Risk-Aware Motion Planning and Control Using CVaR-Constrained Optimization

Astghik Hakobyan, Gyeong Chan Kim, and Insoon Yang, *Member, IEEE*

***Abstract*—We propose a risk-aware motion planning and decision-making method that systematically adjusts the safety and conservativeness in an environment with randomly moving obstacles. The key component of this method is the conditional value-at-risk (CVaR) used to measure the safety risk that a robot faces. Unlike chance constraints, CVaR constraints are coherent, convex, and distinguish between tail events. We propose a two-stage method for safe motion planning and control: a reference trajectory is generated by using RRT* in the first stage, and then a receding horizon controller is employed to limit the safety risk by using CVaR constraints in the second stage. However, the second stage problem is nontrivial to solve as it is a triple-level stochastic program. We develop a computationally tractable approach through (i) a reformulation of the CVaR constraints, (ii) a sample average approximation, and (iii) a *linearly constrained* mixed integer convex program formulation. The performance and utility of this risk-aware method are demonstrated through simulation using a 12-dimensional model of quadrotors.**

***Index Terms*—Optimization and optimal control, Probability and statistical methods, Robot safety, Collision avoidance, Motion and path planning.**

I. Introduction

THE safe operation of robots and autonomous systems in uncertain and dynamic environments, particularly those where humans are involved, has been regarded as a critical challenge. Moving objects and agents in robots' paths pose a major safety issue in practical environments. Unfortunately it is difficult to accurately predict an object's movement in many situations due to uncertainty in or incomplete knowledge about the object motion. The focus of this work is to develop an algorithmic tool for safe motion planning and decision-making in uncertain conditions by integrating prior knowledge about preferred motions of obstacles in a *risk-aware* manner.

Safe motion planning and control approaches under uncertainty have been extensively studied (see, for example, [1] and the references therein) and can be categorized as deterministic or stochastic. Deterministic methods often assume a bounded set of uncertainties and seek decision-making strategies that are robust with respect to this set of uncertainties. To obtain such robust solutions, algorithms using model predictive control [2], reachability [3], and safety funnels [4] have been developed, among others. However, robust methods often lead to unnecessarily conservative solutions, particularly when the set of uncertainties is overestimated. This conservativeness may be alleviated by using adaptive online planning [5]. Stochastic methods can be used to systematically adjust the safety and conservativeness by incorporating probabilistic information about environments into decision making. Chance constraint-based methods have been one of the most popular stochastic tools in motion planning and control as chance constraints have the intuitive and practical role of limiting the probability of unsafe events. In particular, chance constraints have been widely used in conjunction with model predictive control [6], [7], optimal control [8], and sampling-based planning [9], [10]. Due to the nonconvexity of chance-constraints, however, these methods often use an approximation technique or a limited class of probability distributions and system dynamics to obtain a computationally tractable solution.[1]

Departing from chance constraint-based methods, we propose a safe motion planning and control approach using *conditional value-at-risk* (CVaR) constraints. The CVaR of a random loss is equal to the *conditional expectation* of the loss within the $(1-\alpha)$ worst-case quantile of the loss distribution [15], [16]. On the other hand, the value-at-risk (VaR) of a loss represents this quantile of the loss distribution, and thus it is closely related to chance constraints. The CVaR and VaR of a random variable X, the distribution of which has no probability atoms, have the following relationship: $\mathrm{CVaR}_\alpha(X) = \mathbb{E}[X \mid X \geq \mathrm{VaR}_\alpha(X)]$. CVaR constraints have several advantages over VaR or chance constraints, including the following: First, CVaR is a *coherent* risk measure, unlike VaR. According to Artzner *et al.* [17], a risk measure is said to be *coherent* if it satisfies the four axioms of translation invariance, subadditivity, positive homogeneity, and monotonicity. Majumdar and Pavone [18] claim that these four axioms should be satisfied for rational assessments of risk in robotics applications. Second, CVaR constraints can distinguish tail events where losses exceed VaR, while chance constraints cannot [19]. Third, CVaR constraints are convex unlike most

Manuscript received: February 24, 2019; Revised May 24, 2019; Accepted July 1, 2019.

This paper was recommended for publication by Editor Dezhen Song upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by in part by NSF under ECCS-1708906, Research Resettlement Fund for the new faculty of SNU, the Creative-Pioneering Researchers Program through SNU, and the Basic Research Lab Program through the National Research Foundation of Korea funded by the MSIT(2018R1A4A1059976). (*Corresponding author: Insoon Yang.*)

A. Hakobyan, G. C. Kim, and I. Yang are with the Department of Electrical and Computer Engineering, Automation and Systems Research Institute, Seoul National University, Seoul 08826, Korea, {astghikhakobyan, skykim0609, insoonyang}@snu.ac.kr

Digital Object Identifier (DOI): see top of this page.

[1]There have been a few attempts to resolve the nonconvexity issue in using chance constraints for arbitrary probability distributions. In [11], [12], nonconvexity is handled by looking at higher order moments, while [13] considers moment-based ambiguity sets of distributions with given mean and covariance. [14] uses a sampling-based method for approximating arbitrary distribution by finite number of particles.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LRA.2019.2929980, IEEE Robotics and Automation Letters

of the chance constraints [15].

To implement CVaR in a robot's risk assessment for safe motion planning, we propose a safety risk measure by extending our previous work [20] to the case of model predictive control (MPC) with randomly moving obstacles.[2] Specifically, the proposed safety risk measure represents the conditional expectation of the deviation within the $(1-\alpha)$ worst-case quantile of an associated safety loss distribution. With this safety risk measure, we develop a two-stage method for safe motion planning and control. In the first stage, a reference trajectory is generated by a fast sampling algorithm, such as RRT*, given the initial configuration of obstacles. However, as obstacles start to move randomly, this reference trajectory may no longer be safe to follow. Thus, in the second stage, a receding horizon controller is used to limit the risk of unsafety. This MPC problem is a CVaR-constrained stochastic program. Despite the convexity of CVaR constraints, this problem is nontrivial to solve because (i) each CVaR constraint involves bilevel optimization problems (one for CVaR and another for set distance), and (ii) the safe region is nonconvex in many practical cases. To overcome the first challenge, we reformulate the CVaR constraints without sacrificing optimality as more tractable expectation constraints. We then employ the sample average approximation (SAA) of the expectation constraints to further remove the minimization problem for computing a set distance. We show that the optimal value and solution obtained by the proposed SAA converge their original counterpart. The second issue, caused by nonconvex safe regions, can be addressed by recasting the MPC problem as a *linearly constrained* mixed integer convex program.

The remainder of this paper is organized as follows: In Section II, the problem setup is presented including stochastic obstacle movements. In Section III, we describe the proposed motion planning and control method using CVaR-constrained optimization. In Section IV, we demonstrate the performance and utility of our method through simulations using a 12D quadrotor model in a 3D environment.

