2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Provably Safe Trajectory Optimization in the Presence of Uncertain Convex Obstacles

Charles Dawson, Ashkan Jasour, Andreas Hofmann, and Brian Williams

***Abstract*— Real-world environments are inherently uncertain, and to operate safely in these environments robots must be able to plan around this uncertainty. In the context of motion planning, we desire systems that can maintain an acceptable level of safety as the robot moves, even when the exact locations of nearby obstacles are not known. In this paper, we solve this *chance-constrained motion planning* problem using a sequential convex optimization framework. To constrain the risk of collision incurred by planned movements, we employ geometric objects called ϵ-shadows to compute upper bounds on the risk of collision between the robot and uncertain obstacles. We use these ϵ-shadow-based estimates as constraints in a nonlinear trajectory optimization problem, which we then solve by iteratively linearizing the non-convex risk constraints. This sequential optimization approach quickly finds trajectories that accomplish the desired motion while maintaining a user-specified limit on collision risk. Our method can be applied to robots and environments with arbitrary convex geometry; even in complex environments, it runs in less than a second and provides provable guarantees on the safety of planned trajectories, enabling fast, reactive, and safe robot motion in realistic environments.**

I. Introduction

In an ideal world, robots could trust the maps they use to navigate; however, localization uncertainty, sensor noise, human unpredictability, and other factors all ensure that few models survive contact with the real world. Robots must instead plan using uncertain models of their environment, optimizing performance while limiting the amount of risk incurred thereby. In the context of motion planning, this challenge takes the form of optimizing a planned trajectory while limiting the probability of collision with nearby obstacles (e.g. a human with unpredictable future actions).

Motion planning in the absence of uncertainty is a well-studied problem. Approaches such as sampling-based planning (RRT), optimization techniques (TrajOpt [1]), and combined sampling-optimization planners (as in [2]) have demonstrated impressive performance in a wide range of applications. There have been several attempts to apply similar techniques in the uncertain case [3], [4], [5], [6], [7], [8], including modified RRT algorithms [5], [4] and outer-loop optimization approaches [8], [6].

Unfortunately, many uncertainty-aware planners lack support for complex geometry in the robot or environment, limiting their usefulness in many applications (e.g. industrial pick-and-place or home robotics). When the robot can be approximated as a point and environment as a system of linear inequalities, collision risk can be estimated analytically, allowing for efficient path planning under uncertainty [3], [7], [9]. In these simplified environments, techniques such as mixed-integer or disjunctive mathematical programming have been applied to great effect [3].

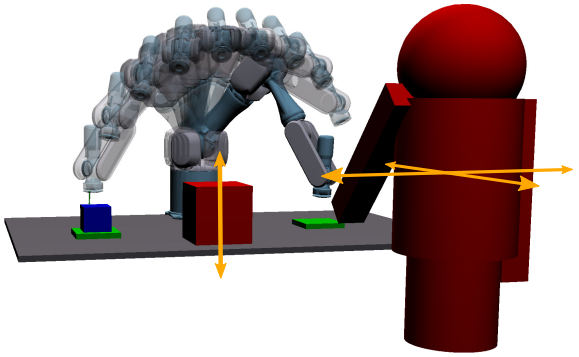

Fig. 1. An example of a chance-constrained trajectory for a pick-and-place task, where the robot navigates around uncertain obstacles (red, with arrows indicating directions in which the obstacles' locations are most uncertain).

The limitations of these approaches can be seen by imagining a mobile manipulator robot navigating a kitchen. Planning motions in this environment must account for uncertainty, as it is not feasible to precisely localize all of the obstacles that might be present (e.g. furniture, dishes, humans), and it must guarantee the safety of its users. Moreover, a point-robot model is not well-suited for this task, requiring approaches that deal with more complex geometry for both the robot and its environment. Analytical solutions generally do not exist beyond the point-robot case; instead, techniques for handling more complex geometry typically rely on computationally expensive sampling- or numerical integration-based methods for estimating collision risk and planning trajectories. These methods are prone to long solve times and often not guaranteed to be accurate (limiting the safety of such approaches) [10], [8]. For example, the approach of Dai et al., which employs an iterative optimization approach to chance-constrained motion planning, supports complex robot and environment geometry, but its reliance on sampling to measure collision risk leads to slow performance and a lack of safety guarantees [2].

In applications such as home robotics or collaborative manufacturing, uncertainty in the state of the environment often dominates any uncertainty in the robot's own state (simply because joint encoders are much more accurate than visual object-detection systems, for example). Furthermore,

The authors are with the Model-Based Embedded and Robotic Systems Lab at the Massachusetts Institute of Technology, Cambridge, MA 02139, USA {cbd, jasour, hofma, williams}@mit.edu. This work was supported by Airbus SE.

when developing robots to operate near humans, we seek to provide guarantees on system safety. As a result, there is a need for algorithms that can quickly plan motions through uncertain environments with rich geometry (as in Fig. 1) while providing strong guarantees on safety.

A. Contributions

This paper makes two main contributions. First, we present an algorithm for calculating differentiable estimates of the risk of collision between a robot and its environment, through the use of geometric objects called *ϵ-shadows* to certify upper bounds on collision risk. These risk estimates and gradients can be computed quickly even when the robot and environment have complex geometry; moreover, the ϵ-shadow formulation provides theoretical guarantees that the true risk of collision will never exceed the certified bound. We focus our approach on the case where the robot and environment can be represented as collections of convex shapes, and we restrict our analysis to the case where the positions of obstacles in the environment are subject to additive Gaussian uncertainty. These estimates may be used for real-time safety monitoring and certification that motion plans meet some specified threshold for safety, and the fact that our method yields differentiable risk estimates means that they are well-suited for use in chance-constrained optimization problems.

