Robotics: Science and Systems 2021
Held Virtually, July 12–16, 2021

Convex Risk Bounded Continuous-Time Trajectory Planning in Uncertain Nonconvex Environments

Ashkan Jasour*, Weiqiao Han*, and Brian Williams
MIT, Computer Science and Artificial Intelligence Laboratory
{jasour,weiqiaoh,williams}@mit.edu

Abstract—**In this paper, we address the trajectory planning problem in uncertain nonconvex static and dynamic environments that contain obstacles with probabilistic location, size, and geometry. To address this problem, we provide a risk bounded trajectory planning method that looks for continuous-time trajectories with guaranteed bounded risk over the planning time horizon. Risk is defined as the probability of collision with uncertain obstacles. Existing approaches to address risk bounded trajectory planning problems either are limited to Gaussian uncertainties and convex obstacles or rely on sampling-based methods that need uncertainty samples and time discretization. To address the risk bounded trajectory planning problem, we leverage the notion of risk contours to transform the risk bounded planning problem into a deterministic optimization problem. Risk contours are the set of all points in the uncertain environment with guaranteed bounded risk. The obtained deterministic optimization is, in general, nonlinear and nonconvex time-varying optimization. We provide convex methods based on sum-of-squares optimization to efficiently solve the obtained nonconvex time-varying optimization problem and obtain the continuous-time risk bounded trajectories without time discretization. The provided approach deals with arbitrary probabilistic uncertainties, nonconvex and nonlinear, static and dynamic obstacles, and is suitable for online trajectory planning problems.**

I. Introduction

In order for robots to navigate safely in the real world, they need to plan safe trajectories to avoid static and moving obstacles, such as humans and vehicles, under perception uncertainties. The motion planning problem in dynamic environments is known to be computationally hard [1]. In this paper, we address the trajectory planning problem in uncertain nonconvex static and dynamic environments that contain obstacles with probabilistic location, size, and geometry. In this problem, the time-varying, nonconvex, and probabilistic nature of the obstacle-free safe regions makes the trajectory planning problem challenging.

Several approaches have been proposed to address the trajectory planning problems. In the absence of obstacles, one can use standard convex optimization to look for polynomial trajectories that satisfy boundary and way-points conditions [2]–[4]. In the presence of obstacles, sampling-based methods, including rapidly exploring random tree (RRT) and probabilistic roadmap (PRM), and virtual potential field methods are widely used to find obstacle-free trajectories [2], [5]. In [6], a mixed-integer optimization is provided for trajectory planning in the presence of convex obstacles. The proposed method first uses convex segmentation to compute convex regions of obstacle-free space. Then, it uses a mixed-integer optimization to assign polynomial trajectories to the computed convex safe regions. Also, [7] provides a moment-sum-of-squares-based convex optimization to obtain piece-wise linear trajectories in the presence of deterministic time-varying polynomial obstacles without the need for time discretization.

Trajectory planning problems under uncertainty look for trajectories with a bounded probability of collision with uncertain obstacles. Existing methods to address trajectory planning problems under uncertainty either are limited to Gaussian uncertainties and convex obstacles [8]–[14] or rely on sampling-based methods [15]–[17]. For example, chance constrained RRT* algorithm in [11] assumes Gaussian uncertainties and linear obstacles and performs a probabilistic collision check for the nodes of the search tree. Hence, it can not guarantee to satisfy the probabilistic safety constraints along the edges of the search tree. The Monte Carlo-based motion planning algorithms, e.g., [17], use a large number of uncertainty samples to estimate the probability of collision of a given trajectory. Sampling-based methods do not provide any analytical bounds on the probability of collision and, due to a large number of samples, can be computationally intractable.

Also, [18], [19] use moment-based approaches to address non-Gaussian uncertainties in motion planning problems in the presence of convex obstacles. More precisely, to obtain the risk bounded trajectories, [18] uses first and second-order moments of uncertainties and RRT* algorithm in the presence of linear obstacles and [19] uses higher-order moments and interior-point nonlinear optimization solvers in the presence of ellipsoidal obstacles.

Statement of Contributions: In this paper, we propose novel convex algorithms for risk bounded continuous-time trajectory planning in uncertain nonconvex static and dynamic environments that contain obstacles with probabilistic location, size, geometry, and trajectories with arbitrary probabilistic distributions. To achieve risk bounded plans:

1) We provide an analytical method to compute risk contours maps. Risk contours allow us to identify risk bounded regions in uncertain environments and transform nonlinear stochastic planning problems into deterministic standard planning problems, in the presence of arbitrary probabilistic uncertainties. Hence, standard deterministic motion planning algorithms, e.g., RRT*, PRM, can be employed to look for

This work was partially supported by the Boeing grant 6943358. *These authors contributed equally to the paper.

safe (risk-bounded) plans.

2) To solve the obtained deterministic planning problems, we provide two planners including i) sum-of-squares-based RRT algorithm and ii) sum-of-squares-based convex optimization that allows us to theoretically look for global optimal plans in nonconvex environments.

3) We also provide continuous-time safety guarantees in stochastic environments. To ensure safety, existing planning under uncertainty algorithms only verify the safety of a finite set of waypoints (time-discretization). Unlike the existing planners, the provided planners of this paper ensure the (risk bounded) safety of the continuous-time trajectories without the need for time discretization.

The outline of the paper is as follows: Section II presents the notation adopted in the paper and definitions of polynomials, moments, and sum-of-squares optimization. In Section III, we provide the problem formulation of risk bounded continuous-time trajectory planning. Section IV provides analytical approaches to compute static and dynamic risk contours defined for static and dynamic uncertain obstacles to identify the risk bounded safe regions in uncertain environments. In Section V, using the obtained risk contours, we provide sum-of-squares-based planners to look for continuous-time risk bounded trajectories in uncertain static and dynamic environments. In Section VI, we present experimental results on the risk bounded planning problems of autonomous and robotic systems followed by a discussion section. Finally, concluding remarks and future work are given in Section VII.

II. Notation and Definitions

This section covers notation and some basic definitions of polynomials, moments of probability distributions, and sum-of-squares optimization. For a vector $\mathbf{x} \in \mathbb{R}^n$ and multi-index $\alpha \in \mathbb{N}^n$, let $x^\alpha = \prod_{i=1}^n x_i^{\alpha_i}$.

Polynomials: Let $\mathbb{R}[x]$ be the set of real polynomials in the variables $\mathbf{x} \in \mathbb{R}^n$. Given polynomial $\mathcal{P}(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$, we represent $\mathcal{P}$ as $\sum_{\alpha \in \mathbb{N}^n} p_\alpha x^\alpha$ where $\{x^\alpha\}_{\alpha \in \mathbb{N}^n}$ are the standard monomial basis of $\mathbb{R}[x]$, $\mathbf{p} = \{p_\alpha\}_{\alpha \in \mathbb{N}^n}$ denotes the coefficients, and $\alpha \in \mathbb{N}^n$. In this paper, we use polynomials to describe uncertain obstacles and continuous-time trajectories. For example the set $\{(x_1, x_2) : 1 - (x_1 - \omega_1)^2 - (x_2 - \omega_2)^2 \geq 0\}$ represents a circle-shaped obstacle whose center is subjected to uncertainty modeled with random variables (ω_1, ω_2). Also, $\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \end{bmatrix} t + \begin{bmatrix} 0.5 \\ 2 \end{bmatrix} t^2$, $t \in [0, 1]$ is an example of polynomial trajectory of order 2 in 2D environment between the points $\mathbf{x}(0) = (0, 0)$ and $\mathbf{x}(1) = (1.5, 1)$.

Moments of Probability Distributions: Moments of random variables are the generalization of mean and covariance and are defined as expected values of monomials of random variables. More precisely, given $(\alpha_1, ..., \alpha_n) \in \mathbb{N}^n$ where $\alpha = \sum_{i=1}^n \alpha_i$, moment of order α of random vector ω is defined as $\mathbb{E}[\Pi_{i=1}^n \omega_i^{\alpha_i}]$. For example, sequence of the moments of order $\alpha = 2$ for $n = 3$ is defined as $\left[\mathbb{E}[w_1^2], \mathbb{E}[\omega_1\omega_2], \mathbb{E}[\omega_1\omega_3], \mathbb{E}[\omega_2^2], \mathbb{E}[\omega_2\omega_3], \mathbb{E}[\omega_3^2]\right]$. Moments of random variables can be easily computed using the characteristic function of probability distributions [20]. We will use a finite sequence of the moments to represent non-Gaussian probability distributions.

Sum of Squares Polynomials and Optimization: In this paper, we will use sum of squares (SOS) techniques to solve nonconvex optimization problems of the risk bounded trajectory planning problems. Polynomial $\mathcal{P}(x)$ is a sum of squares polynomial if it can be written as a sum of *finitely* many squared polynomials, i.e., $\mathcal{P}(x) = \sum_{j=1}^m h_j(x)^2$ for some $m < \infty$ and $h_j(x) \in \mathbb{R}[x]$ for $1 \leq j \leq m$. SOS condition, i.e., $\mathcal{P}(x) \in SOS$, can be represented as a convex constraint of the form of a linear matrix inequality (LMI) in terms of the coefficients of the polynomial, i.e., $\mathcal{P}(x) \in SOS \to \mathcal{P}(x) = \mathbf{x}^T A \mathbf{x}$ where $\mathbf{x}$ is the vector of standard monomial basis and A is a positive semidefinite matrix in terms of the coefficients of the polynomial [21]–[23]. One can use different software packages like Yalmip [24] and Spotless [25] to check the SOS condition of the polynomials.

Sum of squares polynomials are used to obtain the convex relaxations of noncovex polynomial optimization problems [21]–[23]. More precisely, consider the following noncovex polynomial optimization as:

$$\begin{aligned} \underset{\mathbf{x} \in \mathbb{R}^n}{\text{minimize}} \quad & f(\mathbf{x}) \\ \text{subject to} \quad & g_i(\mathbf{x}) \geq 0, \ i = 1, ..., m \end{aligned} \tag{1}$$

where f and $g_i, i = 1, ..., m$ are polynomial functions. We can rewrite the polynomial optimization in (1) as the following form:

$$\begin{aligned} \underset{\gamma \in \mathbb{R}}{\text{maximize}} \quad & \gamma \\ \text{subject to} \quad & f(\mathbf{x}) - \gamma \geq 0, \ \forall \mathbf{x} \in \{\mathbf{x} \in \mathbb{R}^n : g_i(\mathbf{x}) \geq 0 \ |_{i=1}^m\} \end{aligned} \tag{2}$$

where we look for the best lower bound of the function $f(\mathbf{x})$ denoted by γ inside the feasible set of the original optimization problem. Hence, if $\mathbf{x}^*$ is the optimal solution of the original optimization problem in (1), then $\gamma^* = f(\mathbf{x}^*)$ is the optimal solution of the optimization problem in (2).

