} \in \mathbb{R}^n : \overline{y} \in \mathcal{S}\}$ [26, Sec. 1.1]. Given a convex and compact set $\mathcal{S}$, the support function $\rho(\bar{l}; \mathcal{S}) = \max_{\overline{y}\in\mathcal{S}} \bar{l}^\top \overline{y},\ \bar{l} \in \mathbb{R}^n$ characterizes $\mathcal{S}$ [26, Thm. 5.6.4].

A random vector $\boldsymbol{y}$ is a measurable transformation defined in the probability space $(\mathcal{Y}, \mathscr{S}(\mathcal{Y}), \mathbb{P})$ with sample space $\mathcal{Y} \subseteq \mathbb{R}^p$, sigma-algebra $\mathscr{S}(\mathcal{Y})$, and probability measure over $\mathscr{S}(\mathcal{Y})$, $\mathbb{P}$. When $\boldsymbol{y}$ is absolutely continuous, $\boldsymbol{y}$ has a probability density function (PDF) $\psi_{\boldsymbol{y}}$ such that given $\mathcal{S} \in \mathscr{S}(\mathcal{Y})$, $\mathbb{P}_{\boldsymbol{y}}\{\boldsymbol{y} \in \mathcal{S}\} = \int_{\mathcal{S}} \psi_{\boldsymbol{y}}(\overline{y}) d\overline{y}$ where $\overline{y} \in \mathbb{R}^p$ and $\psi_{\boldsymbol{y}}$ is a non-negative Borel measurable function with $\int_{\mathcal{Y}} \psi_{\boldsymbol{y}}(\overline{y}) d\overline{y} = 1$. From [27, Sec. 2], a probability measure $\mathbb{P}_{\boldsymbol{y}}$ is log-concave if for all convex $\mathcal{S}_A, \mathcal{S}_B \in \mathscr{S}(\mathcal{Y})$ and $\theta \in [0, 1]$, $\mathbb{P}_{\boldsymbol{y}}\{\boldsymbol{y} \in (\theta\mathcal{S}_A \oplus (1-\theta)\mathcal{S}_B)\} \geq \mathbb{P}_{\boldsymbol{y}}\{\boldsymbol{y} \in \mathcal{S}_A\}^\theta \mathbb{P}_{\boldsymbol{y}}\{\boldsymbol{y} \in \mathcal{S}_B\}^{1-\theta}$. See [28], [29] for more details.

B. Stochastic motion planning problem

We consider the motion planning problem over the time interval $\mathbb{N}_{[0,T]}$ with a time horizon $T > 0$. The robot dynamics are assumed to be LTI,

$$\overline{x}_{\mathrm{robot}}[t+1] = A_{\mathrm{robot}}\overline{x}_{\mathrm{robot}}[t] + B_{\mathrm{robot}}\overline{u}_{\mathrm{robot}}[t] \qquad (2)$$

with state $\overline{x}_{\mathrm{robot}}[t] \in \mathcal{X} = \mathbb{R}^n$, input $\overline{u}_{\mathrm{robot}}[t] \in \mathcal{U} \subset \mathbb{R}^m$ ($\mathcal{U}$ is compact), and known matrices $A_{\mathrm{robot}}, B_{\mathrm{robot}}$ and initial state $\overline{x}_{\mathrm{robot}}[0]$. The environment is assumed to have N_{obs} obstacles, and their dynamics are also assumed to be LTI ($j \in \mathbb{N}_{[1,N_{\mathrm{obs}}]}$)

$$\boldsymbol{x}_{\mathrm{obs},j}[t+1] = A_{\mathrm{obs},j}\boldsymbol{x}_{\mathrm{obs},j}[t] + F_{\mathrm{obs},j}\boldsymbol{w}_{\mathrm{obs},j}[t], \qquad (3)$$

with state $\boldsymbol{x}_{\mathrm{obs},j}[t] \in \mathcal{X}$, disturbance $\boldsymbol{w}_{\mathrm{obs},j}[t] \in \mathbb{R}^p$ that has a known probability measure $\mathbb{P}_{\boldsymbol{w}_{\mathrm{obs},j}}$, and known matrices $A_{\mathrm{obs},j}, F_{\mathrm{obs},j}$ and initial state $\overline{x}_{\mathrm{obs},j}[0]$. We use $\boldsymbol{X}_{\mathrm{obs}}[t]$ to describe the obstacle configuration $\boldsymbol{X}_{\mathrm{obs}}[t] = [\boldsymbol{x}_{\mathrm{obs},1}^\top[t]\ \boldsymbol{x}_{\mathrm{obs},2}^\top[t]\ \ldots\ \boldsymbol{x}_{\mathrm{obs},N_{\mathrm{obs}}}^\top[t]]^\top$, with associated probability measure $\mathbb{P}_{\boldsymbol{X}_{\mathrm{obs}}}^{t,\overline{X}_{\mathrm{obs}}[0]}$ parameterized by time t and the initial obstacle configuration $\overline{X}_{\mathrm{obs}}[0] \in \mathcal{X}^{N_{\mathrm{obs}}}$.

Assumption 1. *([1, Sec. 4.3.2]) The rigid bodies (the robot and the obstacles) are restricted to translational motion only.*

Under Assumption 1, the rigid body robot is characterized by its state $\overline{x}_{\mathrm{robot}}[t]$ and a compact rigid body shape $\mathcal{R}(\overline{x}_{\mathrm{robot}}[t]) = \{\overline{x}_{\mathrm{robot}}[t]\} \oplus \mathcal{R}(\overline{0}) \subset \mathcal{X}$. Similarly, each of the rigid body obstacles is characterized by their respective state $\boldsymbol{x}_{\mathrm{obs},j}[t]$ and a compact rigid body shape $\mathcal{O}_j(\boldsymbol{x}_{\mathrm{obs},j}[t]) = \{\boldsymbol{x}_{\mathrm{obs},j}[t]\} \oplus \mathcal{O}_j(\overline{0}) \subset \mathcal{X}$ with $j \in \mathbb{N}_{[1,N_{\mathrm{obs}}]}$. Under Assumption 1, the collision avoidance problem can be equivalently reformulated as requiring the robot, now modeled as a point mass at $\overline{x}_{\mathrm{robot}}[t]$, avoid obstacles with the rigid body shape $\mathcal{O}_j^+(\overline{0}) = \mathcal{O}_j(\overline{0}) \oplus (-\mathcal{R}(\overline{0}))$.

Since $\mathcal{R}(\overline{0})$ and $\mathcal{O}_j(\overline{0})$ are compact, the set $\mathcal{O}_j^+(\overline{0})$ is compact [26, Thm. 1.8.10(v)], and therefore bounded. The boundedness of $\mathcal{O}_j^+(\overline{0})$ allows us to specify a separation distance $r_j > 0$ between the states $\boldsymbol{x}_{\mathrm{obs},j}[t]$ and $\overline{x}_{\mathrm{robot}}[t]$ which guarantees collision avoidance. We define the collection of these separation distances r_j as $\overline{r} = [r_1\ r_2\ \ldots\ r_{N_{\mathrm{obs}}}]^\top \in \mathbb{R}_{>0}^{N_{\mathrm{obs}}}$, the *safe separation vector*. We can conservatively approximate the original collision avoidance problem by requiring the point mass describing the robot to stay out of a union of balls, $\cup_{j=1}^{N_{\mathrm{obs}}} \mathrm{Ball}(\boldsymbol{x}_{\mathrm{obs},j}[t], r_j)$. We can use tighter overapproximations of the rigid body obstacles at the cost of higher computational cost [22].