For our second contribution, we demonstrate the use of these risk certificates to optimize robot trajectories subject to a constraint on the risk of collision over the entire trajectory, employing a framework we call sequential convex optimization with risk allocation (SCORA). Using this framework, which we implement on top of existing sequential optimization solvers, we demonstrate path planning with bounded collision risk in simulated environments with non-trivial geometry. This optimization approach not only runs quickly ($< 0.5\,\mathrm{s}$ in our experiments) but provides formal guarantees on the safety of optimized trajectories.

B. Notation

In the following discussion, we use script symbols (e.g. $\mathcal{X}$, $\mathcal{O}$) to denote subsets of $\mathbb{R}^n$, such as the set of points occupied by one link of a robot or the set of points occupied by an obstacle. In the case when obstacles are subject to additive uncertainty in position, we denote the *nominal* geometry of the obstacle, located at the obstacle's expected position, as $O \subset \mathbb{R}^n$. Formally, the set of points $\mathcal{O}$ actually containing the uncertain obstacle is related to O by $\mathcal{O} = \{x + d : x \in O\}$, where the uncertain translation $d \sim \mathcal{N}(0, \Sigma)$ is a zero-mean multivariate Gaussian random variable with covariance Σ. Note that this formulation excludes uncertainty in the orientation of the obstacle (as is the case when d is correlated with x), but in many practical applications there is a strong prior belief on orientation (e.g. an object resting on a flat surface is likely to have a fixed orientation relative to that surface) that allows uncertainty in position to dominate. We will restrict our analysis to convex obstacle geometry, since non-convex geometries can be approximated as convex decompositions [11].

II. Risk certificates and gradient calculation

This section presents our method for estimating the collision risk between a robot and its environment and extends this approach to provide the gradient of these risk estimates as well. We begin by introducing the concept of an ϵ-shadow: a geometric object that is guaranteed to contain an uncertain obstacle with some probability. In particular, as in our previous work on risk estimation [12], we follow Axelrod, Kaelbling, and Lozano-Pérez in considering only *maximal* ϵ-shadows, which we define below [4].

Definition 1. (maximal ϵ-shadow) *A set $\mathcal{S} \subseteq \mathbb{R}^n$ is a maximal ϵ-shadow of an uncertain obstacle $\mathcal{O}$ if the probability $P(\mathcal{O} \subseteq \mathcal{S}) = 1 - \epsilon$.*

Intuitively, a maximal ϵ-shadow is a shape that completely contains an uncertain obstacle with some specified probability. If there exists an ϵ-shadow of an obstacle $\mathcal{O}$ that does not contact the robot, then the risk that the robot collides with that obstacle is guaranteed to be no greater than ϵ. Using these ϵ-shadows, the problem of finding an upper bound on collision risk reduces to the problem of finding a large ϵ-shadow for each obstacle (equivalently, a small upper bound ϵ) that does not intersect with the robot.

How one constructs such shadows varies based on how obstacle uncertainty is modeled. In their work on this subject, Axelrod, Kaelbling, and Lozano-Pérez model uncertain obstacles as the intersection of linear inequalities with normally-distributed coefficients [4]. This model naturally captures the uncertainty of point-cloud obstacles (e.g. LIDAR data), but for obstacles detected by other means (e.g. a visual pose estimation system) it is more natural to model the object as a known shape O with position subject to additive Gaussian noise. By choosing this uncertainty model, we can guarantee that the corresponding ϵ-shadows will be convex as long as the underlying obstacles are convex, enabling efficient collision checking (the same is not true for Axelrod, Kaelbling, and Lozano-Pérez's formulation, which offers efficient collision checking only for point-robot models). Here, we will present our approach for finding large ϵ-shadows that do not intersect the robot (thus certifying an upper bound on collision probability), then present an extension that also provides the gradient of collision risk with respect to robot state.

A. Estimating risk using ϵ-shadows

Recall that the uncertain obstacle $\mathcal{O}$ is related to the nominal geometry O by $\mathcal{O} = \{x + d : x \in O\}$, as shown in Fig. 2a, where $d \sim \mathcal{N}(0, \Sigma)$ is a zero-mean Gaussian random variable. Our goal is to construct a maximal ϵ-shadow for $\mathcal{O}$, which we do by first constructing a maximal ϵ-shadow for the uncertain translation d. By the properties of the multivariate Gaussian distribution, if we define the ellipsoid $\mathcal{D}_{\epsilon_1} = \left\{d : d^T \Sigma^{-1} d \leq \phi^{-1}(1 - \epsilon_1)\right\}$ (as in Fig. 2b), where ϕ^{-1} is the inverse cumulative distribution function (CDF) of

the chi-squared distribution with n degrees of freedom, then $P(d \in \mathcal{D}_{\epsilon_1}) = 1 - \epsilon_1$ [4]. Thus, $\mathcal{D}_{\epsilon_1}$ is a maximal ϵ-shadow of d.

Next, we define the set $\mathcal{S}_{\epsilon_1}$ as the Minkowski sum of the nominal obstacle shape and the ellipsoid $\mathcal{D}_{\epsilon_1}$, as illustrated in Fig. 2c, so that

$$\mathcal{S}_{\epsilon_1} = \{x + d : x \in O, d \in \mathcal{D}_{\epsilon_1}\} \tag{1}$$

Theorem 1. *$\mathcal{S}_{\epsilon_1}$ is a convex maximal ϵ_1-shadow of O.*

Proof: Intuitively, one can think of $\mathcal{D}_{\epsilon_1}$ (centered at some point in O) as an ϵ_1-shadow of that point in $\mathcal{O}$ (since we construct $\mathcal{O}$ by adding the uncertain translation d to each point in O). By taking the Minkowski sum of O and $\mathcal{D}_{\epsilon_1}$, we create a shape that contains every point in $\mathcal{O}$ with probability $1 - \epsilon_1$, and thus contains $\mathcal{O}$ with probability $1 - \epsilon_1$. Since the Minkowski sum of two convex shapes is itself convex, $\mathcal{S}_{\epsilon_1}$ is convex iff the nominal geometry O is convex ($\mathcal{D}_{\epsilon_1}$ is always convex).