II. The Setup

A. System Model

Consider a robotic vehicle, such as a quadrotor, with dynamics that can be modeled as a discrete-time linear system of the form[3]

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t) + Du(t), \end{aligned}$$

where $x(t) \in \mathbb{R}^{n_x}$ and $y(t) \in \mathbb{R}^{n_y}$ are the system state and system output, respectively, and $u(t) \in \mathbb{R}^{n_u}$ is the control input at stage t. We assume that the output vector corresponds to the robotic vehicle's position in the n_y-dimensional configuration space. It is typical that a robotic system is subject to constraints on the system state and the control input:

$$x(t) \in \mathcal{X}, \quad u(t) \in \mathcal{U} \quad \forall t \geq 0. \tag{1}$$

We assume that $\mathcal{X} \subseteq \mathbb{R}^{n_x}$ and $\mathcal{U} \subseteq \mathbb{R}^{n_u}$ are convex sets.

[2]As opposed to the dynamic programming approach in [20], we propose an MPC-based method and a different set of new reformulation procedures to obtain a *linearly constrained* mixed integer convex program, even in the presence of randomly moving obstacles.

[3]The reformulation results in Section III-C are also valid for nonlinear systems. However, we focus on linear systems because the computational costs of nonlinear MPC are often prohibitive.

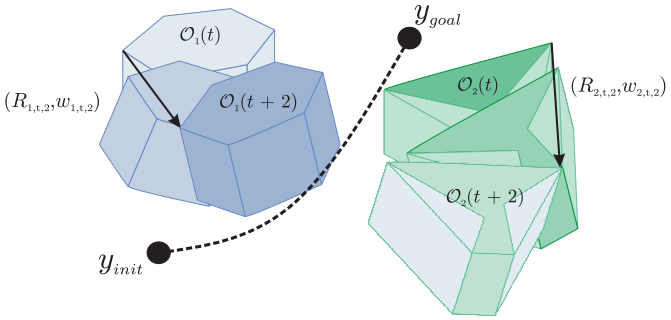


Fig. 1: Robot configuration space with moving obstacles: $\mathcal{O}_1(t)$ is a convex obstacle, while $\mathcal{O}_2(t)$ is a nonconvex obstacle, approximated as its convex hull.

B. Randomly Moving Obstacles

The robot must navigate the space while avoiding L *randomly moving* rigid body obstacles. The region occupied by obstacle ℓ at stage t is represented by subset $\mathcal{O}_\ell(t)$ of $\mathbb{R}^{n_y}$. If an obstacle is not a convex polytope, we over-approximate it as a polytope and form its convex hull as shown in Fig. 1 [21].[4] After such polyhedral approximation, we model the initial location of each obstacle as a convex polytope, i.e.,

$$\mathcal{O}_\ell(0) = \{\mathbf{y} \mid c_{i,\ell}^\top \mathbf{y} \leq d_{i,\ell},\ i = 1, \ldots, m_\ell\}, \tag{2}$$

where m_ℓ is the number of half-spaces defining the obstacle.

To formalize obstacle avoidance problems, we define the *safe region* as the complement to the region occupied by the obstacles. Specifically, let

$$\mathcal{Y}_\ell(t) := \mathbb{R}^{n_y} \setminus \mathcal{O}_\ell^o(t) \quad \forall t \geq 0,$$

where $\mathcal{O}_\ell^o(t)$ denotes the interior of $\mathcal{O}_\ell$. Note that $\mathcal{Y}_\ell(t)$ is a closed set. For safety, the output is subject to the following constraint:

$$y(t) \in \mathcal{Y}_\ell(t) \quad \forall t \geq 0 \ \ \forall \ell.$$

The safe region also changes over time because the obstacle is moving randomly. The obstacle's movement between two stages is assumed to be represented by a linear transform, which includes the composition of rotations and translations, i.e.,

$$\mathcal{O}_\ell(t+k) = R_{\ell,t,k}\mathcal{O}_\ell(t) + w_{\ell,t,k},$$

where the nonsingular matrix $R_{\ell,t,k}$ is the product of random rotation matrices and $w_{\ell,t,k}$ is a random translation vector. Here, adding a vector w to a set $\mathcal{A}$ is defined by $\mathcal{A} + w := \{a + w \mid a \in \mathcal{A}\}$, multiplying a matrix R to a set $\mathcal{A}$ is defined by $R\mathcal{A} := \{R \cdot (a - c_\mathcal{A}) + c_\mathcal{A} \mid a \in \mathcal{A}\}$, where $c_\mathcal{A}$ is the

[4]This over-approximation is reasonable to plan safe paths.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LRA.2019.2929980, IEEE Robotics and Automation Letters

centroid of $\mathcal{A}$. Accordingly, $\mathcal{Y}_\ell(t)$ is shifted to $\mathcal{Y}_\ell(t+k) = \mathbb{R}^{n_y} \setminus \mathcal{O}_\ell^o(t+k)$. It is straightforward to check

$$\mathcal{Y}_\ell(t+k) = R_{\ell,t,k}\mathcal{Y}_\ell(t) + w_{\ell,t,k}.$$

We assume that the region $\mathcal{O}_\ell(0)$ occupied by obstacle ℓ is known in advance for each ℓ.

III. Risk-Aware Motion Planning and Control Using CVaR Constraints

The proposed risk-aware motion planning and control method consists of two stages:

1) fast reference trajectory planning, and
2) risk-constrained MPC.

In the first stage, a reference trajectory, which avoids $\mathcal{O}_\ell(0)$ $\forall \ell$, is generated by using path-planning tools, such as RRT*. However, even if the robotic vehicle follows this reference trajectory, its safety is not guaranteed, due to the interference of randomly moving obstacles. To systematically limit the risk of collision, a risk-aware model predictive controller is employed in the second stage. The risk awareness is incorporated into the receding horizon optimization or MPC problem by using CVaR constraints. We develop a computationally tractable solution approach to the MPC problem through three reformulation procedures.

A. Reference Trajectory Planning

The first step is computing a collision-free path given the initial configuration of obstacles. In this work, RRT* is used for reference trajectory generation [22]. It efficiently searches nonconvex, high-dimensional spaces by randomly building a space-filling tree. The tree is constructed incrementally in a way that quickly reduces the expected distance between a randomly chosen point and the tree. Compared to RRT [23], the main advantage of RRT* is that it provides an asymptotically optimal solution. Furthermore, in RRT* two major features are introduced: tree rewiring and near neighbor search, thus improving the path quality. Note, however, that any fast algorithms that generate a safe reference trajectory can be used in the first stage as a more sophisticated decision-making is performed in the second stage to limit the safety risk that a robot faces during its operation.

B. Safety Risk

As previously mentioned, the reference trajectory may no longer be safe once the obstacles start to move. To account for the randomness of obstacles' movement and develop a risk-aware decision-making tool, we mathematically define a notion of *safety risk* by combining set distance and CVaR.