Probabilistic occupancy functions were introduced in [22] to quantify the collision probability with a rigid body obsta-

Non-stochastic motion planning problem which `SCvx` *and* `SCvx-fast` *can solve*:

$$\begin{aligned}
\underset{\substack{\overline{x}_{\mathrm{robot}}[1],\overline{x}_{\mathrm{robot}}[2],\ldots,\overline{x}_{\mathrm{robot}}[T],\\ \overline{u}_{\mathrm{robot}}[0],\overline{u}_{\mathrm{robot}}[1],\ldots,\overline{u}_{\mathrm{robot}}[T-1]}}{\mathrm{minimize}} \quad & J\big(\overline{u}_{\mathrm{robot}}[0],\overline{u}_{\mathrm{robot}}[1],\ldots,\overline{u}_{\mathrm{robot}}[T-1],\overline{x}_{\mathrm{robot}}[1],\overline{x}_{\mathrm{robot}}[2],\ldots,\overline{x}_{\mathrm{robot}}[T]\big) && \text{(4a)}\\
\text{subject to} \quad & \overline{x}_{\mathrm{robot}}[t+1] = f(\overline{x}_{\mathrm{robot}}[t],\overline{u}_{\mathrm{robot}}[t]) \quad t\in\mathbb{N}_{[0,T-1]}, && \text{(4b)}\\
& \overline{u}_{\mathrm{robot}}[t]\in\mathcal{U} \quad t\in\mathbb{N}_{[0,T-1]}, && \text{(4c)}\\
& \overline{x}_{\mathrm{robot}}[t]\in\mathrm{SafeSet} \quad t\in\mathbb{N}_{[1,T-1]}, && \text{(4d)}\\
& \overline{x}_{\mathrm{robot}}[T]\in\mathrm{GoalSet}, && \text{(4e)}\\
& \overline{x}_{\mathrm{robot}}[t]\notin\bigcup_{j=1}^{N_{\mathrm{obs}}}\mathrm{BadSet}_j \quad t\in\mathbb{N}_{[1,T-1]} && \text{(4f)}
\end{aligned}$$

---

cle that has stochastic dynamics at a time and state of interest. For this problem, we characterize a probabilistic occupancy function for the obstacle configuration, $\phi_{\boldsymbol{X}_{\mathrm{obs}}}:\mathcal{X}\to[0,1]$,

$$\phi_{\boldsymbol{X}_{\mathrm{obs}}}(\overline{y};t,\overline{X}_{\mathrm{obs}}[0],\overline{r}) = \mathbb{P}_{\boldsymbol{X}_{\mathrm{obs}}}^{t,\overline{X}_{\mathrm{obs}}[0]}\left\{\cup_{j=1}^{N_{\mathrm{obs}}}\left\{\boldsymbol{x}_{\mathrm{obs},j}[t]\in\{\overline{z}\in\mathcal{X}:\overline{y}\in\mathrm{Ball}(\overline{z},r_j)\}\right\}\right\} \tag{5}$$

The function $\phi_{\boldsymbol{X}_{\mathrm{obs}}}(\overline{y};t,\overline{X}_{\mathrm{obs}}[0],\overline{r})$ provides the probability that $\cup_{j=1}^{N_{\mathrm{obs}}}\mathrm{Ball}(\boldsymbol{x}_{\mathrm{obs},j}[t],r_j)$ occupies the state of interest $\overline{y}\in\mathcal{X}$ at time t, given the safe separation vector $\overline{r}$, and initial obstacle configuration $\overline{X}_{\mathrm{obs}}[0]$.

The stochastic motion planning problem of interest may be formulated as (1). We seek to minimize a convex cost function $J:\mathcal{U}^T\times\mathcal{X}^T\to\mathbb{R}$ while assuring that 1) the robot stays within a convex and compact safe set $\mathrm{SafeSet}\subseteq\mathcal{X}$ at all times, 2) the robot reaches a convex and compact goal set $\mathrm{GoalSet}\subseteq\mathrm{SafeSet}$ at time T, and 3) the probability of collision of the robot with any of the rigid-body obstacles is below a maximum acceptable probability of collision $\alpha\in(0,1]$ for each instant. We know the initial obstacle configuration $\overline{X}_{\mathrm{obs}}[0]$, the initial robot state $\overline{x}_{\mathrm{robot}}[0]$, the dynamics of the robot (2) and the obstacles (3), and the safe separation vector $\overline{r}\in\mathbb{R}_{>0}^{N_{\mathrm{obs}}}$.

C. Successive convexification for motion planning

We next consider the non-stochastic motion planning problem (4) that can be solved using successive convexification method [23]–[25] with the time horizon as $T>0$. This approach can handle nonlinear discrete-time dynamics for the robot $f:\mathcal{X}\times\mathcal{U}\to\mathcal{X}$ as in (4b). We will refer to the robot trajectory defined for the time interval $\mathbb{N}_{[1,T]}$ as $\overline{x}_{\mathrm{robot}}[\cdot]$. For brevity, we redefine each temporal point in the trajectory as $\overline{x}_t\triangleq\overline{x}_{\mathrm{robot}}[t]\in\mathcal{X},\ \forall t\in\mathbb{N}_{[1,T]}$. The difference between (1) and (4), apart from the robot dynamics, is that as opposed to a probabilistic chance constraint (1f), we have a state hard (4f) which requires the robot to stay out of a union of convex, compact, and obstacle sets $\mathrm{BadSet}_j\subset\mathcal{X}$ which the robot must avoid at all times.

Note that (4) is non-convex due to the constraints (4b) and (4f). To tackle these non-convexities, the authors of [23], [24] proposed the use of *successive convexification* (`SCvx`). `SCvx` is an iterative method by which a non-convex optimal control problem is solved via a sequence of convex subproblems. The algorithm was introduced in [23] to handle nonlinear dynamics, and then extended in [24] to handle problems involving non-convex state constraints. `SCvx` serves as a general purpose solver to obtain a local optimum to (4). It makes use of trust regions to prevent the algorithm from taking aggressive steps. This reliance was altered with `SCvx-fast` [25] which can also solve (4) to local optimality efficiently under the following assumption.

Assumption 2. *In the problem* (4)*, we assume*

1) f *in* (4b) *is twice differentiable, convex, and potentially nonlinear over* $\mathcal{X}\times\mathcal{U}$*,*
2) $\mathrm{BadSet}_j\ \forall j\in\mathbb{N}_{[1,N_{\mathrm{obs}}]}$ *in* (4f) *can be expressed as* $\{\overline{y}\in\mathcal{X}:h_j(\overline{y})\le 0\}$ *with twice-differentiable, convex, and potentially nonlinear* $h_j:\mathcal{X}\to\mathbb{R}$*, and*
3) *a feasible solution is known for* (4)*.*

Note that the constraints (4b) and (4f) are still non-convex under Assumption 2. Assumption 2.2 does not exclude obstacle avoidance problems [25]. In [25], `SCvx` was used in conjunction (to create an initial feasible solution) with `SCvx-fast` to solve the motion planning problem (1) in significantly less computation time than using `SCvx` alone.

We next describe how `SCvx-fast` uses a project-and-linearize approach to convexify (4f) about a given trajectory $\overline{x}_{\mathrm{robot}}[\cdot]$. To employ `SCvx-fast`, we project $\overline{x}_t$ on BadSet_j to obtain $\mathrm{Proj}_{\mathrm{BadSet}_j}(\overline{x}_t)\in\mathrm{BadSet}_j,\forall t\in\mathbb{N}_{[1,T]},\ \forall j\in\mathbb{N}_{[1,N_{\mathrm{obs}}]}$. Since BadSet_j is convex by Assumption 2.2, this projection subproblem is convex [30, Sec. 8.1] [25, Thm. 4]. Next, we linearize the constraint $h_j(\overline{x}_t)>0$ at $\mathrm{Proj}_{\mathrm{BadSet}_j}(\overline{x}_t)$ to obtain a halfspace, $\mathcal{H}(\overline{x}_t,h_j)$, defined as

$$\mathcal{H}(\overline{x}_t,h_j)=\Big\{\overline{y}\in\mathcal{X}:\overline{p}(\overline{x}_t,h_j)^\top\Big(\overline{y}-\mathrm{Proj}_{\mathrm{BadSet}_j}(\overline{x}_t)\Big) + q(\overline{x}_t,h_j)>0\Big\}, \tag{6}$$

with $\overline{p}(\overline{x}_t,h_j)\triangleq\nabla h_j(\mathrm{Proj}_{\mathrm{BadSet}_j}(\overline{x}_t))\in\mathcal{X}$ and $q(\overline{x}_t,h_j)\triangleq h_j(\mathrm{Proj}_{\mathrm{BadSet}_j}(\overline{x}_t))\in\mathbb{R}$. Note that this is equivalent to the (non-convex) constraint $\overline{x}_{\mathrm{robot}}[t]\in(\mathcal{X}\setminus\mathrm{BadSet}_j)$ by Assumption 2.2, which when enforced for all $j\in\mathbb{N}_{[1,N_{\mathrm{obs}}]}$ yields (4f). Using (6), we define a (polyhedral) convex set $\mathcal{F}(\overline{x}_{\mathrm{robot}}[\cdot])$

$$\mathcal{F}(\overline{x}_{\mathrm{robot}}[\cdot]) = \bigcap_{t\in\mathbb{N}_{[1,T]},j\in\mathbb{N}_{[1,N_{\mathrm{obs}}]}} \mathcal{H}(\overline{x}_{\mathrm{robot}}[t], h_j). \tag{7}$$

The set $\mathcal{F}(\overline{x}_{\mathrm{robot}}[\cdot])$, turns out to be an underapproximation of the solution space corresponding to (4f) [25, Lem. 5].