More formally, we observe that the event that $\mathcal{S}_{\epsilon_1}$ contains $\mathcal{O}$ is equivalent to the event that for all $y \in \mathcal{O}$, there exists both an $x \in O$ and a $d \in \mathcal{D}_{\epsilon_1}$ such that $x + d = y$. By construction, every $y \in \mathcal{O}$ equals some $x + d$ for $x \in O$ and $d \sim \mathcal{N}(0, \Sigma)$, so we reduce the event $\mathcal{O} \subseteq \mathcal{S}_{\epsilon_1}$ to the event that $d \in \mathcal{D}_{\epsilon_1}; d \sim \mathcal{N}(0, \Sigma)$. Because of this equivalency, the probabilities of these events must be equal, so

$$P(\mathcal{O} \subseteq \mathcal{S}_{\epsilon_1}) = P(d \in \mathcal{D}_{\epsilon_1}; d \sim \mathcal{N}(0, \Sigma)) = 1 - \epsilon_1 \tag{2}$$

Thus, $\mathcal{S}_{\epsilon_1}$ is a maximal ϵ-shadow of $\mathcal{O}$. Furthermore, since the Minkowski sum of two convex sets is convex, all ellipsoids are convex, and O is convex by assumption, $\mathcal{S}_{\epsilon_1}$ is convex as well. This proof follows our previous treatment of the subject in [12]. □

Since $\mathcal{S}_{\epsilon_1}$ contains $\mathcal{O}$ with probability $1-\epsilon_1$, it follows that if the robot does not collide with $\mathcal{S}_{\epsilon_1}$, then there is *at most* ϵ_1 probability that the robot will collide with the uncertain obstacle itself. As a result, by finding large shadows $\mathcal{S}_{\epsilon_1}$ that do not touch the robot, we can certify an upper bound on the risk the robot incurs in any given configuration. To find the largest such $\mathcal{S}_{\epsilon_1}$ that does not contact the robot (correspondingly, the smallest upper bound on risk ϵ_1), we can employ a simple bisection line search. Because of the convexity of $\mathcal{S}_{\epsilon_1}$, we can apply linear-time algorithms for collision-checking, like the GJK algorithm [13], allowing us to conduct this search without excessive computational cost.

Unfortunately, Fig. 2d illustrates an issue with this simple bisection-search method: ϵ-shadows of this form tend to be very conservative. Because the robot is considered at risk of collision whenever the obstacle protrudes beyond its ϵ-shadow, even if it protrudes *away* from the robot, this simple ellipsoid-sum approach has a high false-positive rate.

To find a less conservative estimate that still certifies an upper bound on collision risk, we can exploit the geometric structure of the problem to refine our approach. As shown in Fig. 2d, there is often free space extending away from the robot where the ϵ-shadow can expand without intersecting the robot. We can take advantage of this free space by

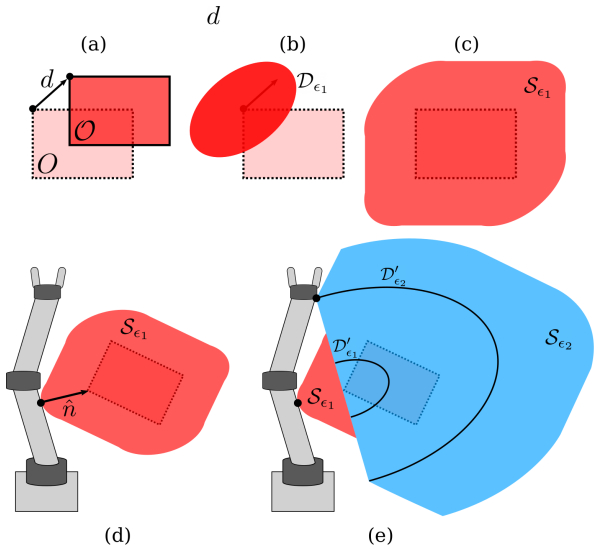

Fig. 2. An illustration of the ϵ-shadow approach to finding certifiable collision risk estimates. (a) The uncertain obstacle is translated from its nominal location by some uncertain vector. (b) We construct an ellipsoid to contain that translation vector with probability $1 - \epsilon_1$. (c) We sum that ellipsoid with the nominal geometry to produce the ϵ_1-shadow, guaranteed by Theorem 1 to contain the obstacle with probability $1 - \epsilon_1$. (d) We find the largest such shadow that does not intersect the robot, but this estimate is conservative. (e) By expanding the ϵ_1-shadow in the $\hat{\mathbf{n}}$ direction away from the robot, as described in Algorithm 1, we find a larger shadow $\mathcal{S}_{\epsilon_1} \cap \mathcal{S}_{\epsilon_2}$ that yields a less conservative estimate $(\epsilon_1 + \epsilon_2)/2$.

preferentially expanding the ϵ-shadow in that direction until a second intersection occurs, as shown in Fig. 2e.

To achieve this preferential expansion, we begin with a single ellipsoid-sum ϵ-shadow $\mathcal{S}_{\epsilon_1}$ that contacts the robot at one point and augment it with a second shadow constructed from a half-ellipsoid, as illustrated in Fig. 2e. If $\hat{\mathbf{n}}$ is the contact normal between $\mathcal{S}_{\epsilon_1}$ and the robot, we define the half-ellipsoid $\mathcal{D}'_{\epsilon_2} = \left\{d : d^T\Sigma^{-1}d \leq \phi^{-1}(1-\epsilon_2), \hat{\mathbf{n}}^T d \geq 0\right\}$ and corresponding convex shadow $\mathcal{S}_{\epsilon_2}$ (extending away from the robot in the $\hat{\mathbf{n}}$ direction) as the Minkowski sum

$$\mathcal{S}_{\epsilon_2} = \left\{x + d : x \in O, d \in \mathcal{D}'_{\epsilon_2}\right\} \tag{3}$$

We can find this expanded shadow using a second line search, which we warm-start with the results of the first search, as illustrated in Algorithm 1 Because of this warm-start, Algorithm 1 will always find some $\epsilon_2 \leq \epsilon_1 \leq 1$ ($\epsilon_2 = \epsilon_1$ only when there is no surrounding free space, and $\epsilon_1 = 1$ when the nominal geometry O is in collision with the robot even before uncertainty is considered). By taking the union of this second shadow $\mathcal{S}_{\epsilon_2}$ with the original $\mathcal{S}_{\epsilon_1}$, we obtain the larger shadow $\mathcal{S}_{\epsilon'} = \mathcal{S}_{\epsilon_1} \cup \mathcal{S}_{\epsilon_2}$ that provides a tighter bound $\epsilon' = (\epsilon_1 + \epsilon_2)/2$ on collision risk by capturing more of the collision-free workspace.