Regarding the obstacle $\mathcal{O}_\ell^o(t)$, we measure the *loss of safety* at stage t as the distance between the robot's position $y(t)$ and the safe region $\mathcal{Y}_\ell(t) = \mathbb{R}^{n_y} \setminus \mathcal{O}_\ell^o(t)$:

$$\operatorname{dist}(y(t), \mathcal{Y}_\ell(t)) := \min_{a \in \mathcal{Y}_\ell(t)} \|y(t) - a\|. \tag{3}$$

Obviously, it is desirable to drive the robot so that $\operatorname{dist}(y(t), \mathcal{Y}_\ell(t)) = 0$ for safety. However, due to the randomness of an obstacle's motion, imposing a hard constraint may lead to a very conservative decision. Instead of making such a deterministic guarantee, we systematically take into account the risk of unsafety to make the corresponding risk-aware decision depending upon the degree of risk that the robot can take. Specifically, we evaluate the risk of system unsafety using the CVaR defined by

$$\operatorname{CVaR}_\alpha(X) := \min_{z \in \mathbb{R}} \mathbb{E}\left[z + \frac{(X-z)^+}{1-\alpha} \right], \tag{4}$$

where $(\boldsymbol{x})^+ = \boldsymbol{x}$ if $\boldsymbol{x} \geq 0$ and $(\boldsymbol{x})^+ = 0$ otherwise [16].

We quantify the *safety risk* of the robot at stage t as $\operatorname{CVaR}_\alpha[\operatorname{dist}(y(t), \mathcal{Y}_\ell(t))]$ concerned with random obstacle ℓ. This safety risk measures the conditional expectation of the distance between the robot position $y(t)$ and the safe region $\mathcal{Y}_\ell(t) = \mathbb{R}^{n_y} \setminus \mathcal{O}_\ell^o(t)$ within the $(1-\alpha)$ worst-case quantile of the safety loss distribution. In the following subsection, we formulate an MPC problem with constraints on the safety risk regarding all the obstacles:

$$\operatorname{CVaR}_\alpha[\operatorname{dist}(y(t), \mathcal{Y}_\ell(t))] \leq \delta_\ell \quad \forall \ell,$$

where $\delta_\ell \geq 0$ is a user-specified risk tolerance parameter for obstacle ℓ. In practice, δ_ℓ can be chosen as the maximum allowable expected deviation of a robot's position from the safe region $\mathcal{Y}_\ell$. On the other hand, another parameter α must be chosen by assessing the risk aversion of the user or the robot. For example, our Monte Carlo simulation results in Section IV indicate that $\alpha = 0.95$ induces extremely risk-averse decisions that cause all sample trajectories to be safe, among 10,000 samples. Another way to determine δ_ℓ and α is to use the *mean-CVaR efficient frontier*, which represents the possible tradeoff between minimizing the expected cost of motion planning and minimizing the safety risk (e.g., [24]).

C. Risk-Constrained Model Predictive Control

The main part of our risk-aware method is the following CVaR-constrained MPC problem (for stage t):[5]

$$\min_{\mathbf{u},\mathbf{x},\mathbf{y}} \quad J(x(t), \mathbf{u}) := \sum_{k=0}^{K-1} r(x_k, u_k) \tag{5a}$$

$$\text{s.t.} \quad x_{k+1} = Ax_k + Bu_k \tag{5b}$$

$$y_k = Cx_k + Du_k \tag{5c}$$

$$x_0 = x(t) \tag{5d}$$

$$x_k \in \mathcal{X},\ u_k \in \mathcal{U} \tag{5e}$$

$$\operatorname{CVaR}_\alpha[\operatorname{dist}(y_k, \mathcal{Y}_\ell(t+k))] \leq \delta_\ell \ \ \forall \ell, \tag{5f}$$

where $\mathbf{u} := (u_0, \ldots, u_{K-1})$, $\mathbf{x} := (x_0, \ldots, x_K)$, $\mathbf{y} := (y_0, \ldots, y_{K-1})$ and all the constraints must hold for $k = 0, \ldots, K-1$ except for $x_k \in \mathcal{X}$ in (5e) which must hold for $k = 0, \ldots, K$. After computing an optimal $\mathbf{u}^\star$, only the first component $u_0^\star$ is chosen to be the control input at stage t, i.e., $u(t) := u_0^\star$. For the next stage, the MPC problem is defined in a receding horizon fashion and solved to obtain

[5]Our problem formulation and solution method is different from the one investigated by Singh *et al.* [25] as they consider uncertainties in model parameters (matrices A and B), whereas we consider uncertainties in obstacles' movement.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LRA.2019.2929980, IEEE Robotics and Automation Letters

$u(t+1)$ given $x(t+1)$. Here, the constraints (5b) and (5c) account for the system state and output predicted in the MPC horizon by initializing x_0 as the current state $x(t)$. The state and input constraints (1) are specified in (5e). The stage-wise cost function $r : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}$ is chosen to evaluate the control performance. For example, given a reference trajectory, $\nu(t)$, obtained by RRT*, the stage-wise cost function may be selected as follows:

$$r(x_k, u_k) := \|y_k - \nu(t+k)\|_Q^2 + \|u_k\|_R^2 \tag{6}$$

where $Q \succeq 0$ and $R \succ 0$ are the penalty weighing matrices, which penalize the running trajectory deviations and large input values, respectively.

Even when the cost function r is convex, it is nontrivial to solve the optimization problem (5) due to the CVaR constraint (5f). This constraint involves bilevel minimization, where the inner problem involves computing the distance $\mathrm{dist}(y(t), \mathcal{Y}_\ell(t))$ using (3) and the outer problem involves evaluating CVaR via (4). To develop a computationally tractable method, we first reformulate the MPC problem as the following stochastic program without loss of optimality:

Theorem 1. *Suppose that the stage-wise cost function r is continuous and the obstacles are convex polytopes as in* (2)*. Then, the set of optimal* $\mathbf{u}$*'s of the CVaR-constrained MPC problem* (5) *is equal to that of the following problem:*

$$\min_{\mathbf{u},\mathbf{x},\mathbf{y},\mathbf{z},\mathbf{h}} \; J(x(t), \mathbf{u}) := \sum_{k=0}^{K-1} r(x_k, u_k) \tag{7a}$$

$$\text{s.t. } \mathbb{E}\left[z_{\ell,k} + \frac{(h_{\ell,k} - z_{\ell,k})^+}{1-\alpha}\right] \le \delta_\ell \tag{7b}$$

$$y_k + \frac{c_{\ell,t,j}}{\|c_{\ell,t,j}\|} h_{\ell,k} \in \mathcal{Y}_\ell(t+k) \tag{7c}$$

$$h_{\ell,k} \ge 0 \tag{7d}$$

$$z_{\ell,k} \in \mathbb{R} \tag{7e}$$

$$\text{(5b)–(5e)}, \tag{7f}$$

where the constraint (7c) *must hold for at least one $j \in \{1, \dots, m_\ell\}$. All the remaining constraints must hold for all $\ell = 1, \dots, L$, and $k = 0, \dots, K-1$ except for $x_k \in \mathcal{X}$ in* (7f) *which must hold for $k = 0, \dots, K$.*