Note that in optimization problem (2), objective function is linear and the constraint is the nonnegativity condition of polynomial $f(\mathbf{x}) - \gamma$. Such nonnegativity condition can be replaced by SOS conditions of polynomials [21]–[23]. Hence, we can transform the optimization in (2) into a convex optimization, i.e., semidefinite program, with LMI constraints in terms of the coefficients of the polynomials of the original optimization in (1). Also, we can recover the optimal solution of the original polynomial optimization in (1), i.e., $\mathbf{x}^*$, using the solution of the dual convex optimization problem of (2) as shown in [22], [23]. One can use different software packages like GloptiPoly [26] to solve the polynomial optimization in (1) using SOS-based primal-dual approach.

Recently, SOS optimization techniques have been extended to address time-varying polynomial optimization problems of the form

$$\begin{aligned} \underset{\mathbf{x} \in \mathbb{R}^n}{\text{minimize}} \quad & f(\mathbf{x}) \\ \text{subject to} \quad & g_i(\mathbf{x}, t) \geq 0, \ \forall t \in [t_0, t_f], \ i = 1, ..., m \end{aligned} \tag{3}$$

where one needs to make sure that time-varying constraints are satisfied over the given time horizon $t \in [t_0, t_f]$. SOS-based techniques can be used to solve the time-varying optimization problem in (3) without the need for time discretization by transforming the problem into a convex optimization, i.e., time-varying semidefinite program [7], [27], [28].

III. Problem Formulation

Suppose $\mathcal{X} \in \mathbb{R}^{n_x}$ is an uncertain environment and the sets $\mathcal{X}_{obs_i}(\omega_i) \subset \mathcal{X},\ i = 1,...,n_{o_s}$ and $\mathcal{X}_{obs_i}(\omega_i, t) \subset \mathcal{X},\ i = n_{o_s}+1,...,n_{o_d}$ are the static and dynamic uncertain obstacles, respectively, where $\omega_i \in \mathbb{R}^{n_\omega},\ i = 1,...,n_{o_s}+n_{o_d}$ are probabilistic uncertain parameters with known probability distributions. We represent static uncertain obstacles in terms of polynomials in $\mathbf{x} \in \mathcal{X}$ and uncertain parameters as follows:

$$\mathcal{X}_{obs_i}(\omega_i) = \{\mathbf{x} \in \mathcal{X} : \mathcal{P}_i(\mathbf{x}, \omega_i) \geq 0\},\ i = 1,...,n_{o_s} \tag{4}$$

where $\mathcal{P}_i : \mathbb{R}^{n_x+n_\omega} \to \mathbb{R}, i = 1,...,n_{o_s}$ are the given polynomials. Similarly, we represent dynamic uncertain obstacles in terms of polynomials in $\mathbf{x} \in \mathcal{X}$, time t, and uncertain parameters as follows:

$$\mathcal{X}_{obs_i}(\omega_i, t) = \{\mathbf{x} \in \mathcal{X} : \mathcal{P}_i(\mathbf{x}, \omega_i, t) \geq 0\}\ |_{i=n_{o_s}+1}^{n_{o_d}} \tag{5}$$

where $\mathcal{P}_i : \mathbb{R}^{n_x+n_\omega+1} \to \mathbb{R}, i = n_{o_s}+1,...,n_{o_d}$ are the given polynomials. Note that, in general, the sets in (4) and (5) represent *nonconvex* probabilistic obstacles, e.g., nonconvex obstacles with uncertain size, location, or geometry [29], [30].

Given static and dynamic uncertain obstacles in (4) and (5), we define risk as the probability of collision with uncertain obstacles in the environment. In the risk bounded continuous-time trajectory planning problem, we aim at finding a continuous-time trajectory $\mathbf{x}(t) : [t_0, t_f] \to \mathbb{R}^{n_x}$ defined over the time horizon $t \in [t_0, t_f]$ between the start and final points $\mathbf{x}_0$ and $\mathbf{x}_f$ such that the probability of collision of the trajectory $\mathbf{x}(t)$ with uncertain obstacles is bounded.

More precisely, we define the risk bounded continuous-time trajectory planning problem as the following probabilistic optimization problem:

$$\underset{\mathbf{x}(t):[t_0,t_f]\to\mathbb{R}^{n_x}}{\text{minimize}} \int_{t_0}^{t_f} \|\dot{\mathbf{x}}(t)\|_2^2\, dt \tag{6}$$

$$\text{subject to} \quad \mathbf{x}(t_0) = \mathbf{x}_0,\ \mathbf{x}(t_f) = \mathbf{x}_f \tag{6a}$$

$$\text{Prob}\left(\mathbf{x}(t) \in \mathcal{X}_{obs_i}(\omega_i)\right) \leq \Delta,\ \forall t \in [t_0, t_f]\ |_{i=1}^{n_{o_s}} \tag{6b}$$

$$\text{Prob}\left(\mathbf{x}(t) \in \mathcal{X}_{obs_i}(\omega_i, t)\right) \leq \Delta,\ \forall t \in [t_0, t_f]\ |_{i=n_{o_s}+1}^{n_{o_d}} \tag{6c}$$

where objective function (6) is the length of the trajectory $\mathbf{x}(t)$ defined in terms of ℓ_2 norm $\|.\|_2$. Also, constraints (6b) and (6c) are the defined risks at time t for trajectory $\mathbf{x}(t)$ in terms of the uncertain static and dynamic obstacles, respectively. Moreover, $0 \leq \Delta \in \mathbb{R} \leq 1$ is the given acceptable risk level. To solve the risk bounded optimization problem in (6), we will look for the following continuous-time trajectories:

i) Polynomial trajectories over the planning horizon $[t_0, t_f]$ of the form

$$\mathbf{x}(t) = \sum_{\alpha=0}^{d} \mathbf{c}_\alpha t^\alpha,\ t \in [t_0, t_f] \tag{7}$$

where $\mathbf{c}_\alpha \in \mathbb{R}^{n_x},\ \alpha = 0,...,d$ are the coefficient vectors,

ii) Piece-wise linear trajectories of the form

$$\mathbf{x}_i(t) = \mathbf{a}_i + \mathbf{b}_i t,\ t \in [t_{i-1}, t_i),\ i = 1,...,s \tag{8}$$

where s is the number of linear pieces defined over the time intervals $t \in [t_{i-1}, t_i]$ of the form $t_{i-1} = t_0 + \frac{(i-1)(t_f-t_0)}{s}$ and $t_i = t_0 + \frac{i(t_f-t_0)}{s}, i = 1,...,s$, and $\mathbf{a}_i, \mathbf{b}_i \in \mathbb{R}^{n_x}$ are the coefficient vectors.

Solving the probabilistic optimization in (6) is challenging, because i) we need to deal with multivariate integrals of the probabilistic constraints in (6b) and (6c) defined over the nonconvex sets of the obstacles, ii) we need to deal with time-varying constraints to ensure that they are all satisfied over the entire planning time horizon $[t_0, t_f]$, and iii) optimization in (6) is, in general, nonconvex optimization; Hence, we cannot guarantee to obtain the global optimal solution.

In this paper, we provide a systematic numerical procedure to efficiently solve the probabilistic optimization problem in (6) in the presence of nonconvex uncertain obstacles with arbitrary probability distributions. For this purpose, we will leverage the notion of risk contours to transform the probabilistic optimization in (6) into a deterministic polynomial optimization problem and use SOS optimization techniques to obtain optimal continuous-time trajectories with guaranteed bounded risk.

IV. Risk Contours

In [29], we define the risk contour with respect to the static uncertain obstacle in (4) and the given acceptable risk level Δ in (6b) as the set of all points in the environment whose probability of collision with the uncertain obstacle is less or equal to Δ. In this paper, we use static and dynamic risk contours defined for static and dynamic uncertain obstacles, respectively, to identify the safe regions in uncertain environments, i.e., the feasible set of optimization (6).

In [29], to construct the risk contours of uncertain static obstacles, we propose an (n_x+n_ω)-dimensional convex optimization in the form of a semidefinite program (SDP). Such optimization is not suitable for online computations and is limited to small dimensions (n_x+n_ω). In this paper, we propose an optimization-free fast approach, i.e., an analytical method, to construct the risk contours both for static and dynamic uncertain obstacles and show how one can use the obtained risk contours to solve the risk-bounded trajectory planning problem in (6).

A. Static Risk Contours

Let $\mathcal{X}_{obs}(\omega) = \{\mathbf{x} \in \mathcal{X} : \mathcal{P}(\mathbf{x}, \omega) \geq 0\}$ be the given static uncertain obstacle as defined in (4) and $\Delta \in [0, 1]$ be the given acceptable risk level. Then, static Δ-risk contour denoted by $\mathcal{C}_r^\Delta$ is defined as the set of all points in the environment, i.e., $\mathbf{x} \in \mathcal{X}$, whose risk is less or equal to Δ. More precisely,

$$\mathcal{C}_r^\Delta := \{\ \mathbf{x} \in \mathcal{X} :\ \text{Prob}(\mathbf{x} \in \mathcal{X}_{obs}(\omega)) \leq \Delta\} \tag{9}$$

The main idea to construct the static risk contour in (9) is to replace the probabilistic constraint, i.e., $\text{Prob}(\mathbf{x} \in \mathcal{X}_{obs}(\omega)) =$

$\text{Prob}(\mathcal{P}(\mathbf{x},\omega) \geq 0) \leq \Delta$, with a deterministic constraint in terms of $\mathbf{x}$. In this paper, we provide an analytical method as follows:

Given the polynomial $\mathcal{P}(\mathbf{x},\omega)$ of the uncertain obstacle $\mathcal{X}_{obs}(\omega)$, we define the set $\hat{\mathcal{C}}_r^{\Delta}$ as follows:

$$\hat{\mathcal{C}}_r^{\Delta} = \left\{ \mathbf{x} \in \mathcal{X} : \begin{array}{c} \frac{\mathbb{E}[\mathcal{P}^2(\mathbf{x},\omega)] - \mathbb{E}[\mathcal{P}(\mathbf{x},\omega)]^2}{\mathbb{E}[\mathcal{P}^2(\mathbf{x},\omega)]} \leq \Delta, \\ \mathbb{E}[\mathcal{P}(\mathbf{x},\omega)] \leq 0 \end{array} \right\} \tag{10}$$

where the expectation is taken with respect to the distribution of uncertain parameter ω. Note that we can compute polynomials $\mathbb{E}[\mathcal{P}^2(\mathbf{x},\omega)]$ and $\mathbb{E}[\mathcal{P}(\mathbf{x},\omega)]$ in terms of $\mathbf{x}$ and known moments of ω. More precisely, $\mathbb{E}[\mathcal{P}^2(\mathbf{x},\omega)]$ and $\mathbb{E}[\mathcal{P}(\mathbf{x},\omega)]$ are polynomials in $\mathbf{x}$ whose coefficients are defined in terms of the moments of ω and the coefficients of polynomial $\mathcal{P}(\mathbf{x},\omega)$.