Theorem 2. *$\mathcal{S}_{\epsilon'}$ is a maximal ϵ'-shadow of $\mathcal{O}$, with $\epsilon' = (\epsilon_1 + \epsilon_2)/2$.*

Proof: The proof follows our approach in [12]. By the symmetry of the multivariate Gaussian (under reflection about the origin), we see that $\mathcal{D}'_{\epsilon_2}$ is a maximal $\epsilon_2/2$-shadow for the uncertain translation $d \sim \mathcal{N}(0, \Sigma)$, since

Algorithm 1: Multiple-expansion method for certifying bounds on robot-obstacle collision risk.

Input: Obstacle $\mathcal{O}$, covariance Σ, and tolerance ϵ_{tol}
Result: Risk estimate ϵ
Conduct bisection search to find smallest $\epsilon_1 \in (0,1)$ to precision ϵ_{tol} such that the ϵ_1-shadow of $\mathcal{O}$ constructed using Eq. (1) does not intersect robot;
$\hat{\mathbf{n}} \leftarrow$ unit vector from the robot into the obstacle at the point of collision;
Conduct bisection search to find smallest $\epsilon_2 \in (0, \epsilon_1)$ to precision ϵ_{tol} such that the ϵ_2-shadow of $\mathcal{O}$ constructed using Eq. (3) does not intersect robot;
$\epsilon \leftarrow (\epsilon_1 + \epsilon_2)/2$

$$P(d \in \mathcal{D}'_{\epsilon_2}) = P(d^T\Sigma^{-1}d \leq \phi^{-1}(1-\epsilon_2) \cap \hat{\mathbf{n}}^T d \geq 0) \quad (4)$$
$$= P(d^T\Sigma^{-1}d \leq \phi^{-1}(1-\epsilon_2))P(\hat{\mathbf{n}}^T d \geq 0) \quad (5)$$
$$= \epsilon_2 \frac{1}{2} \quad (6)$$

By the same reasoning used in Theorem 1, the sum $\mathcal{S}_{\epsilon_2}$ is a maximal $\epsilon_2/2$-shadow of $\mathcal{O}$. In combining $\mathcal{S}_{\epsilon_1}$ with $\mathcal{S}_{\epsilon_2}$, we simply need to account for the overlap between these two shadows. Since $\mathcal{O}$ is offset from O by $d \sim \mathcal{N}(0, \Sigma)$, we see that

$$P(\mathcal{O} \subseteq \mathcal{S}_{\epsilon'}) = P(d \in \mathcal{D}_{\epsilon_1}) + P(d \in \mathcal{D}'_{\epsilon_1}) - P(d \in \mathcal{D}_{\epsilon_1} \cap \mathcal{D}'_{\epsilon_2}) \quad (7)$$
$$= \epsilon_1 + \frac{\epsilon_1}{2} - P(d \in \mathcal{D}_{\epsilon_1} \cap \mathcal{D}'_{\epsilon_2}) \quad (8)$$

Since Algorithm 1 ensures $\epsilon_2 \leq \epsilon_1$, we can simplify the intersection $\mathcal{D}_{\epsilon_1} \cap \mathcal{D}'_{\epsilon_2}$ to $\mathcal{D}'_{\epsilon_1} = \left\{d : d^T\Sigma^{-1}d \leq \phi^{-1}(1-\epsilon_1), \hat{\mathbf{n}}^T d \geq 0\right\}$. Again, by the symmetry of the Gaussian distribution, we see that $P(d \in \mathcal{D}'_{\epsilon_1}) = \epsilon_1/2$, so $P(\mathcal{O} \subseteq \mathcal{S}_{\epsilon'}) = (\epsilon_1 + \epsilon_2)/2$, completing the proof. □

An important benefit of this risk estimation approach (compared to that used by Axelrod, Kaelbling, and Lozano-Pérez) is that both $\mathcal{S}_{\epsilon_1}$ and $\mathcal{S}_{\epsilon_2}$ are convex shapes, allowing fast collision checking between the ϵ-shadows and the robot, even when the robot has non-trivial geometry. Moreover, we can construct these shadows using implicit Minkowski sums (represented as support vector mappings), avoiding the computational cost of constructing set-wise sums. Support vector mappings represent convex shapes as functions taking directions to points on the shape furthest in that direction; they are commonly used in computational geometry [13].

B. Derivation of risk gradient

A key advantage of our technique (beyond its support of complex robot and environment geometry) is that these risk estimates are differentiable with respect to robot state, unlike other estimation methods in the literature (e.g. sampling [8]). The existence of this derivative will allow us to efficiently incorporate collision risk as a constraint in a nonlinear optimization framework.

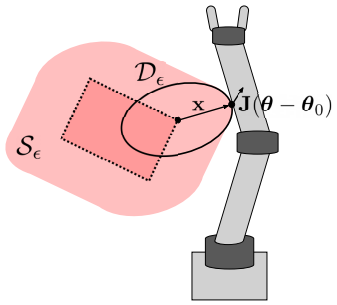

Fig. 3. To derive the gradient, we isolate the component ellipsoid that makes contact with the robot.