Its proof is contained in Appendix A. To numerically solve the reformulated MPC problem (7), we need to compute the expected value in the constraint (7b). The constraint (7c) is equivalent to $y_k + \frac{c_{\ell,t,j}}{\|c_{\ell,t,j}\|} h_{\ell,k} \in R_{\ell,t,k}\mathcal{Y}_\ell(t) + w_{\ell,t,k}$ because $\mathcal{Y}_\ell(t+k) = R_{\ell,t,k}\mathcal{Y}_\ell(t) + w_{\ell,t,k}$. One can rewrite the expectation as an integral with respect to a probability measure, then discretize a probability density to compute the integral. However, this approach involves a multi-dimensional integral, which is computationally demanding. In stochastic programming, a typical way to alleviate this issue is to employ *sample average approximation* (SAA). This approach approximates an expected constraint function or objective function using a sample average estimate, where the sample data are generated according to the underlying distribution. Specifically, given the sample $\{(\hat{R}_{\ell,t,k}^{(1)}, \hat{w}_{\ell,t,k}^{(1)}), \dots, (\hat{R}_{\ell,t,k}^{(N_k)}, \hat{w}_{\ell,t,k}^{(N_k)})\}$ of $(R_{\ell,t,k}, w_{\ell,t,k})$, we approximate the constraint (7b) as

$$\frac{1}{N_k}\sum_{i=1}^{N_k}\left[z_{\ell,k} + \frac{(h_{\ell,k}^{(i)} - z_{\ell,k})^+}{1-\alpha}\right] \le \delta_\ell. \tag{8}$$

Based on the SAA, we propose the following SAA-MPC problem:

$$\min_{\mathbf{u},\mathbf{x},\mathbf{y},\mathbf{z},\mathbf{h},\eta} \; J(x(t), \mathbf{u}) := \sum_{k=0}^{K-1} r(x_k, u_k) \tag{9a}$$

$$\text{s.t. } \frac{1}{N_k}\sum_{i=1}^{N_k}\left[z_{\ell,k} + \frac{\eta_{\ell,k}^{(i)}}{1-\alpha}\right] \le \delta_\ell \tag{9b}$$

$$y_k + \frac{c_{\ell,t,j}}{\|c_{\ell,t,j}\|} h_{\ell,k}^{(i)} \in \hat{R}_{\ell,t,k}^{(i)}\mathcal{Y}_\ell(t) + \hat{w}_{\ell,t,k}^{(i)} \; \forall i \tag{9c}$$

$$\eta_{\ell,k}^{(i)} \ge h_{\ell,k}^{(i)} - z_{\ell,k},\ \eta_{\ell,k}^{(i)} \ge 0,\ h_{\ell,k}^{(i)} \ge 0 \; \forall i \tag{9d}$$

$$z_{\ell,k} \in \mathbb{R} \tag{9e}$$

$$\text{(5b)–(5e)}, \tag{9f}$$

where the constraint (9c) must hold for at least one $j \in \{1, \dots, m_\ell\}$. All the constraints must hold for all $\ell = 1, \dots, L$ and $k = 0, \dots, K-1$ except for $x_k \in \mathcal{X}$ in (9f) which must hold for $k = 0, \dots, K$. Here an auxiliary real variable η is introduced to tackle the nonlinearity of (8).

To establish convergence properties of the proposed approximation method, we assume the following:

Assumption 1. *Suppose that*

1) *the stage-wise cost function r is continuous;*
2) *the set $\mathcal{U}$ is compact; and*
3) *for any optimal $\mathbf{u}^\star$ of the original problem, there exists a sequence $\mathbf{u}_N \in U_N$ such that $\mathbf{u}_N \to \mathbf{u}^\star$ with probability 1 as $N \to \infty$, where U_N is the set of feasible $\mathbf{u}$'s in* (9)*.*

We then have the following convergence results regarding the optimal value and optimal solutions of the approximate problem (9):

Theorem 2. *Let $J^\star$ and $\mathbb{U}^\star$ be the optimal value and the set of optimal* $\mathbf{u}$*'s of the original problem* (5)*. Similarly, let J_N and $\mathbb{U}_N$ be the optimal value and the set of optimal* $\mathbf{u}$*'s of the approximate problem* (9)*, where $N := \min_{k\in\{0,\dots,K-1\}} N_k$. Then, under Assumption 1, we have*

$$J_N \to J^\star \quad \textit{and} \quad D(\mathbb{U}_N, \mathbb{U}^\star) \to 0 \quad \textit{as } N \to \infty,$$

where $D(A, B) := \sup_{\boldsymbol{x}\in A} \mathrm{dist}(\boldsymbol{x}, B)$ denotes the deviation of the set A from the set B.

A proof of this theorem can be found in Appendix B. Note that the proposed SAA method provides a provable *asymptotic guarantee* of satisfying the original CVaR constraints. Although our numerical simulation results in Section IV indicate that the SAA method always satisfies the risk constraints *for all time steps* even with a relatively small sample size ($N_k = 20$), the proposed method has no provable *finite-sample* guarantee of the risk constraints. If such a finite-sample guarantee is required, then the following advanced techniques can be used in conjunction with the proposed method. First, a robust version of SAA, which combines distributionally robust

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LRA.2019.2929980, IEEE Robotics and Automation Letters

optimization and hypothesis testing of goodness-of-fit, can be used to obtain finite-sample guarantees in addition to retaining SAA's tractability and asymptotic properties [26]. Second, out-of-sample performance guarantees can be achieved in a probabilistic manner with a finite sample size by using a data-driven distributionally robust optimization framework [27], [28]. It is worth mentioning that the proposed reformulation procedures can also be used to enhance the computational tractability of the aforementioned techniques applied to motion planning. Specifically, an important feature of the approximate problem (9) is that it does not involve multiple-level optimization, unlike the original MPC problem (5). We can further reformulate (9) as a *linearly constrained* mixed integer convex program, when the cost function is convex, as proposed in the following subsection. A similar benefit may be obtained by using the proposed reformulation methods in conjunction with the advanced techniques for finite-sample guarantees.

D. Linearly Constrained Mixed Integer Convex Program

Due to the nonconvexity of the safe region $\mathcal{Y}_\ell(t)$, it is nontrivial to find an optimal solution to the problem (9). We now use the polyhedral characterization (2) of obstacles to reformulate the nonconvex constraints (9c) and recast the MPC problem as a *linearly constrained* mixed integer convex program (MICP). The proposed approach allows us to solve the MPC problem while retaining the full nonconvex constraints by using commercial MICP solvers. By continuous relaxation, we can also obtain *a posteriori* bounds on the gap between the optimal and approximate solution.