The following result holds true.

Theorem 1: The set $\hat{\mathcal{C}}_r^{\Delta}$ in (10) is an inner approximation of the static risk contour $\mathcal{C}_r^{\Delta}$ in (9).

Proof: See Appendix. ■

Remark 1: The set in (10) is a *rational* polynomial-based inner approximation of the risk contour in (9). It also uses higher order moments of the uncertain parameter ω up to order $2d$ where d is the order of the polynomial obstacle $\mathcal{P}(\mathbf{x},\omega)$.

Note that since $\hat{\mathcal{C}}_r^{\Delta}$ is an inner approximation of $\mathcal{C}_r^{\Delta}$, any trajectory $\mathbf{x}(t) \in \hat{\mathcal{C}}_r^{\Delta}$ $\forall t \in [t_0, t_f]$ is guaranteed to have a risk less or equal to Δ. We now provide an illustrative example to show the performance of the proposed method to construct the static Δ-risk contours and benchmark our method against the optimization-based approach in [29].

Illustrative Example 1: Consider the following illustrative example where $\mathcal{X} = [-1,1]^2$. The set $\mathcal{X}_{obs}(\omega) = \{(x_1,x_2) : \omega^2 - x_1^2 - x_2^2 \geq 0\}$ represents a circle-shaped obstacle whose radius ω has a uniform probability distribution over $[0.3, 0.4]$, [29]. Moment of order α of a uniform distribution defined over $[l,u]$ can be described in a closed-form as $\frac{u^{\alpha+1} - l^{\alpha+1}}{(u-l)(\alpha+1)}$. To construct the static Δ-risk contour, we compute polynomials $\mathbb{E}[\mathcal{P}(\mathbf{x},\omega)]$ and $\mathbb{E}[\mathcal{P}^2(\mathbf{x},\omega)]$ using the polynomial obstacle and the moments of ω as follows:

$$\begin{aligned}
\mathbb{E}[\mathcal{P}(\mathbf{x},\omega)] &= \mathbb{E}[\omega^2 - x_1^2 - x_2^2] = \mathbb{E}[\omega^2] - x_1^2 - x_2^2 = 0.1233 - x_1^2 - x_2^2 \\
\mathbb{E}[\mathcal{P}^2(\mathbf{x},\omega)] &= \mathbb{E}[(\omega^2 - x_1^2 - x_2^2)^2] \\
&= \mathbb{E}[\omega^4] - 2\mathbb{E}[\omega^2]x_1^2 - 2\mathbb{E}[\omega^2]x_2^2 + x_1^4 + 2x_1^2x_2^2 + x_2^4 \\
&= 0.0156 - 0.2466x_1^2 - 0.2466x_2^2 + x_1^4 + 2x_1^2x_2^2 + x_2^4
\end{aligned}$$

As shown in Figures 1 and 2, we use the sublevels of functions $\frac{\mathbb{E}[\mathcal{P}^2(\mathbf{x},\omega)] - \mathbb{E}[\mathcal{P}(\mathbf{x},\omega)]^2}{\mathbb{E}[\mathcal{P}^2(\mathbf{x},\omega)]}$ and $\mathbb{E}[\mathcal{P}(\mathbf{x},\omega)]$ as in (10) to construct the inner approximations of the static Δ-risk contours for different risk levels $\Delta = [0.2, 0.1, 0.07, 0.05]$. We also compare our proposed method in (10) with the optimization-based method in [29] as shown in Figure 2. Using the provided SDP in [29], we obtain a polynomial of order 20 to describe the inner approximations of the Δ-risk contours. We note that the proposed analytical method of this paper provides a tight inner approximation of the Δ-risk contours. This is primarily due to the facts that i) the proposed analytical approach results in a *rational* polynomial representation of the risk-contours as opposed to a *standard* polynomial representation provided in [29] and ii) with the provided analytical approach, we are able to avoid the numerical issues that arise when solving large-scale SDPs [31]. The proposed method of this paper is also suitable for online large scale planning problems.

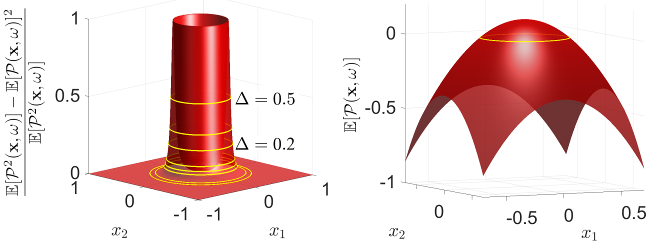

Fig. 1. Illustrative example 1: Intersection of Δ-sublevel set of function $\frac{\mathbb{E}[\mathcal{P}^2(\mathbf{x},\omega)] - \mathbb{E}[\mathcal{P}(\mathbf{x},\omega)]^2}{\mathbb{E}[\mathcal{P}^2(\mathbf{x},\omega)]}$ and 0-sublevel set of function $\mathbb{E}[\mathcal{P}(\mathbf{x},\omega)]$ describes the static Δ-risk contour as in (10).

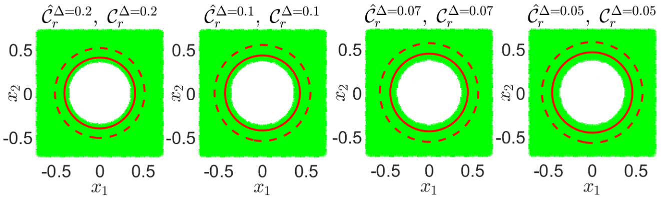

Fig. 2. Illustrative example 1: True static Δ-risk contour $\mathcal{C}_r^{\Delta}$ (green) and inner approximation $\hat{\mathcal{C}}_r^{\Delta}$ obtained using i) the proposed analytical method in (10) (outside of the solid-line) and ii) the proposed optimization-based method in ([29], Fig.4) (outside of the dashed-line). While the optimization-based method in [29] uses a standard polynomial of order 20, the proposed analytical method in (10) uses 4th order rational and 2nd order standard polynomials to describe the risk contours.

B. Dynamic Risk Contours

Let $\mathcal{X}_{obs}(\omega, t) = \{\mathbf{x} \in \mathcal{X} : \mathcal{P}(\mathbf{x},\omega,t) \geq 0\}$ be the given dynamic uncertain obstacle as defined in (5) and $\Delta \in [0,1]$ be the given acceptable risk level in (6c). Then, dynamic Δ-risk contour at time t denoted by $\mathcal{C}_r^{\Delta}(t)$ is defined as the set of all points in the environment, i.e., $\mathbf{x} \in \mathcal{X}$, whose risk at time t is less or equal to Δ. More precisely,

$$\mathcal{C}_r^{\Delta}(t) := \{ \ \mathbf{x} \in \mathcal{X} : \ \text{Prob}(\mathbf{x} \in \mathcal{X}_{obs}(\omega,t)) \leq \Delta\} \tag{11}$$

Similar to the static risk contours, we can replace the probabilistic constraint in (11) with deterministic constraints and construct an inner approximation of the dynamic Δ-risk contour denoted by $\hat{\mathcal{C}}_r^{\Delta}(t)$ as follows:

$$\hat{\mathcal{C}}_r^{\Delta}(t) = \left\{ \mathbf{x} \in \mathcal{X} : \begin{array}{c} \frac{\mathbb{E}[\mathcal{P}^2(\mathbf{x},\omega,t)] - \mathbb{E}[\mathcal{P}(\mathbf{x},\omega,t)]^2}{\mathbb{E}[\mathcal{P}^2(\mathbf{x},\omega,t)]} \leq \Delta, \\ \mathbb{E}[\mathcal{P}(\mathbf{x},\omega,t)] \leq 0 \end{array} \right\} \tag{12}$$

Note that dynamic Δ-risk contour (12) is described in terms of the time-varying constraints.

Illustrative Example 2: Consider the following illustrative example where $\mathcal{X} = [-1,1]^2$. The set $\mathcal{X}_{obs}(\omega, t) =$

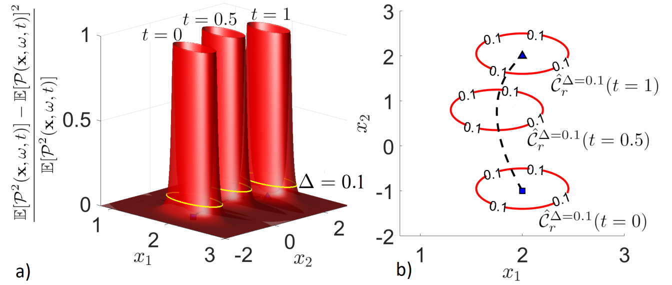

Fig. 3. Illustrative example 2: a) Function $\frac{\mathbb{E}[\mathcal{P}^2(\mathbf{x},\omega,t)] - \mathbb{E}[\mathcal{P}(\mathbf{x},\omega,t)]^2}{\mathbb{E}[\mathcal{P}^2(\mathbf{x},\omega,t)]}$ at time steps $t = 0, 0.5, 1$, b) Dynamic Δ-risk contours for $\Delta = 0.1$ at time steps $t = 0, 0.5, 1$ described in (12). Dashed line shows the expected value of the given uncertain trajectory, i.e., $\mathbb{E}[(p_{x_1}(t,\omega_2), p_{x_2}(t,\omega_3))]$. At each time t, for any point inside $\hat{\mathcal{C}}_r^{\Delta}(t)$ (outside of the closed curve), probability of collision with the moving uncertain obstacle is less or equal to $\Delta = 0.1$.