The basis for our derivation of the gradient is summarized in Fig. 3. Once we have found the largest ϵ-shadow (correspondingly, smallest ϵ) that lies tangent to the robot, we can isolate the ellipsoid $\mathcal{D} = \left\{d : d^T\Sigma^{-1}d \leq \phi^{-1}(1-\epsilon)\right\}$ that makes contact with the robot, centered at some point on O. For simplicity, we will consider just one shadow $\mathcal{S}_\epsilon$, but our approach can be extended to the combined shadow $\mathcal{S}_{\epsilon'} = \mathcal{S}_{\epsilon_1} \cup \mathcal{S}_{\epsilon_2}$, since the gradient of the combined risk estimate $\epsilon' = (\epsilon_1 + \epsilon_2)/2$ is simply the linear combination of the gradients of ϵ_1 and ϵ_2.

Since the size of this isolated ellipsoid $\mathcal{D}_\epsilon$ implicitly sets the risk estimate ϵ, we can examine how this ellipsoid changes with small perturbations to the robot state. We assume that these small joint angle perturbations $\delta\boldsymbol{\theta} \ll \boldsymbol{\theta}_0$ do not change the location of $\mathcal{D}_\epsilon$, for example by causing its center to move to another point on O.

Let $\mathbf{x}$ be the vector from the center of $\mathcal{D}_\epsilon$ to the point of contact. This vector can be calculated easily, as it is simply the support vector of $\mathcal{D}_\epsilon$ in the $-\hat{\mathbf{n}}$ direction. Due to the definition of $\mathcal{D}_\epsilon$,

$$\mathbf{x}^T\Sigma^{-1}\mathbf{x} = \phi^{-1}(1-\epsilon), \quad (9)$$

so we can express the risk estimate in terms of $\mathbf{x}$, as

$$\epsilon = 1 - \phi(\mathbf{x}^T\Sigma^{-1}\mathbf{x}). \quad (10)$$

This expression is differentiable, so we derive the gradient in terms of the chi-squared probability density function (PDF) with n degrees of freedom, χ^2_n,

$$\nabla_{\mathbf{x}}\epsilon = -\chi^2_n(\mathbf{x}^T\Sigma^{-1}\mathbf{x})\left(2\mathbf{x}^T\Sigma^{-1}\right) \quad (11)$$

By linearizing the robot's pose around its current joint state $\boldsymbol{\theta}_0$ and computing the Jacobian $\mathbf{J}$ at the point of contact, we can express the change in $\mathbf{x}$ in terms of a small change in joint state $\delta\boldsymbol{\theta}$ as

$$\delta\mathbf{x} = \mathbf{J}\delta\boldsymbol{\theta} \quad (12)$$

Combining Eqs. (11) and (12) yields the gradient of estimated risk with respect to the robot's joint state:

$$\nabla_{\boldsymbol{\theta}}\epsilon = -\chi^2_n(\mathbf{x}^T\Sigma^{-1}\mathbf{x})\left(2\mathbf{x}^T\Sigma^{-1}\right)\mathbf{J} \quad (13)$$

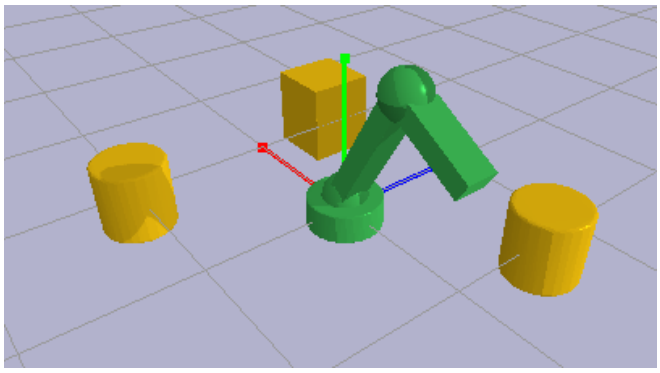

Fig. 4. The experimental setup used to benchmark our proposed risk-estimation algorithm. The robot is shown in green, with obstacles in yellow.

Using this gradient, we can linearize the risk of collision with each obstacle for a given robot configuration $\boldsymbol{\theta}_0$ (where ϵ_0 is the risk estimated for that configuration):

$$\epsilon(\boldsymbol{\theta}) \approx \epsilon_0 - \chi^2_n(\mathbf{x}^T\Sigma^{-1}\mathbf{x})\left(2\mathbf{x}^T\Sigma^{-1}\right)\mathbf{J}\left(\boldsymbol{\theta}-\boldsymbol{\theta}_0\right) \quad (14)$$

Since $\mathbf{x}$ can be calculated quickly using support-vector geometry and $\mathbf{J}$ can be queried from any external kinematics engine, this gradient can be constructed with relatively little computational effort. With this linearization in hand, we can employ sequential convex optimization to find near-locally optimal trajectories that maintain desired risk levels.

C. Performance

Our focus in this paper is on incorporating these differentiable risk certificates into a trajectory optimization framework; however, these certificates are broadly useful in other applications (such as real-time safety monitoring). As a result, it is important to benchmark the performance of these risk certificates relative to other methods of estimating collision risk, so as to better evaluate the merits of this technique as a modular component for broader application.

We implement the risk estimation algorithm presented in Section II using the C++ Bullet collision checking library [14]. In addition, we compare our approach to a naïve Monte Carlo approach using $10,000$ random samples per obstacle and the surface integral technique presented by Park, Park, and Manocha in [10]. To evaluate the performance of these techniques, we used the setup shown in Fig. 4, where the uncertain obstacles are shown in yellow. The robot (shown in green) is composed of 5 shapes. The position of the rightmost and leftmost cylindrical obstacles are subject to Gaussian uncertainty with covariance $\Sigma_1 = (0.05\,\mathrm{m})I$ and $\Sigma_2 = (0.1\,\mathrm{m})I$, respectively, where I is the 3×3 identity matrix. The box obstacle is subject to Gaussian positional uncertainty with covariance

$$\Sigma_3 = \begin{bmatrix} 0.2 & 0 & 0.1 \\ 0 & 0.1 & 0 \\ 0.1 & 0 & 0.2 \end{bmatrix}.$$

All experiments in this section were run on one core of an i7-8565U CPU running at $1.8\,\mathrm{GHz}$.