Because `SCvx-fast` is iterative, we refer to the robot trajectory obtained at the k^{th} iteration as $\overline{x}^{(k)}_{\mathrm{robot}}[\cdot],\ k \geq 0$. Assumption 2.3 ensures the availability of $\overline{x}^{(0)}_{\mathrm{robot}}[\cdot]$ that satisfies (4b)–(4f). The `SCvx-fast` algorithm [25, Alg. 1] can be summarized in the following steps (with k set to zero):

1) transfer losslessly the nonlinear dynamics in (4b) to the objective using exact penalty functions to define a new objective $P : \mathcal{X}^T \times \mathcal{U}^T \to \mathbb{R}$ [25, Thm. 3],
2) convexify (4f) using project-and-linearize about $\overline{x}^{(k)}_{\mathrm{robot}}[\cdot]$ to obtain $\mathcal{F}(\overline{x}^{(k)}_{\mathrm{robot}}[\cdot])$ via (7),
3) solve the tractable convex subproblem — minimize P subject to $\mathcal{F}(\overline{x}^{(k)}_{\mathrm{robot}}[\cdot])$ and the (convex) constraints (4c)–(4e) — to obtain a feasible solution to (4), and
4) increment k, and repeat steps 2–4 until $\overline{x}^{(k)}_{\mathrm{robot}}[\cdot]$ converges [25, Thm. 11].

D. Problem statements

Problem 1. *Solve the stochastic motion planning problem* (1) *in a receding horizon control framework using successive convexification approach and probabilistic occupancy functions derived from forward stochastic reachability.*

Problem 1.a. *Overapproximate the probability-weighted keep-out region for a single rigid-body obstacle at a time of interest, given its stochastic dynamics* (3)*, its initial location* $\overline{x}_{\mathrm{obs},j}[0]$*, the required safe separation vector* r_j*, and a maximum allowed collision avoidance probability* $\beta \in (0, 1]$*.*

Problem 1.b. *Characterize the stochastic optimal control problem in the receding horizon control framework corresponding to* (1) *for a control horizon* $0 < N < T$*.*

Problem 1.c. *Adapt* `SCvx-fast` *to solve the optimal control problem obtained from Problem 1.b.*

III. Stochastic motion planning using SCvx-fast and probabilistic occupancy functions

A. Probabilistic occupancy function

We over-approximate the rigid-body shape of the obstacles with $\mathrm{Ball}(\boldsymbol{x}_{\mathrm{obs},j}[t], r_j)$. For each obstacle, we define a probabilistic occupancy function, $\phi_{\boldsymbol{x}_{\mathrm{obs},j}} : \mathcal{X} \to [0, 1]$, which provides the probability that a given state of interest $\overline{y} \in \mathcal{X}$ is within the ball $\mathrm{Ball}(\boldsymbol{x}_{\mathrm{obs},j}[t], r_j)$ at time $t > 0$, when its initial state is $\overline{x}_{\mathrm{obs},j}[0]$ [22, Sec. 5]. Formally,

$$\begin{aligned}&\phi_{\boldsymbol{x}_{\mathrm{obs},j}}(\overline{y}; t, \overline{x}_{\mathrm{obs},j}[0], r_j)\\ &= \mathbb{P}^{t,\overline{x}_{\mathrm{obs},j}[0]}_{\boldsymbol{x}_{\mathrm{obs},j}}\left\{\boldsymbol{x}_{\mathrm{obs},j}[t] \in \{\overline{z} \in \mathcal{X} : \overline{y} \in \mathrm{Ball}(\overline{z}, r_j)\}\right\}\end{aligned} \tag{8}$$

where $\mathbb{P}^{t,\overline{x}_{\mathrm{obs},j}[0]}_{\boldsymbol{x}_{\mathrm{obs},j}}$ is the probability measure associated with $\boldsymbol{x}_{\mathrm{obs},j}[t]$, the state of obstacle j. We can compute $\mathbb{P}^{t,\overline{x}_{\mathrm{obs},j}[0]}_{\boldsymbol{x}_{\mathrm{obs},j}}$ for arbitrary $\boldsymbol{w}_{\mathrm{obs},j}$ using forward stochastic reachability and Fourier transforms [21], [22]. Additionally, a probability density $\psi_{\boldsymbol{x}_{\mathrm{obs},j}}(\overline{y}; t, \overline{x}_{\mathrm{obs},j}[0])$ may be defined when $\boldsymbol{x}_{\mathrm{obs},j}[t]$ is absolutely continuous. Since $\mathrm{Ball}(\overline{0}, r_j)$ is symmetric, we can simplify (8) using [22, Prop. 1], $\phi_{\boldsymbol{x}_{\mathrm{obs},j}}(\overline{y}; t, \overline{x}_{\mathrm{obs},j}[0], r_j) = \mathbb{P}^{t,\overline{x}_{\mathrm{obs},j}[0]}_{\boldsymbol{x}_{\mathrm{obs},j}}\{\boldsymbol{x}_{\mathrm{obs},j}[t] \in \mathrm{Ball}(\overline{y}, r_j)\}$. Hence, the evaluation of $\phi_{\boldsymbol{x}_{\mathrm{obs},j}}(\overline{y}; t, \overline{x}_{\mathrm{obs},j}[0], r_j)$ may be done using numerical quadrature.

We define the probability-weighted keep-out region, referred to as *β-probability occupied set* $\beta \in [0, 1]$, as

$$\begin{aligned}&\mathrm{PrOccupySet}_{\boldsymbol{x}_{\mathrm{obs},j}}(\beta; t, \overline{x}_{\mathrm{obs},j}[0], r_j)\\ &\quad= \{\overline{y} \in \mathcal{X} : \phi_{\boldsymbol{x}_{\mathrm{obs},j}}(\overline{y}; t, \overline{x}_{\mathrm{obs},j}[0], r_j) \geq \beta\}.\end{aligned} \tag{9}$$

Avoiding the β-probability occupied set at a particular time t ensures the collision probability (separation smaller than r_j) with the obstacle initialized to $\overline{x}_{\mathrm{obs},j}[0]$ is below β.

Lemma 1. *[22, Thm. 5]* ***(Convex and compact*** **(9)*)*** *For any disturbance* $\boldsymbol{w}_{\mathrm{obs},j}$ *with log-concave probability measure* $\mathbb{P}_{\boldsymbol{w}_{\mathrm{obs},j}}$*, the* β*-probability occupied set* (9) *is convex and compact for any* $\beta \in (0, 1]$ *for all* t *and* $\overline{x}_{\mathrm{obs},j}[0]$*.*

Lemma 1 ensures that $\mathrm{PrOccupySet}_{\boldsymbol{x}_{\mathrm{obs},j}}$ may be treated as BadSet_j. We use the Minkowski-sum based overapproximation [22, Algo. 2] to overapproximate $\mathrm{PrOccupySet}_{\boldsymbol{x}_{\mathrm{obs},j}}$. Specifically, we construct $\mathrm{PrOccupySet}^{+}_{\boldsymbol{x}_{\mathrm{obs},j}}(\beta; t, \overline{x}_{\mathrm{obs},j}[0], r_j)$ using [22, Prop. 8] as

$$\begin{aligned}&\mathrm{PrOccupySet}^{+}_{\boldsymbol{x}_{\mathrm{obs},j}}(\beta; t, \overline{x}_{\mathrm{obs},j}[0], r_j)\\ &\quad= \left\{\overline{z} \in \mathcal{X} : \psi_{\boldsymbol{x}_{\mathrm{obs},j}}(\overline{y}; t, \overline{x}_{\mathrm{obs},j}[0]) \geq \frac{\beta}{\mathrm{m}(\mathrm{Ball}(\overline{0}, r_j))}\right\}\\ &\qquad\oplus \mathrm{Ball}(\overline{0}, r_j) \qquad (10)\\ &\quad\supseteq \mathrm{PrOccupySet}_{\boldsymbol{x}_{\mathrm{obs},j}}(\beta; t, \overline{x}_{\mathrm{obs},j}[0], r_j). \qquad (11)\end{aligned}$$