$\{(x_1, x_2) : \omega_1^2 - (x_1 - p_{x_1}(t,\omega_2))^2 - (x_2 - p_{x_2}(t,\omega_3))^2 \geq 0\}$ represents a moving circle-shaped obstacle with uncertain radius ω_1 and uncertain trajectories $p_{x_1}(t,\omega_2) = 2 - t + t^2 + 0.2\omega_2,\ p_{x_2}(t,\omega_3) = -1 + 4t - t^2 + 0.1\omega_3$ that describe the uncertain motion of the obstacle over the time horizon $t \in [0,1]$. Uncertain parameters have uniform, normal, and Beta distributions as $\omega_1 \sim Uniform[0.3, 0.4]$, $\omega_2 \sim \mathcal{N}(0, 0.1)$, $\omega_3 \sim Beta(3,3)$. Moment of order α of a Beta distribution with parameters (a, b) and a normal distribution with mean μ and standard deviation σ can be described in closed-forms as $y_\alpha = \frac{a+\alpha-1}{a+b+\alpha-1} y_{\alpha-1}, y_0 = 1$ and $y_\alpha = \sigma^\alpha(-\sqrt{-1}\sqrt{2})^\alpha kummerU(\frac{-\alpha}{2}, \frac{1}{2}, \frac{-\mu^2}{2\sigma^2})$, respectively, where $kummerU$ is "confluent hyper-geometric Kummer U" function. Similar to illustrative example 1, we compute the polynomials $\mathbb{E}[\mathcal{P}^2(\mathbf{x},\omega,t)]$ and $\mathbb{E}[\mathcal{P}(\mathbf{x},\omega,t)]$ using the moments of the uncertain parameters $\omega_i, i = 1,2,3$ and the polynomial obstacle. We then construct the dynamic Δ-risk contour as a function of time as described in (12). Figure 3 shows the obtained dynamic Δ-risk contours for $\Delta = 0.1$ at time steps $t = 0, 0.5, 1$ along the given uncertain trajectory $(p_{x_1}(t,\omega_2),\ p_{x_2}(t,\omega_3))$.

Remark 2: We can use (10) and (12) to construct static and dynamic risk contours in real-time. Therefore, standard motion planning algorithms such as RRT*, PRM, and motion primitive-based methods can be used for real-time risk bounded motion planning. To accomplish this, one just needs to use the safe regions, i.e, risk contours, to construct the trajectories.

In the next section, we provide *continuous-time* planning algorithms to look for trajectories with guaranteed bounded risk over the entire planning time horizon without the need for *time discretization*.

V. Continuous-Time Risk Bounded Trajectory Planning Using Risk Contours

In this section, we will use the static and dynamic risk contours to solve the continuous-time risk bounded trajectory planning problem defined in (6). More precisely, we use the static and dynamic risk contours to transform the probabilistic optimization in (6) into a deterministic polynomial optimization. The obtained deterministic polynomial optimization is, in general, nonconvex and nonlinear. In addition, we need to ensure that all the obtained deterministic constraints are satisfied over the entire planning time horizon $[t_0, t_f]$. In this section, we provide convex methods based on SOS techniques introduced in Section II to efficiently solve the obtained nonconvex time-varying deterministic planning optimization problem. While the existing planners rely on time discretization to verify the planning safety constraints, the provided SOS-based planners look for continuous-time trajectories with guaranteed bounded risk over the entire planning time horizon without the need for time discretization.

We first begin by addressing the continuous-time risk bounded trajectory planning in static uncertain environments using the static risk contours. We then use the dynamic risk contours to address the planning problems in dynamic uncertain environments.

A. Planning in Static Uncertain Environments

In this section, we are concerned with continuous-time risk bounded trajectory planning in the presence of static uncertain obstacles of the form (4). More precisely, we aim at solving the probabilistic optimization problem in (6) in the presence of static uncertain obstacles $\mathcal{X}_{obs_i}(\omega_i), i = 1, ..., n_{o_s}$, i.e.,

$$\underset{\mathbf{x}(t):[t_0,t_f]\rightarrow\mathbb{R}^{n_x}}{\text{minimize}} \int_{t_0}^{t_f} \|\dot{\mathbf{x}}(t)\|_2^2\, dt \tag{13}$$

$$\text{subject to}\quad \mathbf{x}(t_0) = \mathbf{x}_0,\ \mathbf{x}(t_f) = \mathbf{x}_f \tag{13a}$$

$$\text{Prob}\left(\mathbf{x}(t) \in \mathcal{X}_{obs_i}(\omega_i)\right) \leq \Delta,\ \forall t \in [t_0, t_f],\ i = 1, ..., n_{o_s} \tag{13b}$$

To obtain the deterministic polynomial optimization, we replace probabilistic constraints (13b) with deterministic constraints in terms of the static Δ-risk contours as follows:

$$\underset{\mathbf{x}(t):[t_0,t_f]\rightarrow\mathbb{R}^{n_x}}{\text{minimize}} \int_{t_0}^{t_f} \|\dot{\mathbf{x}}(t)\|_2^2\, dt \tag{14}$$

$$\text{subject to}\quad \mathbf{x}(t_0) = \mathbf{x}_0,\ \mathbf{x}(t_f) = \mathbf{x}_f \tag{14a}$$

$$\mathbf{x}(t) \in \hat{\mathcal{C}}_{r_i}^{\Delta},\ \forall t \in [t_0, t_f],\ i = 1, ..., n_{o_s} \tag{14b}$$

where $\hat{\mathcal{C}}_{r_i}^{\Delta}$ is the static Δ-risk contour of the static uncertain obstacle $\mathcal{X}_{obs_i}(\omega_i)$. The set of $\hat{\mathcal{C}}_{r_i}^{\Delta}, i = 1, ..., n_{o_s}$ represents the inner approximation of the feasible set of the probabilistic optimization in (13). The obtained deterministic optimization in (14) is time-varying optimization problem where we need to ensure to satisfy the constraints over the entire planning time horizon $[t_0, t_f]$.

To solve the deterministic optimization in (14), we will look for i) polynomial trajectory defined in (7) and ii) piecewise linear trajectory defined in (8). By substituting the polynomial trajectory $\mathbf{x}(t) = \sum_{\alpha=0}^{d} \mathbf{c}_\alpha t^\alpha$ in optimization (14), we obtain a deterministic optimization with constraints in terms of the coefficients of the polynomial trajectory $\mathbf{x}(t)$, i.e., $(1-\Delta)\mathbb{E}[\mathcal{P}^2(\sum_{\alpha=0}^{d} \mathbf{c}_\alpha t^\alpha, \omega)] - \mathbb{E}[\mathcal{P}(\sum_{\alpha=0}^{d} \mathbf{c}_\alpha t^\alpha, \omega)]^2 \leq 0$,

$\mathbb{E}[\mathcal{P}(\sum_{\alpha=0}^{d} \mathbf{c}_\alpha t^\alpha, \omega)] \leq 0.$ Similarly, by substituting the piece-wise linear trajectory of (8), we obtain the objective function $\sum_{j=1}^{s} \int_{t_{j-1}}^{t_j} \|\dot{\mathbf{x}}_j(t)\|_2^2 \, dt$ and constraints $\mathbf{a}_j + \mathbf{b}_j t \in \hat{\mathcal{C}}_{r_i}^{\Delta}, \ \forall t \in [t_{j-1}, t_j], \ i = 1, ..., n_{o_s}, j = 1, ..., s$. To solve the obtained time-varying deterministic polynomial optimization problems, we will provide 3 methods based on SOS optimization techniques as follows:

1) Time-Varying SOS Optimization: we use time-varying SOS optimization, introduced in Section II, to solve the time-varying deterministic polynomial optimization in (14). In implementation, we use the heuristic algorithm introduced by [7].

2) Standard SOS Optimization: In this method, we obtain a standard polynomial optimization of the form (1) by eliminating time t. We then use the standard SOS optimization technique in (2) to solve the obtained optimization problem. To achieve this, let $\mathcal{X}_{obs}(\omega) = \{\mathbf{x} \in \mathcal{X} : \mathcal{P}(\mathbf{x}, \omega) \geq 0\}$ be the given static uncertain obstacle defined in (4) and $\mathbf{x}(t)$ be the polynomial trajectory in (7). To eliminate time t, instead of using the instant risk as in (13b), we work with the average risk defined as $\frac{1}{t_f - t_0} \int \int_{\{(t,\omega):\mathcal{P}(\mathbf{x}(t),\omega)\geq 0\}} pr(\omega) d\omega dt$ where $pr(\omega)$ is the probability density function of ω. This is equivalent to treating t as a random variable with a uniform probability distribution over the planning time horizon $[t_0, t_f]$. We should note that the average risk is a *weak* safety measure, which means that just by bounding the average risk, we cannot guarantee to satisfy the constraints of the probabilistic optimization in (13).

By defining the average risk, we substitute the trajectory $\mathbf{x}(t)$ in the probabilistic constraint and follow the same steps as in Section IV-A to obtain an upper bound of the average risk and construct the set of all coefficients $\mathbf{c}_\alpha|_{\alpha=0}^{d}$ that results in a risk bounded trajectory of the form $\mathbf{x}(t) = \sum_{\alpha=0}^{d} \mathbf{c}_\alpha t^\alpha$. More precisely, we obtain the following set for the coefficients:

$$\left\{ \mathbf{c}_\alpha|_{\alpha=0}^{d} : \begin{array}{c} (1-\Delta)\mathbb{E}[\mathcal{P}^2(\sum_{\alpha=0}^{d} \mathbf{c}_\alpha t^\alpha, \omega)] - \mathbb{E}[\mathcal{P}(\sum_{\alpha=0}^{d} \mathbf{c}_\alpha t^\alpha, \omega)]^2 \leq 0, \\ \mathbb{E}[\mathcal{P}(\sum_{\alpha=0}^{d} \mathbf{c}_\alpha t^\alpha, \omega)] \leq 0 \end{array} \right\}$$

where the expectation is taken with respect to the probability distributions of ω and t. We can obtain a similar set for the coefficients of the piece-wise linear trajectories, as well. By computing such sets for coefficients of the trajectories, we can transform the probabilistic optimization problem into a deterministic standard polynomial optimization and use the standard SOS optimization technique in (2) to obtain the optimal solution.