The results of comparing these algorithms are shown in Table I, including runtimes averaged over $10,000$ trials. From these results, we see that both our proposed approach and other risk-estimation algorithms offer greatly improved runtime over a simple sampling-based approach, albeit at the cost of increased conservatism. However, our proposed approach outperforms the existing state-of-the-art by an order of magnitude in runtime without a corresponding increase in conservatism. This significant increase in performance is attributable to our approach's efficient use of computational geometry techniques. The surface integral approach proposed in [10] explicitly constructs a Minkowski sum of two shapes (requiring $O(n^2)$ time with respect to the number of vertices in each shape), then computes an integral over the faces of that shape. In contrast, our approach constructs the Minkowski sum implicitly (a constant time operation), then uses a collision-checking algorithm ($O(n)$ time with respect to the number of vertices [13]), making $O(\log \epsilon_{prec})$ calls to this collision checker (where ϵ_{prec} is the desired precision in the estimated risk ϵ). Both approaches yield risk estimates guaranteed to upper-bound the true risk of collision, but our algorithm is notably more efficient. As an additional benefit, the gradient of the risk estimate produced by our method can be obtained with only a small constant-time overhead.

TABLE I
COMPARISON OF RISK ESTIMATION ALGORITHMS

Algorithm	Runtime (µs, per obstacle)[1]	Estimated collision risk
ϵ-shadows (proposed)	370.4	0.1159
Surface integral [10]	4031.0	0.1204
Monte Carlo[2]	55207.3	0.0033

[1] Averaged over $10,000$ trials.
[2] Using $10,000$ independent samples.

These risk-estimation algorithms are broadly useful in safety monitoring and certification contexts, where real-time performance is necessary. Furthermore, as we discuss in the next section, they are also useful in the context of chance-constrained motion planning. In this context, fast performance is imperative since the constraints will be evaluated many times as the motion planning problem is solved, and a slow risk estimation algorithm will act as a drag on the performance of the entire motion planning system. As a result, the significantly improved performance of our approach, along with the easily availability of a gradient (which is not a feature of other approaches in the literature), will bring corresponding benefits to the motion-planning application.

III. Sequential Convex Trajectory Optimization

Optimizing the trajectory of a robot moving around obstacles is a challenging non-convex optimization problem, even in the absence of uncertainty. Furthermore, complex robot and environment geometries make it difficult to apply traditional linear or quadratic programming methods, many

of which use a point-robot approximation [3]. Instead, deterministic trajectory optimizers that support complex geometry, such as TrajOpt [1], solve this problem using sequential convex optimization (SCO).

At a basic level, SCO, seeks to solve the non-convex optimization problem

$$\begin{aligned} &\text{minimize} && f(\mathbf{x}) && (15)\\ &\text{subject to} && g_i(\mathbf{x}) \leq 0, \quad i = 1, 2, \ldots, n_{ineq} && (16)\\ & && h_j(\mathbf{x}) = 0, \quad j = 1, 2, \ldots, n_{eq} && (17) \end{aligned}$$

where f, g_i, and h_j are (possibly non-convex) objective, inequality constraint, and equality constraint functions, respectively, and $\mathbf{x}$ is a vector of decision variables (typically joint angles $\boldsymbol{\theta}_t$ at each of T timesteps). Since f, g_i, and h_j can be non-convex, SCO repeatedly constructs and optimizes a convex approximation of Problem (15) until the true non-convex constraints are satisfied and the solution converges to a local optimum of f. Some SCO solvers (including TrajOpt) also dualize the constraints by incorporating a cost penalty for constraint violations, which improves performance when initialized with an infeasible solution [1].

In our work, we extend the SCO approach to consider constraints on the probability of collision in addition to deterministic collision-avoidance constraints. In the absence of uncertainty, one avoids collision by constraining the signed distance $\text{sd}_O(\boldsymbol{\theta}_t)$ between the robot and each obstacle O to be greater than some fixed safety margin at each timestep [1]. When the obstacles' locations are uncertain, we need to consider not only the risk of collision at each timestep but also how risk accumulates over the entire trajectory. In practice, we would like to limit the risk incurred by the robot over the entire plan rather than merely limiting risk at each timestep (this type of constraint is referred to as a joint chance constraint). We enforce the joint chance constraint by adding the additional constraint (21), which allows the optimizer to intelligently *allocate* risk across timesteps, taking more risk at some points and less at others in order to satisfy the joint risk threshold. In previous work on chance-constrained optimization, this allocation has been shown to improve the performance of risk-aware systems, as it allows the system to spend and save risk at different times as needed to achieve its objective [3], [9].

To formalize this approach, which we call SCORA (sequential convex optimization with risk allocation), we denote the estimated risk of collision with obstacle O in state $\boldsymbol{\theta}_t$ as $\epsilon_O(\boldsymbol{\theta}_t)$, which we incorporate into the optimization problem:

$$\begin{aligned} &\text{minimize} \quad \sum_{t=1}^{T-1} \|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-1}\|^2 \quad \text{subject to} && (18)\\ &\text{sd}_O(\boldsymbol{\theta}_t) \geq d_{margin}, \quad t = 0, \ldots, T-1; \ \forall O && (19)\\ &\sum_O \epsilon_O(\boldsymbol{\theta}_t) \leq \delta_t, \qquad t = 0, \ldots, T-1 && (20)\\ &\sum_{t=0}^{T-1} \delta_t \leq \Delta && (21) \end{aligned}$$

Our decision variables in this problem are the joint angles at each timestep $\boldsymbol{\theta}_t$ and the risk allocations δ_t. We assume trajectory tracking is accomplished using a low-level controller and do not consider dynamics. Since the risk of collision at each timestep is bounded by δ_t, the risk of collision during the entire plan cannot exceed $\sum_{t=0}^{T-1} \delta_t$. Thus, constraints (20) and (21) ensure that the joint chance constraint is met while providing the freedom to take on more or less risk as needed to achieve good performance. We retain the constraint on signed distance in our formulation because the collision risk estimates saturate at 1 when the robot is in contact with the nominal obstacle geometry, so this constraint is needed to penalize contact and escape risk-saturated configurations.

