Robust Motion Planning in the Presence of Estimation Uncertainty

Lars Lindemann, Matthew Cleaveland, Yiannis Kantaros, and George J. Pappas

***Abstract*— Motion planning is a fundamental problem and focuses on finding control inputs that enable a robot to reach a goal region while safely avoiding obstacles. However, in many situations, the state of the system may not be known but only estimated using, for instance, a Kalman filter. This results in a novel motion planning problem where safety must be ensured in the presence of state uncertainty. Previous approaches to this problem are either conservative or integrate state estimates optimistically which leads to non-robust solutions. Optimistic solutions require frequent replanning to not endanger the safety of the system. We propose a new formulation to this problem with the aim to be robust to state estimation errors while not being overly conservative. In particular, we formulate a stochastic optimal control problem that contains robustified risk-aware safety constraints by incorporating robustness margins to account for state estimation errors. We propose a novel sampling-based approach that builds trees exploring the reachable space of Gaussian distributions that capture uncertainty both in state estimation and in future measurements. We provide robustness guarantees and show, both in theory and on case studies, that the induced robustness margins constitute a trade-off between conservatism and robustness in motion planning under state uncertainty that allows to control the frequency of replanning.**

I. Introduction

Motion planning is a fundamental problem that has received considerable research attention over the past years [1]. Typically, the motion planning problem aims to generate trajectories that reach a desired goal state starting from an initial configuration while avoiding unsafe states (e.g., obstacles) under the assumption that there is no uncertainty about the system's state or its safe state space [2], [3].

In this paper, we consider the problem of robust motion planning in the presence of state uncertainty. In particular, we consider the case where the goal is to control a linear system to reach a desired final state while avoiding known unsafe states. To account for uncertainty in the system's state, due to noisy sensors and exogenous disturbances, we require the system's state to always respect risk-aware safety and reachability requirements. First we formulate this problem as a stochastic optimal control problem that generates control policies that rely on future sensor measurements. To avoid the need of computationally expensive approaches that require control in belief space [4]–[9], we approximate this problem with an approximate stochastic optimal control problem that generates robust and uncertainty-aware control policies. Specifically, the generated policies are *uncertainty-aware* in the sense that they take into account how localization uncertainty may evolve under the execution of these controllers using the Kalman filter Ricatti map [10] and *robust* as they guarantee that risk-aware reachability and safety constraints are met even when the predicted uncertainty and state estimate deviate from the actual ones. In this paper, to solve this approximate stochastic optimal control problem, we build upon the RRT* algorithm [3] and we propose a new sampling-based approach that relies on searching the space of Gaussian distributions for the system's state. We provide correctness guarantees with respect to the approximate control problem and show how those guarantees relate to the original stochastic control problem. Our framework allows to integrate offline robust planning with online replanning. We argue, and show in simulations, that the robustness margins considered for offline planning constitute a fundamental trade-off between conservatism and robustness in the presence of state uncertainty that allows to control the frequency of replanning needed.

Related Literature: Sampling-based approaches for planning in the presence of uncertainty have gained growing interest and several works have appeared in the past years. Specifically, [11], [12] propose CC-RRT, an extension of [3], that incorporates chance constraints to ensure probabilistic feasibility for linear systems subject to process noise. Extensions to account for uncertain dynamic obstacles are considered in [13], [14]. Common in these works is that they lack robustness to state uncertainty and that planning is not integrated with sensing, and, therefore, they are quite conservative as uncertainty may grow unbounded. Sampling-based approaches that exhibit robustness to process noise have been proposed in [15], [16] as well, without considering sensing information while designing paths. The authors in [17] have considered an unscented transformation to estimate state distributions for nonlinear systems and by considering moment-based ambiguity sets to account for potential errors. The authors in [18], [19] propose model predictive control frameworks that contain distributionally robust risk constraints that are based on ambiguity sets defined around an empirical state distribution from input-output data. On the other hand, sensor-based sampling-based approaches that take into account future measurements to design paths using Kalman or particle filters have been proposed in [20]–[22], respectively. Nevertheless, unlike our work, these methods lack robustness to state uncertainty. The authors in [23] have proposed to model such uncertainty in a stochastic way and by interpreting output measurements as distributions. In this work, we instead enforce robustness by imposing hard

The authors are with the GRASP Laboratory, University of Pennsylvania, Philadelphia, PA 19104, USA. Email:{larsl,mcleav, kantaros, pappasg}@seas.upenn.edu This material is based upon work supported by the Air Force Research Laboratory (AFRL) and the Defense Advanced Research Projects Agency (DARPA) under Contract No. FA8750-18-C-0090.

robustness margins around a nominal trajectory for planning under state uncertainty.

Contribution: The contribution of this paper can be summarized as follows. *First*, we formulate the motion planning problem under state uncertainty as a stochastic optimal control problem and propose a tractable approximation that is based on a robust reformulation using nominal predictions. *Second*, we propose a sampling-based algorithm towards solving the approximate problem and we show in what way our solution relates to the original problem. *Third*, we show that our framework allows to integrate offline robust planning with online replanning. The robustness margin considered for offline planning directly affects the frequency of replanning and hence shows a fundamental trade-off between conservatism and robustness as shown in our simulation studies.

II. Background

Let $\mathbb{R}$ and $\mathbb{N}$ be the set of real and natural numbers. Also let $\mathbb{N}_{\geq 0}$ be the set of non-negative natural numbers and $\mathbb{R}^n$ be the real n-dimensional vector space. Let $\mathcal{N}:\mathbb{R}^n\times\mathbb{R}^{n\times n}$ be an n-dimensional Gaussian distribution. The appendix, to which we later in the paper refer, can be found in https://github.com/Lindemann1989/KalmanRRT.