3) RRT-SOS: In this method, we use sampling-based motion planning algorithms like RRT to construct the risk bounded trajectory of the deterministic polynomial optimization in (14). To ensure safety along the edges of the RRT, we use an SOS-based continuous-time technique to verify the constraints of the optimization in (14b) as follows:

Let $\mathbf{x}(t) = \sum_{\alpha=0}^{d} \mathbf{c}_\alpha t^\alpha, t \in [t_1, t_2]$ be the given trajectory between the two samples $\mathbf{x}_1 \in \mathcal{X}$ and $\mathbf{x}_2 \in \mathcal{X}$ in the uncertain environment. Also, let $\mathcal{S} = \{\mathbf{x} : g_i(x) \geq 0 \quad i = 1, ..., \ell\}$ be the feasible set of optimization (14), i.e., the set constructed by the polynomial constraints of all the risk contours $\hat{\mathcal{C}}_{r_i}^{\Delta}, i = 1, ..., n_{o_s}$. Then, the following result holds true:

Polynomial $\mathbf{x}(t)$ satisfies the safety constraints of the deterministic optimization in (14) over the time interval $[t_1, t_2]$, i.e., $\mathbf{x}(t) \in \mathcal{S}$ for all $t \in [t_1, t_2]$, if and only if polynomials $g_i(\mathbf{x}(t)), i = 1, ..., \ell$ take the following SOS representation:

$$g_i(\mathbf{x}(t)) = \sigma_{0i}(t) + \sigma_{1i}(t)(t - t_1) + \sigma_{2i}(t)(t_2 - t) \ |_{i=1}^{\ell} \tag{15}$$

where $\sigma_{0i}(t), \sigma_{1i}(t), \sigma_{2i}(t), i = 1, ..., \ell$ are SOS polynomials with appropriate degrees [22], [23], [32]. Yalmip [24] and Spotless [25] packages can be used to check the SOS condition (15) for the given trajectory $\mathbf{x}(t)$.

Remark 3: The complexity of the safety SOS condition in (15) is independent of the size of the planning time horizon $[t_1, t_2]$ and the length of the polynomial trajectory $\mathbf{x}(t)$. Hence, one can use (15) to easily verify the safety of trajectories in uncertain environments over the long planning time horizons.

The safety condition in (15) can be used in any sampling-based motion planning algorithms to verify the safety of the trajectories between the sample points. In this paper, we will use the following naive RRT algorithm: To expand the RRT, we use a linear trajectory to connect the given sample point to the closest vertex in the tree if the linear trajectory satisfies the SOS condition in (15), i.e., this implies that the linear trajectory between the two points is inside the risk contours in (14b). As a termination condition, we also check the safety of the linear trajectory between the selected safe sample and the goal points via the SOS condition in (15). If the linear trajectory satisfies SOS condition (15), we connect the sample point to the goal point; Hence, a feasible trajectory between the start and goal points can be constructed.

To improve the obtained feasible trajectory, we perform the following steps: i) given the obtained tree, we construct a graph, e.g., PRM, whose nodes are the vertex of the tree and edges of the graph are all the linear trajectories between the nodes that satisfy the SOS condition in (15), ii) we then perform a shortest path algorithm, e.g., Dijkstra algorithm, to obtain a path from the start to the goal point. We can also use smart initialization to guide the RRT-SOS algorithm and improve the run-time. For example, we use the straight line between the start and goal points to initialize the RRT-SOS algorithm. We then perform the sampling in the neighborhood of the given initial path and incrementally increase the size of the neighborhood until a feasible trajectory is obtained.

Illustrative Example 3: Consider the uncertain obstacle in illustrative example 1. We want to find a risk bounded trajectory between the points $\mathbf{x}(0) = [-1, -1]$ and $\mathbf{x}(1) = [1, 1]$ by solving the probabilistic optimization in (13) with $\Delta = 0.1$. For this purpose, we solve the deterministic optimization problem in (14) with respect to the 0.1-risk contour of the uncertain obstacle using the discussed 3 methods as shown in Figure 4. More precisely, we use standard SOS optimization

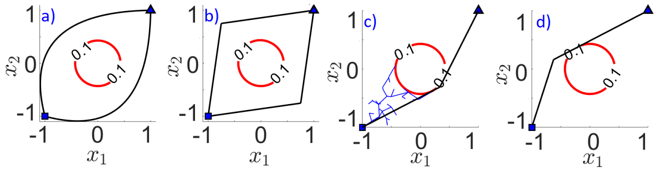

Fig. 4. Illustrative Example 3: a) risk bounded polynomial trajectories of order 2 obtained via standard SOS optimization, b) risk bounded piece-wise linear trajectories obtained via standard SOS optimization, c) risk bounded piece-wise linear trajectory obtained via RRT-SOS algorithm, d) risk bounded piece-wise linear trajectory obtained via time-varying SOS optimization.

to obtain a polynomial trajectory of order 2 and also a piece-wise linear trajectory with 2 pieces. Using GloptiPoly package, we extract two risk bounded trajectories between the given start and goal points as shown in Figure 4-a and 4-b. We also use RRT-SOS algorithm and time-varying SOS optimization to obtain piece-wise linear trajectories as shown in Figure 4-c and 4-d, respectively. We note that the standard SOS optimization-based method results in conservative trajectories as shown in Figure 4-a and 4-b.

We compare our proposed methods with Monte Carlo-based risk bounded RRT algorithm that uses uncertainty samples and time discretization to look for the risk bounded trajectories. To verify the safety of the edges in the RRT, we use only 10 uncertainty samples and 20 uniformly sampled way-points on the edges. Such RRT algorithm is significantly slower and also does not provide any guaranteed risk bounded trajectories.

The run-time for the standard SOS optimization to obtain the piece-wise linear and polynomial trajectories are both roughly 1.5 seconds. Also, the run-time for the time-varying SOS optimization is roughly 2.2 seconds (for more information see Section VI-E). The run-time for the RRT-SOS algorithm to obtain feasible and optimal trajectories are roughly 9.3 and 92.1 seconds, respectively. The run-time for the Monte Carlo-based risk bounded RRT algorithm to obtain feasible and optimal trajectories are roughly 215.12 and 3471.4 seconds, respectively. Also, continuous-time safety verification of each edge in the RRT-SOS algorithm via (15) takes roughly 0.1 seconds while the sampling-based safety verification in the Monte Carlo-based risk bounded RRT algorithm takes roughly 3 seconds.

B. Planning in Dynamic Uncertain Environments

In this section, we are concerned with continuous-time risk bounded trajectory planning in the presence of dynamic uncertain obstacles of the form (5). More precisely, we aim at solving the optimization problem in (6) in the presence of dynamic uncertain obstacles $\mathcal{X}_{obs_i}(\omega_i, t), i = 1, ..., n_{o_d}$, i.e.,

$$\underset{\mathbf{x}(t):[t_0,t_f]\to\mathbb{R}^{n_x}}{\text{minimize}} \int_{t_0}^{t_f} \|\dot{\mathbf{x}}(t)\|_2^2 \, dt \tag{16}$$

$$\text{subject to} \quad \mathbf{x}(t_0) = \mathbf{x}_0, \ \mathbf{x}(t_f) = \mathbf{x}_f \tag{16a}$$

$$\text{Prob}\left(\mathbf{x}(t) \in \mathcal{X}_{obs_i}(\omega_i, t)\right) \leq \Delta, \ \forall t \in [t_0, t_f] \ \Big|_{i=n_{o_s}+1}^{n_{o_d}} \tag{16b}$$

To obtain the deterministic optimization, we replace probabilistic constraints (16b) with deterministic constraints in terms of the dynamic Δ-risk contours as follows:

$$\underset{\mathbf{x}(t):[t_0,t_f]\to\mathbb{R}^{n_x}}{\text{minimize}} \int_{t_0}^{t_f} \|\dot{\mathbf{x}}(t)\|_2^2 \, dt \tag{17}$$

$$\text{subject to} \quad \mathbf{x}(t_0) = \mathbf{x}_0, \ \mathbf{x}(t_f) = \mathbf{x}_f \tag{17a}$$

$$\mathbf{x}(t) \in \hat{\mathcal{C}}^{\Delta}_{r_i}(t), \ \forall t \in [t_0, t_f], \ i = n_{o_s}+1, ..., n_{o_d} \tag{17b}$$

where $\hat{\mathcal{C}}^{\Delta}_{r_i}(t)$ is the dynamic Δ-risk contour of the dynamic uncertain obstacle $\mathcal{X}_{obs_i}(\omega_i, t)$. The set of $\hat{\mathcal{C}}^{\Delta}_{r_i}(t), i = o_{n_s} + 1, ..., o_{n_d}$ represents the inner approximation of the feasible set of the probabilistic optimization in (16). In the deterministic polynomial optimization of (17), we need to make sure that the constraints are satisfied over the entire planning time horizon $[t_0, t_f]$. To solve the time-varying deterministic polynomial optimization in (17), we will use i) time-varying SOS optimization described in Section V-A1 and also ii) sampling-based motion planning algorithm similar to the one in Section V-A3 that uses the SOS-based continuous-time safety verification. In the sampling-based algorithm, given the dynamic nature of the environment, one needs to verify the safety of the trajectory between the given two samples only for the time interval that is needed to traverse between the points. In this paper, we will use the following naive RRT-SOS algorithm: RRT algorithm looks for piece-wise linear trajectories with s number of linear pieces defined over the time intervals $\Delta t_i = [t_{i-1}, t_i), i = 1, ..., s$ as in (8). To construct the tree, we first fix the number of linear pieces s and build the tree for each time interval incrementally. To expand the tree for the time interval Δt_i, we connect the given sample point to a vertex in the tree, built for the time interval Δt_{i-1}, if the linear trajectory between the two points satisfies SOS condition (15) for the time interval Δt_i. Moreover, to construct the tree for the time interval Δt_{s-1}, we verify the safety of the linear trajectory between the given sample and the goal point for the time interval Δt_s as well.

Illustrative Example 4: Consider the uncertain moving obstacle in illustrative example 2. We want to find a risk bounded trajectory between the points $\mathbf{x}(0) = [1, -2]$ and $\mathbf{x}(1) = [3, 2]$ by solving the probabilistic optimization in (16) with $\Delta = 0.1$. For this purpose, we solve the deterministic

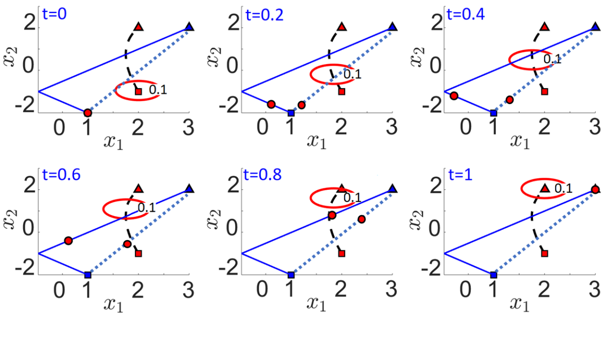

Fig. 5. Illustrative Example 4: Risk bounded trajectories between the start (square) and goal (triangle) points obtained via the time-varying SOS optimization (solid line) and RRT-SOS algorithm (dashed line) in the presence of the probabilistic moving obstacle.

optimization problem in (17) with respect to the dynamic 0.1-risk contour of the uncertain obstacle via the time-varying SOS optimization and RRT-SOS algorithm as shown in Figure 5. Note that although the obtained RRT-SOS trajectory looks like a straight line, it is a piece-wise linear trajectory consisting of 2 pieces with different velocities. The run-time for the time-varying SOS optimization and RRT-SOS algorithm are roughly 6.9 and 0.5 seconds, respectively.