Although the objective (18) and joint chance constraint (21) are convex functions of the decision variables $\boldsymbol{\theta}_t$ and δ_t, the functions $\text{sd}(\boldsymbol{\theta})$ and $\epsilon(\boldsymbol{\theta})$ are non-convex in general, so we need to construct a convex approximation of Problem (18) about the current solution $\boldsymbol{\theta}_{t,0}$ and δ_t. We can linearize the individual chance constraints (20) according to Eq. (14) and the signed distance function (19) according to [1], yielding the convex approximation:

$$\begin{aligned} &\text{minimize} \quad \sum_{t=1}^{T-1} \|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-1}\|^2 \quad \text{subject to} && (22)\\ &\text{sd}(\boldsymbol{\theta}_{t,0}) + \hat{\mathbf{n}}^T \mathbf{J} (\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t,0}) \geq d_{margin}, \\ &\qquad\qquad t = 0, \ldots, T-1; \ \forall O && (23)\\ &\sum_O \left[\epsilon_O(\boldsymbol{\theta}_{t,0}) - \chi^2_n(\mathbf{x}^T \Sigma^{-1} \mathbf{x}) \left(2\mathbf{x}^T \Sigma^{-1}\right) \mathbf{J} (\boldsymbol{\theta} - \boldsymbol{\theta}_{t,0})\right]\Big|_O \\ &\qquad\qquad \leq \delta_t, \quad t = 0, \ldots, T-1 && (24)\\ &\sum_{t=0}^{T-1} \delta_t \leq \Delta && (25) \end{aligned}$$

This approximation is a quadratic program that can be solved quickly using an off-the-shelf convex optimizer. Of course, additional linear (or convexifiable) constraints can be added to this formulation if desired (e.g. to enforce joint angle limits).

IV. Results

We implement our SCORA optimization framework using TrajOpt's built-in SCO solver, to which we add our risk estimation, gradient, and allocation methods [1]. All experiments were run on an Intel i9-7960X CPU. We tested our approach in the scenarios shown in Figs. 5 and 6 (where orange arrows indicate uncertainty in obstacle location, with standard deviations ranging between 7-30 cm). We set a joint chance constraint of 1% and 10% for the scenarios in Figs. 5 and 6, respectively.

To benchmark our approach, we provide two comparisons. The first is with a trajectory optimized without any chance constraints. This risk-blind comparison provides a baseline for both the length of the optimal collision-free path and the time needed to find that path, allowing us to quantify the cost of imposing chance constraints in later examples.

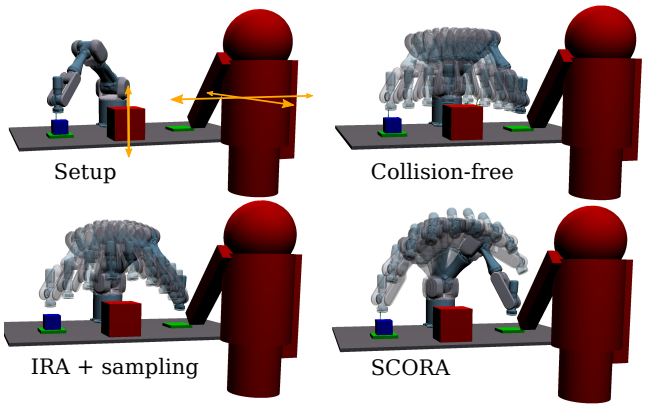

Fig. 5. Optimized trajectories in our example scenario. The orange arrows indicate directions in which obstacle locations are uncertain. Our SCORA approach yields the most conservative trajectory, but it is the only approach that satisfies the 1% collision risk constraint.

The second comparison is with a trajectory optimized using the iterative risk allocation (IRA) method presented presented by Dai et al. in [8]. This approach employs a two-layer optimization, where an outer loop uses sampling to approximate the risk of collision at each step of a trajectory and optimize the allocation of risk to each step, and an inner loop repeatedly re-solves a deterministic optimization with safety margins d_{margin} adjusted based on those allocations. The IRA algorithm presented in [8] focuses on uncertainty in robot state, but we adapt this approach to consider obstacle uncertainty instead. This method provides a means for comparing SCORA against a previously-published chance-constrained trajectory optimization algorithm.

Our results are shown in Table II. The collision-free optimization was seeded with straight lines in joint space, while IRA and SCORA were seeded with the output of the collision-free optimization.

TABLE II

COMPARISON OF TRAJECTORY OPTIMIZATION ALGORITHMS

	Algorithm	Runtime[1] (s)	Path length[1] (rad)	Collision risk[2]
Fig. 5	Risk-blind	0.005	2.35	59.16%
$\Delta = 1\%$	IRA	0.480	2.65	20.85%
$T = 10$	SCORA	0.195	4.56	0.48%
Fig. 6	Risk-blind	0.154	7.21	18.87%
$\Delta = 10\%$	IRA	2.202	7.22	9.22%
$T = 17$	SCORA	0.358	7.72	1.65%

[1] Averaged over $1,000$ trials.
[2] Averaged over $100,000$ trials.