Next, we show that we can compute (10) efficiently for a Gaussian $\boldsymbol{w}_{\mathrm{obs},j} \sim \mathcal{N}(\overline{\mu}_{\boldsymbol{w}_{\mathrm{obs},j}}, \Sigma_{\boldsymbol{w}_{\mathrm{obs},j}})$. Linearity of (3) means $\boldsymbol{x}_{\mathrm{obs},j}[t] \sim \mathcal{N}(\overline{\mu}_{\boldsymbol{x}_{\mathrm{obs},j}}[t], \Sigma_{\boldsymbol{x}_{\mathrm{obs},j}}[t])$ (see [22, Sec. 4]),

$$\overline{\mu}_{\boldsymbol{x}_{\mathrm{obs},j}}[t] = A^t_{\mathrm{obs},j}\overline{x}_{\mathrm{obs},j}[0] + \mathscr{C}_W(t)(1_{t\times 1} \otimes \overline{\mu}_{\boldsymbol{w}_{\mathrm{obs},j}}) \tag{12a}$$

$$\Sigma_{\boldsymbol{x}_{\mathrm{obs},j}}[t] = \mathscr{C}_W(t)(I_\tau \otimes \Sigma_{\boldsymbol{w}_{\mathrm{obs},j}})\mathscr{C}_W^\top(t) \tag{12b}$$

where $\mathscr{C}_W(t) = [F_{\mathrm{obs},j}\ A_{\mathrm{obs},j}F_{\mathrm{obs},j}\ \ldots\ A^{t-1}_{\mathrm{obs},j}F_{\mathrm{obs},j}]$ is the extended controllability matrix for the dynamics (3). Note that the $\frac{\beta}{\mathrm{m}(\mathrm{Ball}(\overline{0},r_j))}$-superlevel set of the Gaussian is an ellipsoid [31, Defn 2.1.4], $\mathcal{E}(\overline{\mu}_{\boldsymbol{x}_{\mathrm{obs},j}}[t], Q_{\boldsymbol{x}_{\mathrm{obs},j}}[t]) = \{\overline{z} : (\overline{z} - \overline{\mu}_{\boldsymbol{x}_{\mathrm{obs},j}}[t])^\top Q^{-1}_{\boldsymbol{x}_{\mathrm{obs},j}}[t](\overline{z} - \overline{\mu}_{\boldsymbol{x}_{\mathrm{obs},j}}[t]) \leq 1\}$ with

$$Q_{\boldsymbol{x}_{\mathrm{obs},j}}[t] = -2\log\left(\frac{\beta\sqrt{|2\pi\Sigma_{\boldsymbol{x}_{\mathrm{obs},j}}[t]|}}{\mathrm{m}(\mathrm{Ball}(\overline{0}, r_j))}\right)\Sigma_{\boldsymbol{x}_{\mathrm{obs},j}}[t]. \tag{13}$$

The support function of $\mathcal{E}(\overline{\mu}_{\boldsymbol{x}_{\mathrm{obs},j}}[t], Q_{\boldsymbol{x}_{\mathrm{obs},j}}[t])$ is $\rho(\overline{l}; \mathcal{E}(\overline{\mu}_{\boldsymbol{x}_{\mathrm{obs},j}}[t], Q_{\boldsymbol{x}_{\mathrm{obs},j}}[t])) = \overline{l}^\top\overline{\mu}_{\boldsymbol{x}_{\mathrm{obs},j}}[t] + \|Q^{\frac{1}{2}}_{\boldsymbol{x}_{\mathrm{obs},j}}[t]\overline{l}\|_2$. Hence, (10) is the Minkowski sum of the ellipsoid $\mathcal{E}(\overline{\mu}_{\boldsymbol{x}_{\mathrm{obs},j}}[t], Q_{\boldsymbol{x}_{\mathrm{obs},j}}[t])$ with another ellipsoid $\mathcal{E}(\overline{0}, r_j^2 I_n) = \mathrm{Ball}(0, r_j)$. Recall that the class of ellipsoids is not closed under Minkowski sums [31, Pg. 97]. Nevertheless, we

Algorithm 1 Ellipsoidal overapproximation of $\mathrm{PrOccupySet}_{\boldsymbol{x}_{\mathrm{obs},j}}(\beta; t, \overline{x}_{\mathrm{obs},j}[0], r_j)$ for Gaussian $\boldsymbol{x}_{\mathrm{obs},j}[t]$

Input: Time of interest t, obstacle dynamics (3), initial obstacle state $\overline{x}_{\mathrm{obs},j}[0]$, separation distance r_j, maximum allowed collision probability $\beta \in (0, 1]$, tightness direction $\overline{l}_0$

Output: $\mathcal{E}(\overline{\mu}_{\boldsymbol{x}_{\mathrm{obs},j}}[t], Q^{+}_{\boldsymbol{x}_{\mathrm{obs},j}}[t])$

Compute $(\overline{\mu}_{\boldsymbol{x}_{\mathrm{obs},j}}[t], \Sigma_{\boldsymbol{x}_{\mathrm{obs},j}}[t])$ using (12a) and (12b)
if $\phi_{\boldsymbol{x}_{\mathrm{obs},j}}(\overline{\mu}_{\boldsymbol{x}_{\mathrm{obs},j}}; t, \overline{x}_{\mathrm{obs},j}[0], r_j) \leq \beta$ **then**
    return $\emptyset$ by Lemma 2
else
    Compute $Q_{\boldsymbol{x}_{\mathrm{obs},j}}[t]$ using (13)
    Compute $Q^{+}_{\boldsymbol{x}_{\mathrm{obs},j}}[t]$ using (15)
    return $\mathcal{E}(\overline{\mu}_{\boldsymbol{x}_{\mathrm{obs},j}}[t], Q^{+}_{\boldsymbol{x}_{\mathrm{obs},j}}[t])$
end if

have a closed-form expression for the support function of $\mathrm{PrOccupySet}^{+}_{\boldsymbol{x}_{\mathrm{obs},j}}(\beta; t, \overline{x}_{\mathrm{obs},j}[0], r_j)$ (see [22, Sec. 6.B]), $\rho(\overline{l}; \mathrm{PrOccupySet}^{+}_{\boldsymbol{x}_{\mathrm{obs},j}}(\beta; t, \overline{x}_{\mathrm{obs},j}[0], r_j)) = \overline{l}^{\top} \overline{\mu}_{\boldsymbol{x}_{\mathrm{obs},j}}[t] + \sqrt{\overline{l}^{\top} Q_{\boldsymbol{x}_{\mathrm{obs},j}}[t] \overline{l}} + r_j \|\overline{l}\|_2$. Using ellipsoidal calculus techniques, we can also create an ellipsoidal outer approximation that is tight along any specified direction $\overline{l}_0 \in \mathbb{R}^n$ [31, Lem. 2.2.1],

$$\mathrm{PrOccupySet}^{+}_{\boldsymbol{x}_{\mathrm{obs},j}}(\beta; t, \overline{x}_{\mathrm{obs},j}[0], r_j) \subseteq \mathcal{E}(\overline{\mu}_{\boldsymbol{x}_{\mathrm{obs},j}}[t], Q^{+}_{\boldsymbol{x}_{\mathrm{obs},j}}[t]) \qquad (14)$$

$$Q^{+}_{\boldsymbol{x}_{\mathrm{obs},j}}[t] = \left(\sqrt{\overline{l}_0^{\top} Q_{\boldsymbol{x}_{\mathrm{obs},j}}[t] \overline{l}_0} + r_j \|\overline{l}_0\|_2 \right) \times \left(\frac{Q_{\boldsymbol{x}_{\mathrm{obs},j}}[t]}{\sqrt{\overline{l}_0^{\top} Q_{\boldsymbol{x}_{\mathrm{obs},j}}[t] \overline{l}_0}} + \frac{r_j}{\|\overline{l}_0\|_2} I_n \right), \qquad (15)$$

meaning that the over approximation $\mathcal{E}(\overline{\mu}_{\boldsymbol{x}_{\mathrm{obs},j}}[t], Q^{+}_{\boldsymbol{x}_{\mathrm{obs},j}}[t])$ and $\mathrm{PrOccupySet}^{+}_{\boldsymbol{x}_{\mathrm{obs},j}}(\beta; t, \overline{x}_{\mathrm{obs},j}[0], r_j)$ share the same hyperplane defined using $\overline{l}_0$. Algorithm 1 solves Problem 1.a.