VI. Experiments

In this section, numerical examples are presented to illustrate the performance of the proposed approaches. To obtain the risk bounded continuous-time trajectories, we use the time-varying SOS optimization and the RRT-SOS algorithm described in Section V[1]. Note that the provided RRT-SOS algorithm uses a naive tree search as explained in Section V. The main objective of the provided RRT-SOS algorithm is to demonstrate how the risk contours in (10) and (12) and the SOS-based continuous-time safety constraints in (15) can be incorporated into sampling-based motion planning algorithms to look for guaranteed risk bounded continuous-time trajectories in stochastic environments. The computations in this section were performed on a computer with Intel i7 2.7 GHz processors and 8 GB RAM. We use the Spotless MATLAB package [25] to verify the SOS-based continuous-time safety constraints in the RRT-SOS algorithm and the Julia package provided by [7] to solve the time-varying SOS optimization.

A. Risk Contours

The purpose of this example is to demonstrate how the provided approach can be used to compute the risk contours in the presence of highly complex uncertain unsafe regions.

1) 2D Uncertain Obstacle: Static uncertain obstacle of the form (4) is described by the polynomial $\mathcal{P}(\mathbf{x},\omega) = -0.42x_1^5 - 1.18x_1^4x_2 - 0.47x_1^4 + 0.3x_1^3x_2^2 - 0.57x_1^3x_2 + 0.6x_1^3 - 0.65x_1^2x_2^3 + 0.17x_1^2x_2^2 + 1.87x_1^2x_2 + 0.06x_1^2 + 0.69x_1x_2^4 - 0.14x_1x_2^3 - 0.85x_1x_2^2 + 0.6x_1x_2 - 0.21x_1 + 0.01x_2^5 - 0.06x_2^4 - 0.07x_2^3 - 0.41x_2^2 - 0.08x_2 + 0.07 - 0.1w$ where the uncertain parameter ω has a Beta distribution with parameters $(9, 0.5)$ over $[0, 1]$. Figure 6 shows the uncertain obstacle for different values of the uncertain parameter ω. We obtain the static risk contours defined in (10) for different risk levels Δ as shown in Figure 6 and 7.

2) 3D Uncertain Obstacle: Static uncertain obstacle of the form (4) is described by the polynomial $\mathcal{P}(\mathbf{x},\omega) = 0.94 - 0.002x_1 - 0.004x_2 - 0.04x_3 - 0.38x_1^2 + 0.04x_1x_2 - 0.31x_2^2 - 0.05x_1x_3 - 0.01x_2x_3 - 0.4x_3^2 - 0.1x_1^3 - 0.02x_1^2x_2 + 0.09x_1x_2^2 - 0.05x_2^3 + 0.14x_1^2x_3 - 1.83x_1x_2x_3 + 0.11x_2^2x_3 - 0.1x_1x_3^2 + 0.12x_2x_3^2 + 0.34x_3^3 - 0.32x_1^4 - 0.13x_1^3x_2 + 0.48x_1^2x_2^2 + 0.11x_1x_2^3 - 0.34x_2^4 + 0.03x_1^3x_3 + 0.01x_1^2x_2x_3 - 0.005x_1x_2^2x_3 - 0.05x_2^3x_3 + 0.54x_1^2x_3^2 - 0.06x_1x_2x_3^2 + 0.48x_2^2x_3^2 + 0.008x_1x_3^3 + 0.06x_2x_3^3 - 0.3x_3^4 + 0.12x_1^5 + 0.005x_1^4x_2 - 0.1x_1^3x_2^2 + 0.007x_1^2x_2^3 + 0.005x_1x_2^4 + 0.071x_2^5 - 0.02x_1^4x_3 + 0.73x_1^3x_2x_3 - 0.07x_1^2x_2^2x_3 + 0.72x_1x_2^3x_3 - 0.20x_2^4x_3 + 0.03x_1^3x_3^2 - 0.01x_1^2x_2x_3^2 + 0.02x_1x_2^2x_3^2 - 0.05x_2^3x_3^2 - 0.07x_1^2x_3^3 + 0.73x_1x_2x_3^3 + 0.09x_2^2x_3^3 + 0.03x_1x_3^4 - 0.06x_2x_3^4 - 0.31x_3^5 - \omega - 0.84$ where the uncertain parameter ω has a normal distribution with mean 0.1 and variance 0.001, [30]. Figure 8 shows the uncertain obstacle for different values of the uncertain parameter ω. We obtain the static risk contours defined in (10) for different risk levels Δ as shown in Figure 8.

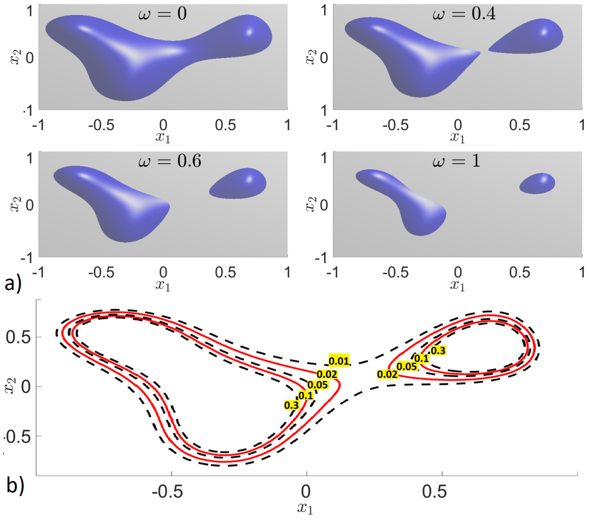

Fig. 6. Example A-1: a) Nonconvex obstacle with uncertain parameter $\omega \sim Beta(9, 0.5)$, b) Obtained static risk contours $\hat{\mathcal{C}}_r^{\Delta}$ defined in (10) for different risk levels $\Delta = [0.3, 0.1, 0.05, 0.02, 0.01]$.

B. Risk Bounded Lane Changing for Autonomous Vehicles

In this example, we generate a risk bounded trajectory for the lane-change maneuver of an autonomous vehicle in the presence of surrounding vehicles. In this scenario, uncertain locations of the surrounding vehicles are modeled as the following sets: $\mathcal{X}_{obs_1}(\omega_1, t) = \{(x_1, x_2) : 0.3^2 - (x_1 - p_1(t,\omega_1))^2 - (x_2 - 1)^2 \geq 0\}$ and $\mathcal{X}_{obs_2}(\omega_2, t) = \{(x_1, x_2) : 0.3^2 - (x_1 - p_2(t,\omega_1))^2 - x_2^2 \geq 0\}$ where $p_1(t,\omega_1) = t + 0.4 + \omega_1$ and $p_2(t,\omega_2) = 2t + 0.6 + \omega_2$ are the uncertain trajectories of the surrounding vehicles with uncertain parameters $\omega_i \sim Uniform[-0.1, 0.1], i = 1, 2$. For the lane-change maneuver, we look for a risk bounded trajectory between the points $\mathbf{x}(0) = (0, 0)$ and $\mathbf{x}(1) = (2, 0)$ over the planning time horizon $t \in [0, 1]$. Figure 9 shows the obtained trajectory using the time-varying SOS optimization and RRT-SOS algorithm considering the dynamic 0.1-risk contours of the surrounding

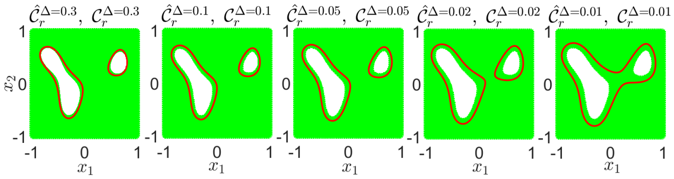

Fig. 7. Example A-1: True Δ-risk contour $\mathcal{C}_r^{\Delta}$ (green) and its inner approximation $\hat{\mathcal{C}}_r^{\Delta}$ (outside of the solid-line). Inside the risk contours, probability of collision with the uncertain obstacle is less or equal to the associated risk level Δ.

[1] github.com/jasour/Risk-Bounded-Continuous-Time-Trajectory-Planning

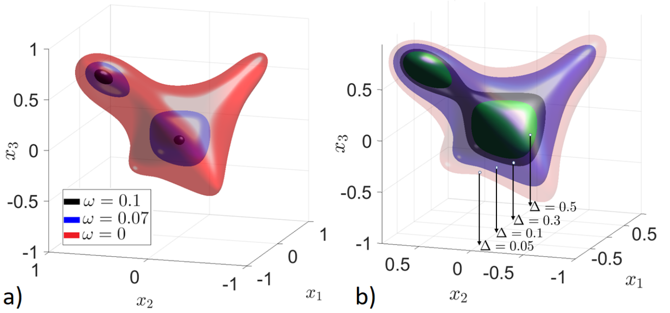

Fig. 8. Example A-2: a) Nonconvex obstacle with uncertain parameter $\omega \sim \mathcal{N}(0.1, 0.001)$, b) Obtained static risk contours $\hat{\mathcal{C}}_r^{\Delta}$ defined in (10) for different risk levels $\Delta = [0.5, 0.3, 0.1, 0.05]$.

vehicles. The run-time for the time-varying SOS optimization and RRT-SOS are roughly 1.2 and 6.5 seconds, respectively.