Due to the polyhedral characterization (2) of $\mathcal{O}_\ell(t)$, the constraint (9c) can be written as the following disjunctive representation:

$$\bigvee_{j=1}^{m_\ell} c_{\ell,t,j}^\top \left[(\hat{R}_{\ell,t,k}^{(i)})^{-1} (y_k + \frac{c_{\ell,t,j}}{\|c_{\ell,t,j}\|} h_{\ell,k}^{(i)}) - \hat{w}_{\ell,t,k}^{(i)} \right] \geq d_{\ell,t,j},$$

where m_ℓ is the number of half-spaces defining the obstacle, and $\bigvee$ denotes the logical disjunction operation. In order to tackle the 'OR' operation, the *Big-M reformulation* using binary variables is introduced as follows [29]:

$$\begin{aligned} & c_{\ell,t,j}^\top \left[(\hat{R}_{\ell,t,k}^{(i)})^{-1} (y_k + \frac{c_{\ell,t,j}}{\|c_{\ell,t,j}\|} h_{\ell,k}^{(i)}) - \hat{w}_{\ell,t,k}^{(i)} \right] \\ & \geq d_{\ell,t,j} - M_{j,\ell} \zeta_{j,\ell} \quad \forall j = 1, \ldots, m_\ell, && \text{(10a)} \\ & \zeta_{j,\ell} \in \{0,1\} \quad \forall j = 1, \ldots, m_\ell, && \text{(10b)} \\ & \sum_{j=1}^{m_\ell} \zeta_{j,\ell} \leq m_\ell - 1, && \text{(10c)} \end{aligned}$$

where $\ell = 1, \ldots, L$ and $k = 0, \ldots, K-1$. Here, $M_{i,\ell}$ is a constant greater than any possible value that the left-hand side of the inequality (10a) can have. By this inequality, when $\zeta_{i,\ell} = 0$, we have $c_{\ell,t,j}^\top [(\hat{R}_{\ell,t,k}^{(i)})^{-1}(y_k + \frac{c_{\ell,t,j}}{\|c_{\ell,t,j}\|} h_{\ell,k}^{(i)}) - \hat{w}_{\ell,t,k}^{(i)}] \geq d_{\ell,t,j}$. The inequality (10c) ensures that at least one of the binary variables is zero, and therefore (9c) holds at least for one $j \in 1, ..., m_\ell$.

By using this Big-M reformulation, the optimization problem (9) can be reformulated as follows:

$$\begin{aligned} \min_{\mathbf{u},\mathbf{x},\mathbf{y},\mathbf{z},\eta,\zeta} \; & J(x(t), \mathbf{u}) := \sum_{k=0}^{K-1} r(x_k, u_k) && \text{(11a)} \\ \text{s.t.} \; & \frac{1}{N_k} \sum_{i=1}^{N_k} \left[z_{\ell,k} + \frac{\eta_{\ell,k}^{(i)}}{1-\alpha} \right] \leq \delta_\ell \; \forall i && \text{(11b)} \\ & \eta_{\ell,k}^{(i)} \geq h_{\ell,k}^{(i)} - z_{\ell,k}, \; \eta_{\ell,k}^{(i)} \geq 0, \; h_{\ell,k}^{(i)} \geq 0 \; \forall i && \text{(11c)} \\ & z_{\ell,k} \in \mathbb{R}, && \text{(11d)} \\ & \text{(5b)–(5e), (10a)–(10c)} && \text{(11e)} \end{aligned}$$

where all the constraints must hold for all $\ell = 1, \ldots, L$ and $k = 0, \ldots, K-1$ except for $x_k \in \mathcal{X}$ in (11e) which must hold for $k = 0, \ldots, K$. This problem is a *linearly constrained* MICP when the cost function is convex.

Proposition 1. *Suppose that* $r : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}$ *is convex. Then, the reformulated MPC problem* (11) *is a linearly constrained mixed integer convex program.*

The linearly constrained MICP problem can be solved by using several methods, such as branch-and-bound [30], outer approximation [31] and polyhedral approximation [32]. State-of-the-art solvers like Bonmin [33], SCIP [34], and Artelys Knitro [35] support other techniques, such as cutting planes, that accelerate the solution-search process.

IV. Simulation Studies

In this section, we present simulation results that demonstrate the performance of the proposed approach. Consider a quadrotor that aims to travel from a starting point y_{init} to a goal point y_{goal} in a 3D space. Dynamic obstacles interfere with the quadrotor's possible paths. The position of a 6 DOF quadrotor can be expressed in the space of (x, y, z, ϕ, θ, ψ). The first three variables—x, y, and z—represent the distances of the quadrotor's center of mass along the X, Y and Z axes, respectively, from an Earth-fixed frame, whereas ϕ, θ, and ψ are the three Euler angles that represent the orientation of the quadrotor. Note that ϕ, θ, and ψ are the roll, pitch, and yaw angles about the X, Y, and Z axes, respectively. The dynamics of the quadrotor can then be modeled as $\ddot{\mathrm{x}} = -g\theta$, $\ddot{\mathrm{y}} = g\phi$, $\ddot{\mathrm{z}} = -\frac{1}{m}u_1$, $\ddot{\phi} = \frac{l}{I_{xx}}u_2$, $\ddot{\theta} = \frac{l}{I_{yy}}u_3$, $\ddot{\psi} = \frac{l}{I_{yy}}u_4$, where m is the quadrotor's mass, g is the gravitational acceleration, and I_{xx}, I_{yy}, and I_{zz} are the area moments of inertia about the principle axes in the body frame. Accordingly, the state space model of the quadrotor has 12 states, including its position and orientation in 3D space, as well as the corresponding velocities and rates. The output is chosen as the (x, y, z) position of the quadrotor.[6] The following parameters were employed throughout simulation [36]: $m = 0.65\ kg$, $l = 0.23\ m$, $I_{xx} = 0.0075\ kg \cdot m^2$, $I_{yy} = 0.0075\ kg \cdot m^2$, $I_{zz} = 0.013\ kg \cdot m^2$, $g = 9.81\ ms^2$.

[6]The rotors can be operated within a specific range of velocity. Thus, the input feasible set $\mathcal{U} := \{\boldsymbol{u} \in \mathbb{R}^4 \mid u_{\min} \leq \boldsymbol{u} \leq u_{\max}\}$ has been selected according to the chosen motor specifications. The set $\mathcal{X} := \{\boldsymbol{x} \mid -\pi \leq \phi \leq \pi \; , \; -\frac{\pi}{2} \leq \theta \leq \frac{\pi}{2}, \; -\pi \leq \psi \leq \pi\}$ has been selected to limit the angles to avoid kinematic singularity.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LRA.2019.2929980, IEEE Robotics and Automation Letters

As the first step, RRT* is used to generate a safe reference trajectory given the initial configuration of the obstacles. The quadrotor starts tracking the reference trajectory by using the receding horizon controller obtained by solving the MICP (11) using Gurobi 8.1.0. The MPC horizon and the number of time steps are selected as $K = 15$ and $T = 50$, respectively, and the weights in the stage-wise cost function (6) are chosen as $Q = I$ and $R = 0.01I$.