Lemma 2. ***(Non-empty*** $\mathrm{PrOccupySet}_{\boldsymbol{x}_{\mathrm{obs},j}}(\beta; t, \overline{x}_{\mathrm{obs},j}[0], r_j)$***)*** *For a Gaussian* $\boldsymbol{x}_{\mathrm{obs},j}[t]$*, if* $\phi_{\boldsymbol{x}_{\mathrm{obs},j}}(\overline{\mu}_{\boldsymbol{x}_{\mathrm{obs},j}}; t, \overline{x}_{\mathrm{obs},j}[0], r_j) \leq \beta$*, then* $\mathrm{PrOccupySet}_{\boldsymbol{x}_{\mathrm{obs},j}}(\beta; t, \overline{x}_{\mathrm{obs},j}[0], r_j)$ *is empty.*

Proof: For Gaussian $\boldsymbol{x}_{\mathrm{obs},j}[t]$, the maximum of the probabilistic occupancy function occurs at the mean $\overline{\mu}_{\boldsymbol{x}_{\mathrm{obs},j}}[t]$ by [22, Prop. 3]. ■

B. Receding horizon control framework to solve (1)

The uncertainty in the stochastic obstacle dynamics grows over time, resulting in $\mathrm{PrOccupySet}_{\boldsymbol{x}_{\mathrm{obs},j}}(\beta; t, \overline{x}_{\mathrm{obs},j}[0], r_j)$ being large for large t for any $\beta \in (0, 1]$. Even conservative approximations of these large sets may induce infeasibility, even when the original problem is feasible. Replanning the robot trajectory based on the information available about the obstacle movement can reduce this conservativism. This motivates the use of receding horizon control. We choose a control horizon $0 < N < T$ and solve a problem similar to (1), referred to as the *receding horizon optimal control problem* (RHOCP), for the time interval $\mathbb{N}_{[t,t+N]}$.

Assumption 3. *Full-state information is available about all the obstacles after executing the first action prescribed by the solution to the RHOCP.*

We will formulate the RHOCP in the form (4) to make it amenable to `SCvx-fast`. First, we retain the constraints (1b)–(1d) in (1). Second, we alter (1e) since $\overline{x}_{\mathrm{robot}}[T]$ is no longer a decision variable. We solve an additional convex optimization problem to find $\overline{x}_{\mathrm{robot}}[N]$ closest to the GoalSet,

$$\underset{\overline{y}}{\mathrm{minimize}} \quad \|\overline{x}_{\mathrm{robot}}[N] - \overline{y}\|_2 \qquad (16a)$$

$$\mathrm{subject\ to} \quad \overline{y} \in \mathrm{GoalSet} \qquad (16b)$$

The optimal value of (16) is zero only when $\overline{x}_{\mathrm{robot}}[N] \in \mathrm{GoalSet}$, in which case the optimal solution is $\overline{y}^{*} = \overline{x}_{\mathrm{robot}}[N]$. Using the fixed-risk approach in [12], we replace (1f) with

$$\overline{x}_{\mathrm{robot}}[t] \notin \bigcup_{j=1}^{N_{\mathrm{obs}}} \mathrm{PrOccupySet}_{\boldsymbol{x}_{\mathrm{obs},j}}\left(\frac{\alpha}{N_{\mathrm{obs}}}; t, \overline{x}_{\mathrm{obs},j}[0], r_j \right). \qquad (17)$$

Proposition 1. *Satisfaction of* (17) *implies satisfaction of* (1f).

Proof: Let $\overline{y} \in \mathcal{X}$ satisfy (17), i.e., $\overline{y} \in \left(\mathcal{X} \setminus \cup_{j=1}^{N_{\mathrm{obs}}} \mathrm{PrOccupySet}_{\boldsymbol{x}_{\mathrm{obs},j}}\left(\frac{\alpha}{N_{\mathrm{obs}}}; t, \overline{x}_{\mathrm{obs},j}[0], r_j \right) \right) \Rightarrow \overline{y} \in \cap_{j=1}^{N_{\mathrm{obs}}} \left(\mathcal{X} \setminus \mathrm{PrOccupySet}_{\boldsymbol{x}_{\mathrm{obs},j}}\left(\frac{\alpha}{N_{\mathrm{obs}}}; t, \overline{x}_{\mathrm{obs},j}[0], r_j \right) \right)$. From (9), $\forall j \in \mathbb{N}_{[1,N_{\mathrm{obs}}]}$, $\phi_{\boldsymbol{x}_{\mathrm{obs},j}}(\overline{y}; t, \overline{x}_{\mathrm{obs},j}[0], r_j) < \frac{\alpha}{N_{\mathrm{obs}}} \Rightarrow \sum_{j=1}^{N_{\mathrm{obs}}} \phi_{\boldsymbol{x}_{\mathrm{obs},j}}(\overline{y}; t, \overline{x}_{\mathrm{obs},j}[0], r_j) < \alpha$. Recall that $\mathbb{P}\{\mathcal{S}_1 \cup \mathcal{S}_2\} \leq \mathbb{P}\{\mathcal{S}_1\} + \mathbb{P}\{\mathcal{S}_2\}$, for any two Borel sets $\mathcal{S}_1, \mathcal{S}_2$ by Boole's identity. By (5), $\phi_{\boldsymbol{X}_{\mathrm{obs}}}(\overline{y}; t, \overline{X}_{\mathrm{obs}}[0], \overline{r}) \leq \sum_{j=1}^{N_{\mathrm{obs}}} \phi_{\boldsymbol{x}_{\mathrm{obs},j}}(\overline{y}; t, \overline{x}_{\mathrm{obs},j}[0], r_j)$. Thus, $\overline{y}$ satisfies (1f). ■

Lastly, using (16) and (17), we obtain a bi-criterion optimization problem [30, Sec. 4.7] that is a conservative approximation of (1),

$$\begin{aligned} \underset{\substack{\overline{y}, \overline{x}_{\mathrm{robot}}[1], \ldots, \overline{x}_{\mathrm{robot}}[N] \\ \overline{u}_{\mathrm{robot}}[0], \ldots, \overline{u}_{\mathrm{robot}}[N-1]}}{\mathrm{minimize}} \quad & \begin{bmatrix} J_R(\overline{x}_{\mathrm{robot}}[1], \ldots, \overline{x}_{\mathrm{robot}}[N], \\ \quad \overline{u}_{\mathrm{robot}}[0], \ldots, \overline{u}_{\mathrm{robot}}[N-1]) \\ \text{(16a)} \end{bmatrix} \\ \mathrm{subject\ to} \quad & \begin{cases} \text{(1b), (1c), (1d),} \\ \text{(1e), (16b), (17)} \end{cases} \end{aligned} \qquad (18)$$

where $J_R : \mathcal{X}^N \times \mathcal{U}^N \to \mathbb{R}$ approximates the original cost J (defined for the time horizon T) over the planning horizon N. We finally obtain the RHOCP in (19) by introducing the parameter $\lambda \geq 0$ and scalarizing the bi-criterion optimization problem (see [30, Sec. 4.7.4]). This solves Problem 1.b.

The objective in (19) is convex with respect to the decision variables $\overline{x}_{\mathrm{robot}}[\cdot]$ and $\overline{u}_{\mathrm{robot}}[\cdot]$. We can interpret λ in (19) as a way to emphasize the relative importance of being close to the goal set with respect to the optimization of the cost function J_R. For large λ, the solver attempts to generate a trajectory that will minimize the distance of the terminal state at the expense of a potential increase in the cost function J_R.

Receding horizon problem formulation of (1) *cast in the form* (4):

$$\underset{\substack{\overline{y},\overline{x}_{\mathrm{robot}}[1],\ldots,\overline{x}_{\mathrm{robot}}[N],\\ \overline{u}_{\mathrm{robot}}[0],\ldots,\overline{u}_{\mathrm{robot}}[N-1]}}{\mathrm{minimize}} \quad J_R\big(\overline{u}_{\mathrm{robot}}[0],\overline{u}_{\mathrm{robot}}[1],\ldots,\overline{u}_{\mathrm{robot}}[N-1],\overline{x}_{\mathrm{robot}}[1],\overline{x}_{\mathrm{robot}}[2],\ldots,\overline{x}_{\mathrm{robot}}[N]\big)+\lambda\|\overline{x}_{\mathrm{robot}}[N]-\overline{y}\|_2 \tag{19a}$$

$$\begin{aligned}
\text{subject to} \quad & \overline{x}_{\mathrm{robot}}[t+1]=A_{\mathrm{robot}}\overline{x}_{\mathrm{robot}}[t]+B_{\mathrm{robot}}\overline{u}_{\mathrm{robot}}[t] && t\in\mathbb{N}_{[0,N-1]}, && \text{(19b)}\\
& \overline{u}_{\mathrm{robot}}[t]\in\mathcal{U} && t\in\mathbb{N}_{[0,N-1]}, && \text{(19c)}\\
& \overline{x}_{\mathrm{robot}}[t]\in\mathrm{SafeSet} && t\in\mathbb{N}_{[1,N-1]}, && \text{(19d)}\\
& \overline{y}\in\mathrm{GoalSet}, && && \text{(19e)}\\
& \overline{x}_{\mathrm{robot}}[t]\not\in\bigcup_{j=1}^{N_{\mathrm{obs}}}\mathrm{PrOccupySet}_{\boldsymbol{x}_{\mathrm{obs},j}}\left(\frac{\alpha}{N_{\mathrm{obs}}};t,\overline{x}_{\mathrm{obs},j}[0],r_j\right) && t\in\mathbb{N}_{[1,N]} && \text{(19f)}
\end{aligned}$$