C. Risk Bounded Trajectory Planning for Delivery Robots

In this example, we generate a risk bounded trajectory for a delivery robot in the presence of uncertain moving obstacles. In this scenario, uncertain locations of the moving obstacles are modeled as the following sets: $\mathcal{X}_{obs_1}(\omega_1, t) = \{(x_1, x_2) : 0.4^2 - (-x_1 + p_1(t, \omega_1))^2 - (x_2 - 1)^2 \geq 0\}$, $\mathcal{X}_{obs_2}(\omega_2, t) = \{(x_1, x_2) : 0.4^2 - (x_1 + p_2(t, \omega_1))^2 - (x_2 - 2)^2 \geq 0\}$, and $\mathcal{X}_{obs_3}(\omega_3, t) = \{(x_1, x_2) : 0.4^2 - (-x_1 + p_3(t, \omega_3))^2 - (x_2 - 3)^2 \geq 0\}$ where $p_1(t, \omega_1) = t - 0.5 + \omega_1$, $p_2(t, \omega_1) = 2t - \omega_2 - 0.8$, and $p_3(t, \omega_3) = 1.8t - 0.7 + \omega_3$ are the uncertain trajectories of the moving obstacles with uncertain parameters $\omega_i \sim Uniform[-0.1, 0.1], i = 1, 2, 3$. We look for a risk bounded trajectory between the start and destination points, $\mathbf{x}(0) = (0, 0)$ and $\mathbf{x}(1) = (0, 4)$, over the planning time horizon $t \in [0, 1]$. Figure 10 shows the obtained trajectories using the time-varying SOS optimization and RRT-SOS algorithm considering the dynamic 0.1-risk contours of the moving obstacles. The run-time for the time-varying SOS optimization and RRT-SOS are roughly 7 and 197 seconds, respectively.

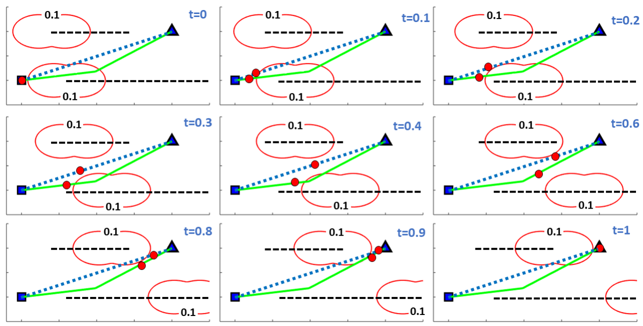

Fig. 9. Example B: Risk bounded trajectories for the lane-change maneuver between the start (square) and goal (triangle) points obtained via the time-varying SOS optimization (solid line) and RRT-SOS algorithm (dotted line) in the presence of surrounding vehicles with uncertain trajectories. Dashed lines show the expected values of the uncertain trajectories of the surrounding vehicles.

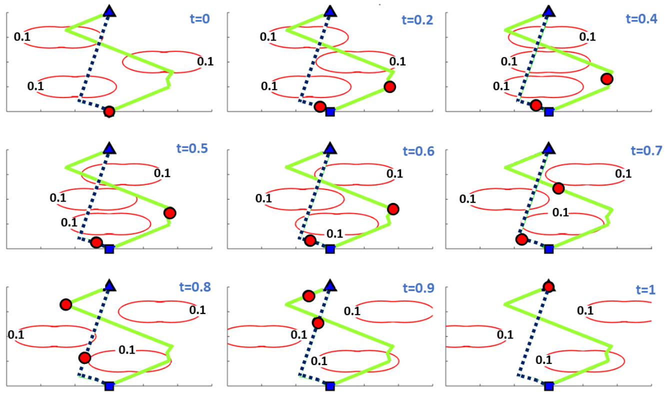

Fig. 10. Example C: Risk bounded trajectories for a delivery robot between the start (square) and goal (triangle) points obtained via the time-varying SOS optimization (solid line) and RRT-SOS algorithm (dotted line) in the presence of moving uncertain obstacles.

D. Planning in Cluttered Uncertain Environments

In this example, we look for risk bounded trajectories in cluttered static and dynamic uncertain environments.

1) 2D Static Environment: In this scenario, we use the RRT-SOS algorithm to obtain a risk bounded trajectory with $\Delta = 0.1$ between the start and goal points, $\mathbf{x}(0) = (0, 0)$, $\mathbf{x}(1) = (5, 5)$, in the presence of circle-shaped obstacles with uncertain position as shown in Figure 11. The position of the obstacles is subjected to an additive zero mean Gaussian noise with 0.001 variance. The run-time to obtain the feasible and optimal trajectories are roughly 27 and 92 seconds, respectively.

2) 3D Dynamic Environment: In this scenario, we use the RRT-SOS algorithm to obtain a risk bounded trajectory with $\Delta = 0.1$ between the start and goal points, $\mathbf{x}(0) = (-1, -1, 0)$, $\mathbf{x}(1) = (1, 1, 1)$, in the presence of moving sphere-shaped obstacles with uncertain radius and uncertain trajectories as shown in Figure 12. Radius of the obstacles has a uniform distribution over $[0.1, 0.2]$. Also, trajectories of the obstacles are subjected to an additive zero mean Gaussian noise with 0.001 variance. The run-time to obtain the risk bounded trajectory is roughly 29 seconds.

E. Discussion

The provided risk bounded algorithms not only are faster than Monte Carlo-based risk bounded RRT algorithms (illus-

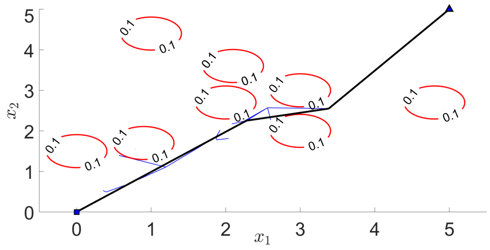

Fig. 11. Example D-1: Risk bounded trajectory between the start (square) and goal (triangle) points obtained via the RRT-SOS algorithm in the static uncertain cluttered environment.

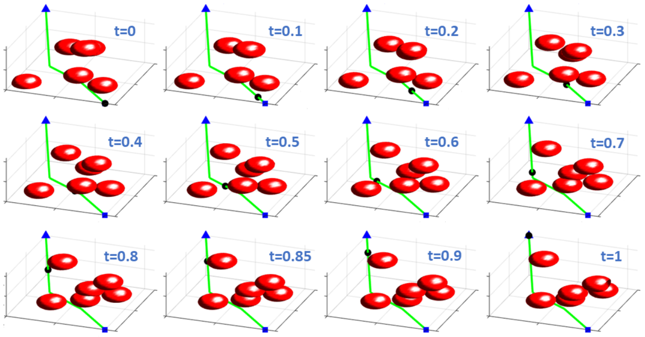

Fig. 12. Example D-2: Risk bounded trajectory between the start (square) and goal (triangle) points obtained via the RRT-SOS algorithm in the dynamic uncertain cluttered environment.

trative example 3), but also provide continuous-time trajectories with guaranteed bounded risk. In addition, compared with the RRT-SOS algorithm, the time-varying SOS optimization is generally faster. The run-time of the time-varying SOS optimization is a function of the order of polynomials of risk contours, the number of the linear pieces of risk bounded trajectories, and the number of iterations of the heuristic algorithm. We use the heuristic algorithm introduced by [7] that trades off theoretical guarantees for more efficiency to avoid large scale time-varying SOS optimization. Hence, the heuristic algorithm may fail to obtain a feasible trajectory as observed in Experiment D. To ensure safety, we verify the trajectory returned by the heuristic algorithm using the SOS condition in (15). In the time-varying SOS optimization, one can initially start with a small number of iterations and linear pieces and then increase the parameters if the obtained trajectory is infeasible. On the other hand, the RRT-SOS algorithm is more robust and always returns a feasible trajectory. The provided SOS-based RRT algorithm uses a naive tree search algorithm to look for the risk bounded trajectories. One can improve the performance and run-time by considering efficient sampling-based motion planning algorithms and high-order polynomial trajectories between the samples.

VII. Conclusion

In this paper, we provided continuous-time trajectory planning algorithms to obtain risk bounded polynomial trajectories in uncertain nonconvex environments that contain static and dynamic obstacles with probabilistic location, size, and geometry. The provided algorithms leverage the notion of risk contours to transform the probabilistic trajectory planning problem into a deterministic planning problem and use convex methods to obtain the continuous-time trajectories with guaranteed bounded risk without the need for time discretization and uncertainty samples. The provided algorithms are suitable for online and large scale planning problems. For the future work, we will incorporate the provided approaches into a model predictive control (MPC) framework to address the online planning problems. Also, we will extend the provided approaches to deal with inaccurate models of uncertain parameters, e.g., inaccurate probability distributions and moments.

VIII. Appendix: Proof of Theorem 1

To compute the static risk contour in (9), we obtain the deterministic constraint as follows: For a given point $\mathbf{x} \in \mathcal{X}$, the probability of collision with the uncertain static obstacle, i.e., Prob$(\mathbf{x} \in \mathcal{X}_{obs}(\omega))$, is equivalent to the expectation of the indicator function of the superlevel set of $\mathcal{P}(\mathbf{x},\omega)$ as follows [29], [30]:

$$\text{Prob}(\mathcal{P}(\mathbf{x},\omega) \geq 0) = \int_{\{(\mathbf{x},\omega):\mathcal{P}(\mathbf{x},\omega)\geq 0\}} pr(\omega)d\omega = \mathbb{E}[\mathbb{I}_{\mathcal{P}\geq 0}] \quad (18)$$

where $pr(\omega)$ is the probability density function of ω and $\mathbb{I}_{\mathcal{P}\geq 0}$ is the indicator function of the superlevel set of $\mathcal{P}(\mathbf{x},\omega)$ defined as $\mathbb{I}_{\mathcal{P}\geq 0} = 1$ if $(\mathbf{x},\omega) \in \{(\mathbf{x},\omega) : \mathcal{P}(\mathbf{x},\omega) \geq 0\}$, and 0 otherwise. The expectation of the indicator function, however, is not necessarily easily computable. To compute the expectation value, we will find a polynomial description of the indicator function that upper bounds the true indicator function $\mathbb{I}_{\mathcal{P}\geq 0}$. If we can find a polynomial $P_{\mathbb{I}} : \mathbb{R}^{n_x+n_\omega} \rightarrow \mathbb{R}$ of the order d with coefficients c_{ij} that upper bounds the indicator function, i.e., $P_{\mathbb{I}}(\mathbf{x},\omega) := \sum_{(i,j)} c_{ij}x^i\omega^j \geq \mathbb{I}_{\mathcal{P}\geq 0}$, then we can apply the expectation w.r.t. the probability density function of ω to the both sides and describe the upper bound probability in terms of the known moments of ω as follows:

$$\mathbb{E}[P_{\mathbb{I}}(\mathbf{x},\omega)] = \sum_{(i,j)} c_{ij}x^i\mathbb{E}[\omega^j] \geq \mathbb{E}[\mathbb{I}_{\mathcal{P}\geq 0}] = \text{Prob}(\mathcal{P}(\mathbf{x},\omega) \geq 0) \quad (19)$$

where $\mathbb{E}[\omega^j]$ is the moment of order j of random vector ω defined in Section II. Hence, we can construct an inner approximation of the Δ-risk contour in (9), denoted by $\hat{\mathcal{C}}_r^{\Delta}$, using the upper bound probability in (19) as follows:

$$\hat{\mathcal{C}}_r^{\Delta} = \{\ \mathbf{x} \in \mathcal{X} :\ \ P(\mathbf{x}) \leq \Delta\} \quad (20)$$

where polynomial $P(\mathbf{x}) = \mathbb{E}[P_{\mathbb{I}}(\mathbf{x},\omega)] = \sum_{(i,j)} c_{ij}\mathbb{E}[\omega^j]x^i$ as shown in (19).