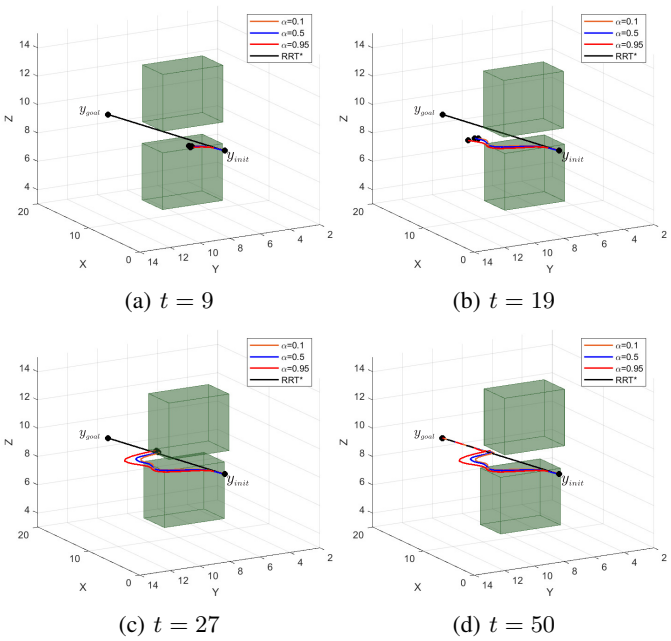
A. Effect of Confidence Level

In the first scenario, two randomly translating obstacles are interfering with the quadcopter's initially feasible optimal path. To demonstrate the the effect of confidence level, we consider three different cases with $\alpha = 0.1, 0.5, 0.95$ and $\delta = 0.04$. The resulting trajectories are shown in Fig. 2. The simulations were performed with $N_k = 20$ samples, where the random movement of each obstacle in each stage $t + k$ is uniformly distributed over $[-0.4, 0.4]^3$ and summed up from 1 to k to form $\hat{w}^{(i)}_{\ell,t,k}$.

In early stages (before $t = 9$), for all α's, the quadrotor deviates from its reference trajectory, even if it is not close to any obstacles, as shown in Fig. 2 (a). This is because the robot should always satisfy all the CVaR constraints within the prediction horizon. Until $t = 19$, the quadrotor is close to the obstacles, as shown in Fig. 2 (b). With the $\alpha = 0.95$, the robot makes sure that it is a safe margin away from the obstacles, so as not to violate the CVaR constraints within the prediction horizon. Such a deviation also occurs when $\alpha = 0.5, 0.1$, but with less magnitude than in the previous case. Fig. 2 (c) shows the quadrotor's position after it passes the obstacles at $t = 27$. The quadrotor starts to follow the reference trajectory without deviation. The complete trajectories are shown in Fig. 2 (d).

To empirically demonstrate the safety guarantees in the proposed SAA method, we test if the CVaR value estimated by SAA and Monte Carlo simulations satisfies the original risk constraint. We first calculate SAA-CVaR by using (4) with the trajectories generated by SAA-MPC with $N_k = 20$ samples. In addition, given the risk-aware controller, we performed Monte Carlo simulations using 10,000 new samples generated at each time step to compute MC-CVaR. The results for the first obstacle for different confidence levels at $t = [10, 25]$ are summarized in Fig. 3. In other stages, both of these values are 0, as the vehicle is far from the obstacle. These results demonstrate the capability of this method in adjusting the safety and conservativeness. It can be noticed that as the confidence level α increases, the robot's risk aversion increases and this encourages conservative decisions that induce high control costs $\sum_{t=0}^{T} J(x(t), u(t))$ as shown in Table I. For all confidence levels, the SAA-CVaR and MC-CVaR are always strictly less than the risk tolerance level δ. This confirms that the safety risk constraint (5f) is satisfied even when the proposed SAA method uses small number of samples ($N_k = 20$). The computational time also depends on the confidence level.

In order to compare the proposed method with existing chance-constrained approach, we also implemented MPC problem with sampling-based chance-constraints using the



(a) $t = 9$ (b) $t = 19$

(c) $t = 27$ (d) $t = 50$

Fig. 2: Generated quadrotor trajectories at difference stages with confidence levels $\alpha = 0.1, 0.5, 0.95$.

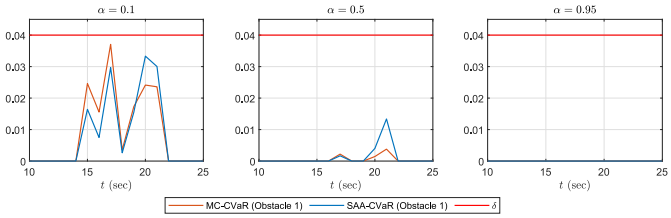


Fig. 3: Comparison of SAA-CVaR and MC-CVaR to the threshold δ for $\alpha = 0.1, 0.5, 0.95$.

results from [37] and [14]. Specifically, the following chance constraint is used: $\Pr[\text{dist}(y_k, \mathcal{Y}_{\ell,t,k}) = 0] \geq 0.95$. For comparison, we use $\delta = 0$ and $\alpha = 0.95$ in the CVaR constraint.[7] Fig. 4 shows the trajectories obtained by the chance-constrained approach and our method. The result obtained by CVaR-constrained optimization is safer than the one obtained by chance-constrained approach. This is explained by the fact that CVaR takes into account the tail events, while chance-constraint performs decision making only for the $(1-\alpha)$-worst case quantile. In fact, $\text{CVaR}_\alpha(X) \geq \text{VaR}_\alpha(X)$ by definition, and thus CVaR induces a safer decision-making than VaR or chance-constraints.

B. Effect of Sample Size

In the second scenario, the rotational motion of the first obstacle is uniformly distributed over $[-0.1, 0.1]^3$. In addition to the random rotation, a deterministic translation is considered so that the obstacle obstructs the nominal trajectory as time goes. On the other hand, both of the obstacles present random translational motion, sampled from a Gaussian distribution

[7]The risk tolerance level $\delta = 0$ induces the CVaR constraint to become a hard (deterministic) constraint.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LRA.2019.2929980, IEEE Robotics and Automation Letters

TABLE I: Results for Scenario 1 ($N_k = 20$, $\delta = 0.04$)

α	0.1	0.5	0.95
Computation time (sec)	697.44	533.34	175.65
Total cost	5228.0	1253.2	1323.3

TABLE II: Results for Scenario 2 ($\alpha = 0.99$, $\delta = 0.02$)

N_k	50	90	100
Computation time (sec)	799.72	2167.33	2580.51
Total cost	1589.57	1726.47	1737.16