Note that (19) is in the form of (4) with linear robot dynamics. We define $\mathrm{BadSet}_j[t]=\{\overline{y}\in\mathcal{X}:h_{j,t}(\overline{y})\leq 0\}$ for (4f) as $\mathrm{PrOccupySet}^{+}_{\boldsymbol{x}_{\mathrm{obs},j}}\left(\frac{\alpha}{N_{\mathrm{obs}}};t,\overline{x}_{\mathrm{obs},j}[0],r_j\right)$ by,

$$h_{j,t}(\overline{y})\triangleq\Big((\overline{y}-\overline{\mu}_{\boldsymbol{x}_{\mathrm{obs},j}}[t])^\top(Q^{+}_{\boldsymbol{x}_{\mathrm{obs},j}}[t])^{-1}(\overline{y}-\overline{\mu}_{\boldsymbol{x}_{\mathrm{obs},j}}[t])\Big)-1 \tag{20}$$

for every $t\in\mathbb{N}_{[1,N]}$ as seen from (11) and (14). Note that $h_{j,t}$ satisfies Assumption 2.2. We will use Algorithm 1 to generate $\mathrm{BadSet}_j[t]$. We next discuss how `SCvx-fast` can be used to solve (4) and (19) for linear robot dynamics.

C. Using successive convexification to solve (19)

We have a convex relaxation of (19) (excluding (19f)),

$$\begin{aligned}
&\underset{\substack{\overline{x}_{\mathrm{robot}}[1],\overline{x}_{\mathrm{robot}}[2],\ldots,\overline{x}_{\mathrm{robot}}[N],\\ \overline{u}_{\mathrm{robot}}[0],\overline{u}_{\mathrm{robot}}[1],\ldots,\overline{u}_{\mathrm{robot}}[N-1]}}{\mathrm{minimize}} && \text{(19a)}\\
&\text{subject to} && \begin{cases}\text{(19b)},\text{(19c)},\\ \text{(19d)},\text{(19e)}\end{cases}
\end{aligned} \tag{21}$$

Let $(\overline{x}^{\dagger}_{\mathrm{robot}}[\cdot],\overline{u}^{\dagger}_{\mathrm{robot}}[\cdot])$ denote the solution (global optimum) to (21) with $\overline{x}^{\dagger}_t\triangleq\overline{x}^{\dagger}_{\mathrm{robot}}[t]$, $\forall t\in\mathbb{N}_{[1,N]}$. If $\overline{x}^{\dagger}_{\mathrm{robot}}[\cdot]$ does not violate (19f), then $\overline{x}^{\dagger}_{\mathrm{robot}}[\cdot]$ is the global optimum for (19). On the other hand, if $\overline{x}^{\dagger}_{\mathrm{robot}}[\cdot]$ violates (19f) for some t,j (as typically will be the case), we can use `SCvx-fast` to obtain a locally optimal trajectory to the non-convex problem (19). To use `SCvx-fast`, we however need to provide $(\overline{x}^{(0)}_{\mathrm{robot}}[\cdot],\overline{u}^{(0)}_{\mathrm{robot}}[\cdot])$ to satisfy Assumption 2.3.

We can use `SCvx` to compute $(\overline{x}^{(0)}_{\mathrm{robot}}[\cdot],\overline{u}^{(0)}_{\mathrm{robot}}[\cdot])$ as done in [25]. In this section, we provide a non-iterative alternative to `SCvx` for initializing `SCvx-fast` when the robot dynamics are linear—solving the feasibility problem (see [30, Sec. 4.1.1]) (22), defined using $\overline{x}^{\dagger}_{\mathrm{robot}}[\cdot]$ and $\mathcal{H}_{\mathrm{supp,j}}(\overline{x}^{\dagger}_t)=\mathcal{H}(\overline{x}_t,h_{j,t})$ (see (6)),

$$\begin{aligned}
\text{find} \quad & \overline{x}^{(0)}_{\mathrm{robot}}[1],\ldots,\overline{x}^{(0)}_{\mathrm{robot}}[N],\overline{u}^{(0)}_{\mathrm{robot}}[0],\ldots,\overline{u}^{(0)}_{\mathrm{robot}}[N-1]\\
\text{s.t.} \quad & \begin{cases}\text{(19b)},\text{(19c)},\text{(19d)},\text{(19e)}\\ \overline{x}^{(0)}_{\mathrm{robot}}[t]\in\cap_{j=1}^{N_{\mathrm{obs}}}\mathcal{H}_{\mathrm{supp,j}}(\overline{x}^{\dagger}_t)\quad\forall t\in\mathbb{N}_{[1,N]}\end{cases}
\end{aligned} \tag{22}$$

Note that (22) is a convex optimization problem since the robot dynamics are linear, and the sets SafeSet, GoalSet, and $\mathcal{H}(\cdot,\cdot)$ are convex. Additionally, (22) is a linear program when SafeSet and GoalSet are polytopic, i.e., characterized by a set of linear inequalities.

Proposition 2. ***(Satisfaction of Assumption 2.3)*** *If* (22) *has a feasible solution, then this solution is feasible for* (19).

Proof: Comparing (22) with (19), only (19f) has been replaced. We have to show that $\mathcal{H}(\overline{x}_t,h_{j,t})\subseteq\mathcal{X}\setminus\mathrm{BadSet}_j[t]$, $\forall t\in\mathbb{N}_{[1,N]},\forall j\in\mathbb{N}_{[1,N_{\mathrm{obs}}]}$. By Assumption 2.2, $h_{j,t}(\overline{y})$ is convex in $\overline{y}\in\mathcal{X},$, $\forall t\in\mathbb{N}_{[1,N]},\forall j\in\mathbb{N}_{[1,N_{\mathrm{obs}}]}$. Recall that the first-order approximation of a convex function is a *global underestimator* of the function [30, Sec. 3.1.3]. Therefore, for any $\overline{x}_t,\overline{y}\in\mathcal{X}$, $j\in\mathbb{N}_{[1,N_{\mathrm{obs}}]}$, and $t\in\mathbb{N}_{[1,N]}$, $h_{j,t}(\overline{y})\geq\overline{p}(\overline{x}_t,h_{j,t})^\top\left(\overline{y}-\mathrm{Proj}_{\mathrm{BadSet}_j}(\overline{x}_t)\right)+q(\overline{x}_t,h_{j,t})$. By (6), $\overline{y}\in\mathcal{H}(\overline{x}_t,h_{j,t})\Rightarrow\overline{y}\in\mathcal{X}\setminus\mathrm{BadSet}_j[t]$. ∎

Note that (22) may be infeasible even when (19) is feasible, due to the additional conservativeness introduced by the linear overapproximations of the keep-out regions. In such situations, we have to rely on `SCvx` to satisfy Assumption 2.3 as done in [23]. Due to its iterative approach and use of trust regions, `SCvx` can find a feasible trajectory. Thus, Algorithm 2 solves (19) and Problem 1.c, without requiring the user to provide a feasible initial trajectory.

We summarize the receding horizon control-based solution to (1) using successive convexification and probabilistic occupancy function in Algorithm 3, solving Problem 1.

IV. NUMERICAL RESULTS

All computations were done using MATLAB on an Intel Xeon CPU with 3.5GHz clock rate and 16 GB RAM. We used `ET` [32] for the ellipsoidal computations, `MPT` [33] for the polyhedral computations, and `CVX` [34] for solving the convex optimization problems.