According to (19) and (20), the problem of constructing Δ-risk contour (9), reduces to the problem of finding an upper bound probability and an upper bound polynomial indicator function of the superlevel set of $\mathcal{P}(\mathbf{x},\omega)$. In [29], to compute the upper bound polynomial indicator functions, we provide an $(n_x + n_\omega)$-dimensional convex optimization problem in the form of a semidefinite program. Such optimization is not suitable for online computations and is limited to small dimensions (n_x+n_ω). In this paper, we propose an optimization-free approach to obtain upper bound probability and polynomial indicator function that is suitable for online computations and large scale problems as follows:

Let $\mathcal{X}_{obs}(\omega) = \{\mathbf{x} \in \mathcal{X} : \mathcal{P}(\mathbf{x},\omega) \geq 0\}$ be the given static uncertain obstacle as defined in (4). To obtain an upper bound of the probability Prob$(\mathbf{x} \in \mathcal{X}_{obs}(\omega))$ = Prob$(\mathcal{P}(\mathbf{x},\omega) \geq 0)$, we begin by defining a new random variable $z \in \mathbb{R}$ as follows:

$$z = \mathcal{P}(\mathbf{x},\omega) \quad (21)$$

Note that $z \in \mathbb{R}$ is a random variable, while $\mathcal{P}(\mathbf{x},\omega) : \mathbb{R}^{n_x+n_\omega} \rightarrow \mathbb{R}$ is a polynomial in $\mathbf{x} \in \mathbb{R}^{n_x}$ and random vector $\omega \in \mathbb{R}^{n_\omega}$. By doing so, we can transform the $(n_x + n_\omega)$-dimensional probability assessment problem into a one-dimensional probability assessment problem in terms of the new defined random variable z, i.e., $\text{Prob}(\mathbf{x} \in \mathcal{X}_{obs}(\omega)) = \text{Prob}(\mathcal{P}(\mathbf{x}, \omega) \geq 0) = \text{Prob}(z \geq 0)$, [30]. Note that the statistics of the random variable z, e.g., moments, are functions of $\mathbf{x}$ and the statistics of the random vector ω.

Now, to compute an upper bound of the probability $\text{Prob}(z \geq 0)$, we just need an upper bound polynomial description of one-dimensional indicator function $\mathbb{I}_{z \geq 0}$ defined as $\mathbb{I}_{z \geq 0} = 1$ if $z \geq 0$, and 0 otherwise. In this paper, we will use the upper bound polynomial indicator function and the upper bound probability provided by *Cantelli's inequality* defined for scalar random variables as $\text{Prob}(z \geq 0) \leq \frac{\mathbb{E}[z^2] - \mathbb{E}[z]^2}{\mathbb{E}[z^2]}$ whenever $\mathbb{E}[z] \leq 0$. In other words, the probability of collision, i.e., $\text{Prob}(z \geq 0) = \text{Prob}(\mathcal{P}(\mathbf{x}, \omega) \geq 0)$, is bounded if the expected value of remaining safe is nonnegative, i.e, $\mathbb{E}[z] = \mathbb{E}[\mathcal{P}(\mathbf{x}, \omega)] \leq 0$. For other different one-dimensional indicator function-based probability bounds see [30], [33], [34].

Hence, the upper bound of the probability of collision can be described in terms of the polynomial of the uncertain obstacle as follows:

$$\text{Prob}(\mathbf{x} \in \mathcal{X}_{obs}(\omega)) \leq \frac{\mathbb{E}[\mathcal{P}^2(\mathbf{x}, \omega)] - \mathbb{E}[\mathcal{P}(\mathbf{x}, \omega)]^2}{\mathbb{E}[\mathcal{P}^2(\mathbf{x}, \omega)]} \tag{22}$$

whenever $\mathbb{E}[\mathcal{P}(\mathbf{x}, \omega)] \leq 0$. This will results in an inner approximation of the Δ-risk contour as in (10). Note that although the standard Cantelli's inequality uses the first two moments of the scalar random variable z, we need higher order moments of random vector ω to construct the set in (10). In fact, in this paper, we generalize the standard *scalar* Cantelli probability bound to obtain multivariate probability bound (22) involving nonconvex and nonlinear sets of obstacles.

References

[1] J. Reif and M. Sharir, "Motion planning in the presence of moving obstacles," *Journal of the ACM*, vol. 41, no. 4, pp. 764–790, 1994.

[2] K. M. Lynch and F. C. Park, *Modern Robotics*. Cambridge University Press, 2017.

[3] V. Kumar, M. Zefran, and J. Ostrowski, "Motion planning and control of robots," *New York: Wiley*, 1999.

[4] R. Charles, A. Bry, and N. Roy, "Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments," *Robotics Research, Springer*, pp. vol. 114, 649–666, 2016.

[5] S. M. LaValle, *Planning algorithms*. Cambridge university press, 2006.

[6] R. Deits and R. Tedrake, "Efficient mixed-integer planning for uavs in cluttered environments," *IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[7] B. E. Khadir, J. B. Lasserre, and V. Sindhwani, "Piecewise-linear motion planning amidst static, moving, or morphing obstacles," *IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[8] L. Blackmore and M. Ono, "Convex chance constrained predictive control without sampling," *AIAA Guidance, Navigation, and Control Conference*, 2009.

[9] L. Blackmore, M. Ono, A. Bektassov, and B. C. Williams, "A probabilistic particle-control approximation of chance-constrained stochastic predictive control," *IEEE Transactions on Robotics*, vol. 26, no. 3, pp. 502–517, 2010.

[10] W. Schwarting, J. Alonso-Mora, L. Pauli, S. Karaman, and D. Rus, "Parallel autonomy in automated vehicles: Safe motion generation with minimal intervention," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1928–1935, 2017.

[11] B. Luders, M. Kothari, and J. How, "Chance constrained rrt for probabilistic robustness to environmental uncertainty," *AIAA Guidance, Navigation, and Control Conference*, 2010.

[12] 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, 2018.

[13] C. Dawson, A. Jasour, A. Hofmann, and B. Williams, "Provably safe trajectory optimization in the presence of uncertain convex obstacles," *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[14] S. Dai, S. Schaffert, A. Jasour, A. Hofmann, and B. Williams, "Chance constrained motion planning for high-dimensional robots," *IEEE International Conference on Robotics and Automation (ICRA)*, 2019.

[15] M. Cannon, "Chance-constrained optimization with tight confidence bounds," *arXiv preprint arXiv:1711.03747*, 2017.

[16] G. C. Calafiore and M. C. Campi, "The scenario approach to robust control design," *IEEE Transactions on Automatic Control*, vol. 51, no. 5, pp. 742–753, 2006.

[17] L. Janson, E. Schmerling, and M. Pavone, "Monte carlo motion planning for robot trajectory optimization under uncertainty," *Robotics Research*, pp. 343–361, 2018.

[18] T. Summers, "Distributionally robust sampling-based motion planning under uncertainty," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6518–6523, 2018.

[19] A. Wang, A. Jasour, and B. C. Williams, "Non-gaussian chance-constrained trajectory planning for autonomous vehicles under agent uncertainty," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6041–6048, 2020.

[20] J. Jacod and P. Protter, *Probability essentials*. Springer Science & Business Media, 2012.

[21] P. A. Parrilo, "Semidefinite programming relaxations for semialgebraic problems," *Mathematical programming*, vol. 96, no. 2, 2003.

[22] J. B. Lasserre, "Global optimization with polynomials and the problem of moments," *SIAM Journal on optimization*, vol. 11, no. 3, pp. 796–817, 2001.

[23] M. Laurent, "Sums of squares, moment matrices and optimization over polynomials," *Emerging Applications of Algebraic Geometry, Springer*, pp. 157–270, 2009.

[24] J. Lofberg, "Yalmip: A toolbox for modeling and optimization in matlab," *IEEE International Conference on Robotics and Automation (ICRA)*, 2004.

[25] M. M. Tobenkin, F. Permenter, and A. Megretski, "spotless: Polynomial and conic optimization," 2013. [Online]. Available: github.com/spot-toolbox/spotless

[26] D. Henrion, J.-B. Lasserre, and J. Löfberg, "Gloptipoly 3: moments, optimization and semidefinite programming," *Optimization Methods & Software*, vol. 24, no. 4-5, pp. 761–779, 2009.

[27] A. A. Ahmadi and B. E. Khadir, "Time-varying semidefinite programs," *Mathematics of Operations Research, arXiv:1808.03994*, 2021.

[28] A. Bellon, D. Henrion, V. Kungurtsev, and J. Marecek, "Time-varying semidefinite programming: Geometry of the trajectory of solutions," *arXiv preprint arXiv:2104.05445*, 2021.

[29] A. Jasour and B. Williams, "Risk contours map for risk bounded motion planning under perception uncertainties," *Robotics: Science and Systems*, 2019.

[30] A. Jasour, A. Hofmann, and B. C. Williams, "Moment-sum-of-squares approach for fast risk estimation in uncertain environments," *IEEE Conference on Decision and Control (CDC)*, pp. 2445–2451, 2018.

[31] P. Roux, Y.-L. Voronin, and S. Sankaranarayanan, "Validating numerical semidefinite programming solvers for polynomial invariants," *Formal Methods in System Design*, vol. 53, no. 2, pp. 286–312, 2018.

[32] M. Putinar, "Positive polynomials on compact semi-algebraic sets," *Indiana University Mathematics Journal*, vol. 42, no. 3, pp. 969–984, 1993.

[33] A. Jasour, "Risk aware and robust nonlinear planning (rarnop)," *Course Notes for MIT 16.S498*, 2019. [Online]. Available: rarnop.mit.edu

[34] A. Nemirovski and A. Shapiro, "Convex approximations of chance constrained programs," *SIAM Journal on Optimization*, vol. 17, no. 4, pp. 969–996, 2007.