Since neither scenario is a "piano-mover" problem where very few feasible paths exist, we see that the risk-blind optimization quickly finds a nominally collision-free trajectory in both cases. Unsurprisingly, this trajectory is not robust to uncertainty in obstacle position. The IRA approach is somewhat more robust, but its reliance on sampling to estimate collision risk means that even though the algorithm converges, it often significantly underestimates the true risk of collision. In particular, the sampling strategy it employs

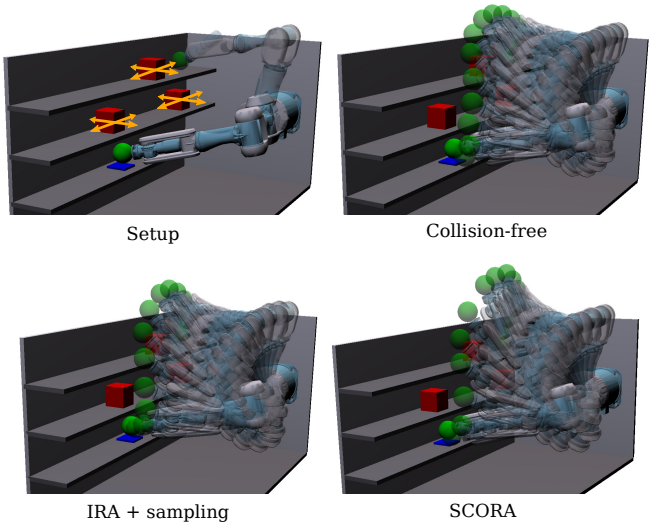

Fig. 6. Optimized trajectories with a 10% collision risk constraint in a pick-and-place task. Our SCORA approach not only satisfies the chance constraint but also runs much faster than the IRA algorithm.

cannot accurately measure low probabilities. As a result, IRA successfully achieves the 10% risk bound but fails to achieve the 1% bound.

In contrast, the SCORA approach proposed here not only satisfies the chance constraint in both cases (due to the formal guarantees of the ϵ-shadow method) but also converges 2.5-6 times more quickly than IRA. Our approach yields longer, more conservative trajectories, in part due to the conservatism of the ϵ-shadow bound, but this performance-safety trade-off is typical in risk-aware systems, as the robot can improve its performance by taking on more risk.

V. CONCLUSIONS

In this paper, we develop a sequential convex optimization approach for solving the chance-constrained motion planning problem. To quickly estimate the probability of collision between the robot and uncertain obstacles, we use ϵ-shadows to simultaneously certify an upper bound on collision risk and derive the gradient of that risk with respect to robot state. This gradient allows us to construct a convex approximation of the chance-constrained trajectory optimization problem, which we solve efficiently using sequential convex optimization with risk allocation (SCORA).

We demonstrate our approach in simulation, yielding planning times under $0.5\,\mathrm{s}$ (2.5-6 times faster than previously-published approaches) while ensuring that the optimized trajectory respects the user-specified risk bound. Because ϵ-shadow risk certificates are guaranteed never to underestimate the true risk of collision, our approach produces provably safe motion plans even in the presence of obstacles with rich geometry and uncertain location.

REFERENCES

[1] J. Schulman, J. Ho, A. X. Lee, I. Awwal, H. Bradlow, and P. Abbeel, "Finding Locally Optimal, Collision-Free Trajectories with Sequential Convex Optimization," in *Robotics: Science and Systems*, 2013.

[2] S. Dai, M. Orton, S. Schaffert, A. Hofmann, and B. Williams, "Improving Trajectory Optimization Using a Roadmap Framework," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2018, pp. 8674–8681.

[3] L. Blackmore, M. Ono, and B. C. Williams, "Chance-Constrained Optimal Path Planning With Obstacles," *IEEE Transactions on Robotics*, vol. 27, no. 6, pp. 1080–1094, Dec. 2011.

[4] B. Axelrod, L. P. Kaelbling, and T. Lozano-Pérez, "Provably safe robot navigation with obstacle uncertainty," *The International Journal of Robotics Research*, vol. 37, no. 13-14, pp. 1760–1774, Dec. 2018.

[5] B. Luders, M. Kothari, and J. How, "Chance constrained RRT for probabilistic robustness to environmental uncertainty," in *AIAA Guidance, Navigation, and Control Conference*, 2010, p. 8160.

[6] J. van den Berg, P. Abbeel, and K. Goldberg, "LQG-MP: Optimized path planning for robots with motion uncertainty and imperfect state information," *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 895–913, June 2011.

[7] W. Sun, L. G. Torres, J. van den Berg, and R. Alterovitz, "Safe Motion Planning for Imprecise Robotic Manipulators by Minimizing Probability of Collision," in *Robotics Research: The 16th International Symposium ISRR*, 2016, pp. 685–701.

[8] S. Dai, S. Schaffert, A. Jasour, A. G. Hofmann, and B. C. Williams, "Chance Constrained Motion Planning for High-Dimensional Robots," *2019 International Conference on Robotics and Automation (ICRA)*, pp. 8805–8811, 2018.

[9] M. Ono and B. C. Williams, "Iterative Risk Allocation: A new approach to robust Model Predictive Control with a joint chance constraint," in *2008 47th IEEE Conference on Decision and Control*, Dec. 2008, pp. 3427–3432.

[10] J. S. Park, C. Park, and D. Manocha, "Efficient probabilistic collision detection for non-convex shapes," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 1944–1951.

[11] K. Mamou and F. Ghorbel, "A simple and efficient approach for 3d mesh approximate convex decomposition," in *2009 16th IEEE International Conference on Image Processing (ICIP)*, Nov 2009, pp. 3501–3504.

[12] C. Dawson, A. Hofmann, and B. C. Williams, "Fast Certification of Collision Probability Bounds with Uncertain Convex Obstacles," in *Robotics: Science and Systems*, Submitted 2020, Under Review.

[13] E. Gilbert, D. Johnson, and S. Keerthi, "A fast procedure for computing the distance between complex objects in three-dimensional space," *IEEE Journal on Robotics and Automation*, vol. 4, no. 2, pp. 193–203, Apr. 1988.

[14] E. Coumans, "Bullet physics engine," https://pybullet.org/.