We consider the planning problem with double integrator robot dynamics (2),

$$A_{\mathrm{robot}}=\begin{bmatrix}1&0&T_\Delta&0\\0&1&0&T_\Delta\\0&0&1&0\\0&0&0&1\end{bmatrix}, \& \ B_{\mathrm{robot}}=\begin{bmatrix}\frac{T_\Delta^2}{2}&0\\0&\frac{T_\Delta^2}{2}\\T_\Delta&0\\0&T_\Delta\end{bmatrix}.$$

with sampling time $T_\Delta=0.25$ s, the initial robot location $\overline{x}_{\mathrm{robot}}[0]=[10\ 0]^\top$, and $\mathcal{U}=[-2,2]^2$. We

Algorithm 2 Assumption 2.3-relaxed `SCvx-fast` for (2)

Input: Robot and obstacle dynamics (2) and (3), robot and obstacle initial locations $\overline{x}_{\mathrm{robot}}[0]$ and $\overline{x}_{\mathrm{obs},j}[0]$, safe set SafeSet, goal set GoalSet $\subseteq$ SafeSet, cost function J_R, sets to avoid BadSet_j, time horizon $N > 0$

Output: Locally optimal solution to (19)

Compute $\overline{x}^{\dagger}_{\mathrm{robot}}[\cdot], \overline{u}^{\dagger}_{\mathrm{robot}}[\cdot]$ by solving (21)
if $\overline{x}^{\dagger}_{\mathrm{robot}}[\cdot]$ satisfies (19f) **then**
    return $\overline{x}^{\dagger}_{\mathrm{robot}}[\cdot], \overline{u}^{\dagger}_{\mathrm{robot}}[\cdot]$
else
    try
        Compute $\overline{x}^{(0)}_{\mathrm{robot}}[\cdot], \overline{u}^{(0)}_{\mathrm{robot}}[\cdot]$ by solving (22) defined using $\overline{x}^{\dagger}_{\mathrm{robot}}[\cdot]$
    catch (22) is infeasible
        Compute $\overline{x}^{(0)}_{\mathrm{robot}}[\cdot], \overline{u}^{(0)}_{\mathrm{robot}}[\cdot]$ with `SCvx` [24]
    end try
    Compute $\overline{x}^{*}_{\mathrm{robot}}[\cdot], \overline{u}^{*}_{\mathrm{robot}}[\cdot]$ with `SCvx-fast` [25]
    return $\overline{x}^{*}_{\mathrm{robot}}[\cdot], \overline{u}^{*}_{\mathrm{robot}}[\cdot]$
end if

Algorithm 3 Receding horizon control framework to solve (1) using `SCvx-fast` and $\mathrm{PrOccupySet}^{+}_{\boldsymbol{x}_{\mathrm{obs},j}}$

Input: Robot and obstacle dynamics (2) and (3), robot and obstacle initial locations $\overline{x}_{\mathrm{robot}}[0]$ and $\overline{x}_{\mathrm{obs},j}[0]$, safe set SafeSet, goal set GoalSet $\subseteq$ SafeSet, cost function J, maximum allowed collision probability α, safe separation vector $\overline{r}$, time horizon $T > 0$, control horizon $0 < N < T$, λ for (19a)

Output: Robot trajectory and controller, feasible for (1)

$t \leftarrow 0$
while $t < T$ **do**
    for all $j \in \mathbb{N}_{[1,N_{\mathrm{obs}}]}, k \in \mathbb{N}_{[1,N]}$ **do**
        $\mathrm{BadSet}_j[t+k] \leftarrow \mathcal{E}(\overline{\mu}_{\boldsymbol{x}_{\mathrm{obs},j}}[t+k], Q^{+}_{\boldsymbol{x}_{\mathrm{obs},j}}[t+k])$ using Algorithm 1
    end for
    Solve (19) using Algorithm 2 and $\mathrm{BadSet}_j[\cdot]$
    Update the robot state to $\overline{x}_{\mathrm{robot}}[t+1]$
    Get new obstacle states $\boldsymbol{x}_{\mathrm{obs},j}[t+1]$, $\forall j \in \mathbb{N}_{[1,N_{\mathrm{obs}}]}$
    $t \leftarrow t+1$ and $N \leftarrow \min(N, T-t)$
end while

set $N_{\mathrm{obs}} = 3$, and presume double integrator dynamics for the obstacles (3) with $A_{\mathrm{obs},j} = A_{\mathrm{robot}}$, $F_{\mathrm{obs},j} = B_{\mathrm{robot}}$, and Gaussian $\boldsymbol{w}_{\mathrm{obs},j}[t] \sim \mathcal{N}(\overline{\mu}_{\boldsymbol{w}_{\mathrm{obs},j}}, \Sigma_{\boldsymbol{w}_{\mathrm{obs},j}})$, $j \in \mathbb{N}_{[1,N_{\mathrm{obs}}]}$. We choose the obstacle initial positions $\overline{x}_{\mathrm{obs},1}[0] = [10\ 25]^\top$, $\overline{x}_{\mathrm{obs},2}[0] = [35\ 5]^\top$, $\overline{x}_{\mathrm{obs},3}[0] = [70\ 80]^\top$, the mean values $\overline{\mu}_{\boldsymbol{w}_{\mathrm{obs},1}} = [3\ -3]^\top$, $\overline{\mu}_{\boldsymbol{w}_{\mathrm{obs},2}} = [-2\ 3]^\top$, $\overline{\mu}_{\boldsymbol{w}_{\mathrm{obs},3}} = [-3\ -5]^\top$, and the covariance matrix $\Sigma_{\boldsymbol{w}_{\mathrm{obs},j}} = \begin{bmatrix} 2.60 & 0.09 \\ 0.09 & 0.58 \end{bmatrix}$ $\forall j \in \{1,2,3\}$. We solve (1) with a time horizon $T = 15$ s (60 time steps), the cost function $J_R(\cdot) = \sum_{t=0}^{T-1} \|\overline{u}_{\mathrm{robot}}[t]\|_2$ to minimize as the control effort, SafeSet $= \mathrm{ConvexHull}((50,40),(40,0),(0,-5),(10,30))$, GoalSet $= \mathrm{Ball}([40\ 35]^\top, 2)$, control horizon $N = 13$, maximum acceptable collision probability $\alpha = 0.001$, scalarization parameter $\lambda = 10^4$, and safe separation $\overline{r} = [2\ 2\ 2]$.

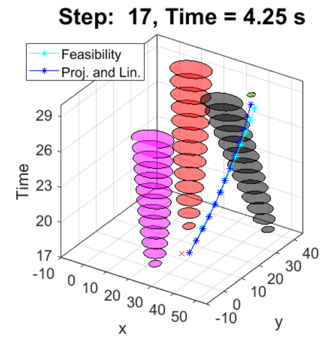


Fig. 1: Planning over a single control horizon with the uncertainty in the obstacle location growing over time

Computation step	Mean (s)	Std. Dev (s)
Algorithm 1	0.20	0.03
Solve (22)	0.05	0.07
Execute `SCvx-fast`	0.08	0.11
Total time (Solve (19))	0.33	0.17

TABLE I: Computation times statistics for Figure 3.

Figure 1 shows the evolution of $\mathcal{E}(\overline{\mu}_{\boldsymbol{x}_{\mathrm{obs},j}}[t], Q^{+}_{\boldsymbol{x}_{\mathrm{obs},j}}[t])$ via Algorithm 1 for a single control horizon, for both the initial feasible solution and project-and-linearize solutions.

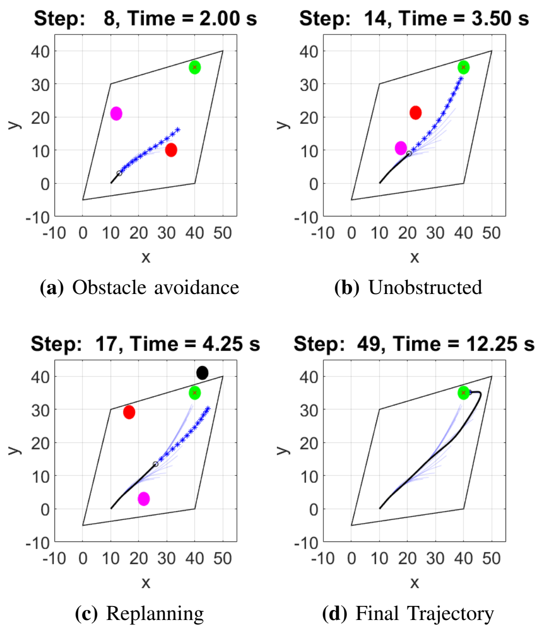
Figure 2 shows the realization of the stochastic motion planning problem at different time instants. We simulated the obstacle motion by drawing samples from their respective disturbance probability densities. The probability of an obstacle (black) reaching the GoalSet becomes non-trivial at $t = 4.25s$, causing the planner to re-plan its trajectory.