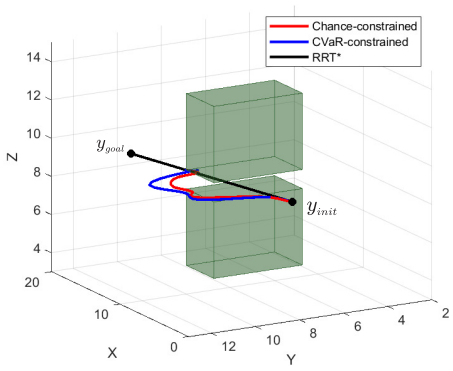
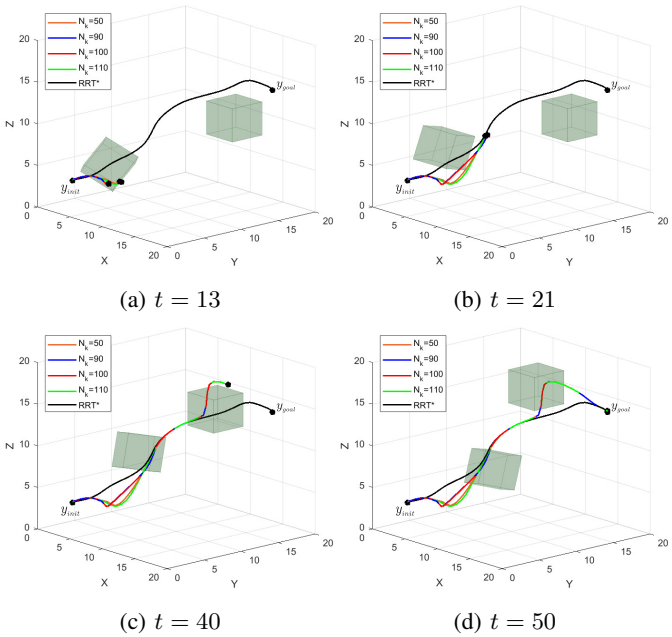


Fig. 4: Comparison between the chance-constrained and CVaR-constrained motion control.



(a) $t = 13$ (b) $t = 21$

(c) $t = 40$ (d) $t = 50$

Fig. 5: Generated quadrotor trajectories for different number of samples $N_k = 50, 90, 100, 110$.

with mean $\mu = 0$ and covariance $\sigma = 0.01$ in each stage. Fig. 5 illustrates the simulation results for different numbers of samples to observe their effect on decision making. In all cases, the risk tolerance and confidence levels were chosen as $\delta = 0.02$ and $\alpha = 0.95$, respectively. It can be seen from Table II that the computation time increases with the number of samples, as expected. In addition, we examine the effect of N_k on the total cost and CVaR. As shown in Table II, the cost value converges as the number of sample data increases, which can also be seen in Fig. 5, where the trajectories generated for $N_k = 100$ and $N_k = 110$ are the same. This convergence is consistent with Theorem 2. Regarding SAA-CVaR and MC-CVaR, both of them are equal to 0 for all N_k's because that big α and small δ are used. The second scenario requires more random samples than the first scenario because of two factors: the obstacles configuration and the existence of additional rotational uncertainties.

V. Conclusions

A risk-aware motion planning and control approach has been presented for robots operating in uncertain and dynamic environments. Our strategy consists of two stages: (i) generating a safe reference trajectory by using RRT*, and (ii) utilizing CVaR to assess safety risks and design a CVaR constrained receding horizon controller to track the reference trajectory. A computationally tractable solution to the MPC problem has been developed using the following three procedures. First, we reformulated the CVaR constraints without loss of optimality. Second, we proposed a convergent SAA method to completely remove multi-level optimization. Third, the nonconvexity of safe regions was addressed by recasting the MPC problem as a linearly constrained mixed integer convex program. Simulations using a quadrotor in a 3D environment demonstrate this method's capability to systematically adjust the safety and conservativeness in motion planning and control, as well as the effect of sample size on risk-aware decision-making.

Appendix A
Proof of Theorem 1

Proof. Let $J^\star$ and J' be the optimal value of the original and reformulated MPC problems, respectively. We first show that $J^\star = J'$. Let $(\mathbf{u}^\star, \mathbf{x}^\star, \mathbf{y}^\star)$ be an optimal solution of the original MPC problem (5). By the definition of CVaR, $\mathrm{CVaR}_\alpha[\mathrm{dist}(y_k^\star, \mathcal{Y}_\ell(t+k))] = \min_{z_{\ell,k}\in\mathbb{R}} \mathbb{E}[z_{\ell,k} + \frac{(\mathrm{dist}(y_k, \mathcal{Y}_\ell(t+k)) - z_{\ell,k})^+}{1-\alpha}]$ for each ℓ and k. Let $z_{\ell,k}^\star \in \mathbb{R}$ be an optimal solution of the minimization problem on the right-hand side. Due to the constraint (5f), we have $\mathbb{E}[z_{\ell,k}^\star + \frac{(\mathrm{dist}(y_k^\star, \mathcal{Y}_\ell(t+k)) - z_{\ell,k}^\star)^+}{1-\alpha}] \leq \delta_\ell$. By setting $h_{\ell,k}^\star = \mathrm{dist}(y_k^\star, \mathcal{Y}_\ell(t+k))$, we see that $(y_k^\star, z_{\ell,k}^\star)$ satisfies the constraints (7b) and (7e) in the reformulated problem, since $h_{\ell,k}^\star$ satisfies the condition in (7c) and (7d). All the other constraints in the reformulated problem also appear in the original one. Thus, $(\mathbf{u}^\star, \mathbf{x}^\star, \mathbf{y}^\star, \mathbf{z}^\star, \mathbf{h}^\star)$ satisfies all the constraints (7b)–(7f). This implies that $J' \leq J(x(t), \mathbf{u}^\star) = J^\star$.

We now show that $J' \geq J^\star$. Let $(\mathbf{u}', \mathbf{x}', \mathbf{y}', \mathbf{z}', \mathbf{h}')$ be an optimal solution of the reformulated stochastic program. We first observe that

$$\begin{aligned}
&\mathrm{CVaR}_\alpha[\mathrm{dist}(y_k', \mathcal{Y}_\ell(t+k))] \\
&= \min_{z\in\mathbb{R}} \mathbb{E}\left[z_{\ell,k} + \frac{(\mathrm{dist}(y_k', \mathcal{Y}_\ell(t+k)) - z_{\ell,k})^+}{1-\alpha} \right] \\
&\leq \mathbb{E}\left[z_{\ell,k}' + \frac{(h_{\ell,k}' - z_{\ell,k}')^+}{1-\alpha} \right] \leq \delta_\ell,
\end{aligned}$$

where the first inequality is valid due to the constraint (7c) and (7d) and last inequality is valid due to the constraint (7b). Therefore, y'_k satisfies the CVaR constraint (5f) in the original problem. All the other constraints in (5) clearly hold with $(\mathbf{u}', \mathbf{x}', \mathbf{y}')$. Thus, we have $J^\star \leq J(x(t), \mathbf{u}') = J'$.

We conclude that $J^\star = J(x(t), \mathbf{u}^\star) = J(x(t), \mathbf{u}') = J'$. Therefore, any optimal $\mathbf{u}^\star$ of the original MPC problem is optimal to the reformulated one, and conversely any optimal $\mathbf{u}'$ of the reformulated problem is optimal to the original one, as desired. □

APPENDIX B
PROOF OF THEOREM 2

Proof. We call the sample average approximation of (7) as SAA-MPC. Applying SAA to (7) replaces the constraint (7b) with (8). Let J_N^{SAA} and $\mathbb{U}_N^{\text{SAA}}$ denote the optimal value and the set of optimal $\mathbf{u}$ of the SAA-MPC problem. It is trivial that $J_N^{\text{SAA}} = J_N$ and $\mathbb{U}_N^{\text{SAA}} = \mathbb{U}_N$, thus (7) is equivalent to (9).