Figure 3 and Table I describes the computation time for components of Algorithm 3. Because we do not invoke `SCvx-fast` when the path from the robot to the final location is free of obstacles (see Alg. 2), sharp changes are evident in computation time when the robot encounters the black obstacle. The computation time for `SCvx-fast` and solution to (22) are for the solver computation times after `CVX` parses the optimization problem. In [35], the implementation of `SCvx-fast` with a custom solver was able to solve complex problems on the order of milliseconds.

V. Conclusions and Future Work

We have proposed a real-time implementable stochastic motion planner which generates dynamically-feasible trajectories for a robot with LTI dynamics while satisfying hard state and input constraints and guaranteeing probabilistic safety in the presence of rigid-body obstacles that have stochastic LTI dynamics. Our receding horizon control-based approach builds on recent results in forward stochastic reachability and successive convexification.

In future work, we will study properties of (22) and characterize sufficient conditions for its feasiblity.



(a) Obstacle avoidance **(b)** Unobstructed

(c) Replanning **(d)** Final Trajectory

Fig. 2: Implementation of Algorithm 3 to solve (1). The robot must stay within the quadrilateral, eventually reach the green region, and avoid stochastically moving obstacles (red, pink, and black). The faded blue lines and the black line show the intermediate plans and the executed trajectory respectively.

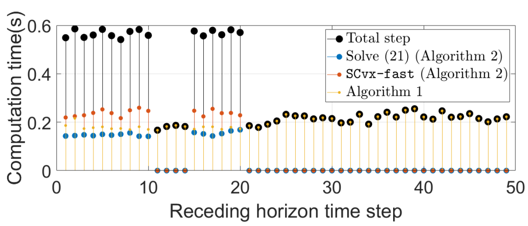


Fig. 3: Computation time for various steps in Algorithm 3.

References

[1] S. M. LaValle, *Planning Algorithms*. Cambridge Univ. Press, 2006.

[2] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT Press, 2005.

[3] M. Ono, M. Pavone, Y. Kuwata, and J. Balaram, "Chance-constrained dynamic programming with application to risk-aware robotic space exploration," *Autonomous Robots*, vol. 39, no. 4, pp. 555–571, 2015.

[4] L. Blackmore, M. Ono, and B. Williams, "Chance-constrained optimal path planning with obstacles," *IEEE Trans. Robot.*, vol. 27, no. 6, pp. 1080–1094, 2011.

[5] M. Ono, M. Quadrelli, and T. L. Huntsberger, "Safe maritime autonomous path planning in a high sea state," in *Proc. American Ctrl. Conf.*, 2014, pp. 4727–4734.

[6] A. Wu and J. P. How, "Guaranteed infinite horizon avoidance of unpredictable, dynamically constrained obstacles," *Autonomous Robots*, vol. 32, no. 3, pp. 227–242, 2012.

[7] A. Aswani, H. Gonzalez, S. Sastry, and C. Tomlin, "Provably safe and robust learning-based model predictive control," *Automatica*, vol. 49, no. 5, pp. 1216–1226, 2013.

[8] J. Fisac, M. Chen, C. Tomlin, and S. Sastry, "Reach-avoid problems with time-varying dynamics, targets and constraints," in *Proc. Hybrid Syst.: Comput. and Ctrl.*, 2015, pp. 11–20.

[9] B. HomChaudhuri, A. P. Vinod, and M. Oishi, "Computation of forward stochastic reach sets: Application to stochastic, dynamic obstacle avoidance," in *Proc. American Ctrl. Conf.*, 2017, pp. 4404–4411.

[10] A. Abate, M. Prandini, J. Lygeros, and S. Sastry, "Probabilistic reachability and safety for controlled discrete time stochastic hybrid systems," *Automatica*, vol. 44, no. 11, pp. 2724–2734, 2008.

[11] S. Summers, M. Kamgarpour, J. Lygeros, and C. Tomlin, "A stochastic reach-avoid problem with random obstacles," in *Proc. Hybrid Syst.: Comput. and Ctrl.*, 2011, pp. 251 – 260.

[12] L. Blackmore, Hui Li, and B. Williams, "A probabilistic approach to optimal robust path planning with obstacles," in *Proc. American Ctrl. Conf.*, 2006, pp. 2831–2837.

[13] A. Abate, S. Amin, M. Prandini, J. Lygeros, and S. Sastry, "Computational approaches to reachability analysis of stochastic hybrid systems," in *Proc. Hybrid Syst.: Comput. and Ctrl.*, 2007, pp. 4–17.

[14] G. Aoude, B. Luders, J. Joseph, N. Roy, and J. How, "Probabilistically safe motion planning to avoid dynamic obstacles with uncertain motion patterns," *Autonomous Robots*, vol. 35, no. 1, pp. 51–76, 2013.

[15] N. Malone, H.-T. Chiang, K. Lesser, M. Oishi, and L. Tapia, "Hybrid dynamic moving obstacle avoidance using a stochastic reachable set-based potential field," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1124–1138, 2017.

[16] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int'l J. Robotics and Research*, vol. 30, no. 7, pp. 846–894, 2011.

[17] L. Janson, E. Schmerling, A. Clark, and M. Pavone, "Fast marching tree: A fast marching sampling-based method for optimal motion planning in many dimensions," *Int'l J. Robotics and Research*, vol. 34, no. 7, pp. 883–921, Jun. 2015.

[18] H.-T. Chiang, N. Malone, K. Lesser, M. Oishi, and L. Tapia, "Aggressive moving obstacle avoidance using a stochastic reachable set based potential field," in *Algorithmic Found. of Robotics*. Springer, 2015, pp. 73–89.

[19] B. Ichter, E. Schmerling, A. Agha-mohammadi, and M. Pavone, "Real-time stochastic kinodynamic motion planning via multiobjective search on gpus," in *Proc. IEEE Int'l Conf. Robotics and Autom.*, 2017, pp. 5019–5026.

[20] G. Calafiore and M. Campi, "The scenario approach to robust control design," *IEEE Trans. Autom. Control*, vol. 51, no. 5, pp. 742–753, 2006.

[21] A. P. Vinod, B. HomChaudhuri, and M. Oishi, "Forward stochastic reachability analysis for uncontrolled linear systems using Fourier Transforms," in *Proc. Hybrid Syst.: Comput. and Ctrl.*, 2017, pp. 35–44.

[22] A. P. Vinod and M. Oishi, "Probabilistic Occupancy Function and Sets Using Forward Stochastic Reachability for Rigid-Body Dynamic Obstacles," *IEEE Trans. Autom. Control*, 2018, (submitted). [Online]. Available: https://arxiv.org/abs/1803.07180v2

[23] Y. Mao, M. Szmuk, and B. Açıkmeşe, "Successive convexification of non-convex optimal control problems and its convergence properties," in *Proc. IEEE Conf. Dec. & Ctrl.*, 2016, pp. 3636–3641.

[24] D. Dueri, Y. Mao, Z. Mian, J. Ding, and B. Açıkmeşe, "Trajectory optimization with inter-sample obstacle avoidance via successive convexification," in *Proc. IEEE Conf. Dec. & Ctrl.*, 2017, pp. 1150–1156.

[25] Y. Mao, D. Dueri, M. Szmuk, and B. Açıkmeşe, "Successive convexification of non-convex optimal control problems with state constraints," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 4063 – 4069, 2017.

[26] R. Webster, *Convexity*. Oxford Univ. Press, 1994.

[27] A. Prékopa, "Logarithmic Concave Measures and Related Topics," *Stochastic Programming*, 1980.

[28] P. Billingsley, *Probability and measure*, 3rd ed. Wiley, 1995.

[29] Y. Chow and H. Teicher, *Probability Theory: Independence, Interchangeability, Martingales*, 3rd ed. Springer New York, 1997.

[30] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge Univ. Press, 2004.

[31] A. Kurzhanskiĭ and I. Vályi, *Ellipsoidal calculus for estimation and control*. Laxenburg, Austria, 1997.

[32] P. Gagarinov and A. A. Kurzhanskiy, "Ellipsoidal Toolbox."

[33] M. Herceg, M. Kvasnica, C. Jones, and M. Morari, "Multi-Parametric Toolbox 3.0," in *Proc. European Ctrl. Conf.*, 2013, pp. 502–510.

[34] M. Grant and S. Boyd, "CVX," Mar. 2014.

[35] D. Dueri, B. Açıkmeşe, D. Scharf, and M. Harris, "Customized real-time interior-point methods for onboard powered-descent guidance," *J. of Guidance, Ctrl., and Dyn.*, pp. 1–16, 2016.