It now suffices to show that

$$J_N^{\text{SAA}} \to J^\star \quad \text{and} \quad D(\mathbb{U}_N^{\text{SAA}}, \mathbb{U}^\star) \to 0 \quad \text{as } N \to \infty.$$

Due to the continuity of the function f and Assumption 1, all the conditions of Theorems 5.3 and 5.5 in Shapiro et al. [38] are satisfied. Therefore, the result follows. □

REFERENCES

[1] M. Hoy, A. S. Matveev, and A. V. Savkin, "Algorithms for collision-free navigation of mobile robots in complex cluttered environments: a survey," *Robotica*, vol. 33, no. 3, pp. 463–497, 2015.

[2] Y. Kuwata, T. Schouwenaars, A. Richards, and J. How, "Robust constrained receding horizon control for trajectory planning," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2005.

[3] S. L. Herbert, M. Chen, S. Han, S. Bansal, J. F. Fisac, and C. J. Tomlin, "FaSTrack: a modular framework for fast and guaranteed safe motion planning," in *IEEE 56th Annual Conference on Decision and Control*, 2017.

[4] A. Majumdar and R. Tedrake, "Funnel libraries for real-time robust feedback motion planning," *The International Journal of Robotics Research*, vol. 36, no. 8, pp. 947–982, 2017.

[5] D. Fridovich-Keil, S. L. Herbert, J. F. Fisac, S. Deglurkar, and C. J. Tomlin, "Planning, fast and slow: A framework for adaptive real-time safe trajectory planning," in *IEEE International Conference on Robotics and Automation*, 2018.

[6] 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, 2011.

[7] N. E. Du Toit and J. W. Burdick, "Robot motion planning in dynamic, uncertain environments," *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 101–115, 2012.

[8] M. Ono, M. Pavone, Y. Kuwata, and J. Balaram, "Chance-constrained dynamic programming with application to risk-aware robotic space exploration," *Autonomous Robots*, vol. 39, pp. 555–571, 2015.

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

[10] A. Bry and N. Roy, "Rapidly-exploring random belief trees for motion planning under uncertainty," in *IEEE International Conference on Robotics and Automation*, 2011.

[11] A. Jasour, N. S. Aybat, and C. M. Lagoa, "Semidefinite programming for chance constrained optimization over semialgebraic sets," *SIAM Journal on Optimization*, vol. 25, no. 3, pp. 1411–1440, 2015.

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

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

[14] 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.

[15] R. T. Rockafellar and S. Uryasev, "Optimization of conditional value-at-risk," *Journal of Risk*, vol. 2, pp. 21–42, 2000.

[16] ——, "Conditional value-at-risk for general loss distribution," *Journal of Banking & Finance*, vol. 26, pp. 1443–1471, 2002.

[17] P. Artzner, F. Delbaen, J.-M. Eber, and D. Heath, "Coherent measures of risk," *Mathematical Finance*, vol. 9, no. 3, pp. 203–228, 1999.

[18] A. Majumdar and M. Pavone, "How should a robot assess risk? towards an axiomatic theory of risk in robotics," in *International Symposium on Robotics Research*, 2017.

[19] R. T. Rockafellar, S. P. Uryasev, and M. Zabarankin, "Deviation measures in risk analysis and optimization," *University of Florida, Department of Industrial & Systems Engineering Working Paper*, 2002.

[20] S. Samuelson and I. Yang, "Safety-aware optimal control of stochastic systems using conditional value-at-risk," in *American Control Conference*, 2018.

[21] T. Schouwenaars, A. Stubbs, J. Paduano, and E. Feron, "Multivehicle path planning for nonline-of-sight communication," *Journal of Field Robotics*, vol. 23, no. 3-4, pp. 269–290, 2006.

[22] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.

[23] S. M. Lavalle, "Rapidly-Exploring Random Trees: A New Tool for Path Planning," Tech. Rep., 1998.

[24] C. W. Miller and I. Yang, "Optimal control of conditional value-at-risk in continuous time," *SIAM Journal on Control and Optimization*, vol. 55, no. 2, pp. 856–884, 2017.

[25] S. Singh, Y.-L. Chow, A. Majumdar, and M. Pavone, "A framework for time-consistent, risk-sensitive model predictive control: Theory and algorithms," *IEEE Transactions on Automatic Control*, to appear.

[26] D. Bertsimas, V. Gupta, and N. Kallus, "Robust sample average approximation," *Mathematical Programming*, vol. 171, pp. 217–282, 2018.

[27] P. Mohajerin Esfahani and D. Kuhn, "Data-driven distributionally robust optimization using the Wasserstein metric: performance guarantees and tractable reformulations," *Mathematical Programming*, vol. 171, pp. 115–166, 2018.

[28] I. Yang, "Wasserstein distributionally robust stochastic control: A data-driven approach," *arXiv preprint arXiv:1812.09808.*, 2018.

[29] A. Vecchietti, S. Lee, and I. E. Grossmann, "Modeling of discrete/continuous optimization problems: Characterization and formulation of disjunctions and their relaxations," *Computers and Chemical Engineering*, vol. 27, no. 3, pp. 433–448, 2003.

[30] P. Bonami, J. Lee, S. Leyffer, and A. Wächter, "On branching rules for convex mixed-integer nonlinear optimization," *Journal of Experimental Algorithmics*, vol. 18, pp. 2–6, 2013.

[31] M. A. Duran and I. E. Grossmann, "An outer-approximation algorithm for a class of mixed-integer nonlinear programs," *Mathematical Programming*, vol. 36, no. 3, pp. 307–339, 1986.

[32] M. Lubin, E. Yamangil, R. Bent, and J. P. Vielma, "Polyhedral approximation in mixed-integer convex optimization," *Mathematical Programming*, vol. 172, no. 1-2, pp. 139–168, 2018.

[33] P. Bonami, L. T. Biegler, A. R. Conn, G. Cornuéjols, I. E. Grossmann, C. D. Laird, J. Lee, A. Lodi, F. Margot, N. Sawaya, and A. Wächter, "An algorithmic framework for convex mixed integer nonlinear programs," *Discrete Optimization*, vol. 5, no. 2, pp. 186–204, 2008.

[34] T. Achterberg, "SCIP: solving constraint integer programs," *Mathematical Programming Computation*, vol. 1, no. 1, pp. 1–41, 2009.

[35] R. H. Byrd, J. Nocedal, and R. A. Waltz, "Knitro: An integrated package for nonlinear optimization," in *Large-Scale Nonlinear Optimization*. Springer, 2006, pp. 35–59.

[36] A. Nagaty, S. Saeedi, C. Thibault, M. Seto, and H. Li, "Control and navigation framework for quadrotor helicopters," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 70, no. 1-4, pp. 1–12, 2013.

[37] L. Blackmore, "A probabilistic particle control approach to optimal, robust predictive control," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2006.

[38] A. Shapiro, D. Dentcheva, and A. Ruszczyński, *Lectures on Stochastic Programming: Modeling and Theory*, 2nd ed. SIAM, 2014.