A. Random Variables and Risk Theory

Consider the *probability space* $(\Omega,\mathcal{F},P)$ where Ω is the sample space, $\mathcal{F}$ is a σ-algebra of Ω, and $P:\mathcal{F}\to[0,1]$ is a probability measure. More intuitively, an element in Ω is an *outcome* of an experiment, while an element in $\mathcal{F}$ is an *event* that consists of one or more outcomes whose probabilities can be measured by the probability measure P.

1) Random Variables: Let X denote a real-valued *random vector*, i.e., a measurable function $X:\Omega\to\mathbb{R}^n$. We refer to $X(\omega)$ as a realization of the random vector X where $\omega\in\Omega$. Since X is a measurable function, a probability space can be defined for X so that probabilities can be assigned to events related to values of X.

2) Risk Theory: Let $\mathfrak{F}(\Omega,\mathbb{R})$ denote the set of measurable functions mapping from the domain Ω into the domain $\mathbb{R}$. A *risk measure* is a function $R:\mathfrak{F}(\Omega,\mathbb{R})\to\mathbb{R}$ that maps from the set of real-valued random variables to the real numbers. In particular, we refer to the input of a risk measure R as the *cost random variable* since typically a cost is associated with the input of R. Risk measures allow for a risk assessment in terms of such cost random variables. Commonly used risk measures are the expected value, the variance, or the conditional value-at-risk [24]. In Appendix I, we summarize desireable properties of R and provide a summary of existing risk measures.

B. Stochastic Control System

Consider the discrete-time stochastic control system

$$X(t+1)=AX(t)+Bu(t)+W(t),\ X(0):=X_0 \tag{1a}$$
$$Y(t)=CX(t)+V(t) \tag{1b}$$

where $X(t)\in\mathbb{R}^n$, $u(t)\in\mathcal{U}\subseteq\mathbb{R}^m$, $Y(t)\in\mathbb{R}^p$ denote the state of the system, the control input, and the measurement at time t. The set $\mathcal{U}$ denotes the set of admissible control inputs. Also, $W(t)\in\mathbb{R}^n$ and $V(t)\in\mathbb{R}^p$ denote the state disturbance and the measurement noise at time t and they are assumed to follow a Gaussian distribution, i.e., $W(t)\sim\mathcal{N}(\mu_W,\Sigma_W)$ and $V(t)\sim\mathcal{N}(\mu_V,\Sigma_V)$, with known mean vectors μ_W and μ_V and known covariance matrices Σ_W and Σ_V. The initial condition $X_0\in\mathbb{R}^n$ also follows a Gaussian distribution, i.e, $X_0\sim\mathcal{N}(\mu(0),\Sigma(0))$. We assume that $X(0)$, $W(t)$, and $V(t)$ are mutually independent. Note that we have here dropped the underlying sample space Ω of $W(t)$, $V(t)$, and $X(0)$ for convenience.

C. Kalman Filter for State Estimation

Note that the state $X(t)$ and the output measurements $Y(t)$ also become Gaussian random variables since the system in (1) is linear in $W(t)$, $V(t)$, and $X(t)$. The state $X(t)$ defines a stochastic process with a mean $\mu(t)$ and a covariance matrix $\Sigma(t)$ that can recursively be calculated as stated in Appendix II. The estimates $\mu(t)$ and $\Sigma(t)$ are conservative and do not incorporate available output measurements, i.e., the realizations $y(t)$ of $Y(t)$. Let us denote the realized output measurements up until time t as $Y_t:=\begin{bmatrix}y(0)^T & \ldots & y(t)^T\end{bmatrix}^T$. We can now refine the above estimates to obtain optimal estimates based on Y_t by means of a Kalman filter. For $s\leq t$, let us for brevity define the random variable $X(t|s):=X(t)|Y_s$ as the random variable $X(t)$ conditioned on knowledge of the realized output measurements Y_s with the conditional mean $\mu(t|s)$ and the conditional covariance matrix $\Sigma(t|s)$. We can calculate these quantities recursively as

$$\mu(t+1|t+1)=F_\mu(\mu(t|t),u(t),y(t+1)) \tag{2a}$$
$$\Sigma(t+1|t+1)=F_\Sigma(\Sigma(t|t)) \tag{2b}$$

where the functions $F_\mu:\mathbb{R}^n\times\mathbb{R}^m\times\mathbb{R}^p\to\mathbb{R}^n$ and $F_\Sigma:\mathbb{R}^{n\times n}\to\mathbb{R}^{n\times n}$ are defined in Appendix II.

Remark 2.1: It holds that $\mu(t)=\mu(t|t)$ for $t\in\mathbb{N}_{\geq 0}$ if and only if the innovation term within the Kalman filter is zero for all $t\in\mathbb{N}_{\geq 0}$, i.e., if $y(t)=C\mu(t|t-1)$ for all $t\in\mathbb{N}_{\geq 0}$. However, if the innovation term is not always zero, i.e., if $y(t)\neq C\mu(t|t-1)$ for some $t\in\mathbb{N}_{\geq 0}$, then $\mu(t)\neq\mu(t|t)$ for at least one time $t\in\mathbb{N}_{\geq 0}$.

III. Problem Formulation

A. Risk-Aware Stochastic Optimal Control Problem

Consider a stochastic system of the form (1) operating in a compact environment occupied by M regions of interest $O_i\in\mathbb{R}^n$ for $i\in\mathcal{M}:=\{1,\ldots,M\}$. The goal is to safely navigate the system to the goal region O_1 by means of the control input $u(t)$. Additionally, the systems has to avoid or be close to regions $O_2,\ldots,O_M$. For instance, consider a robot that always has to avoid obstacles and eventually be within communication range of a static wifi spot. Consider therefore J measurable functions $d_j:\mathbb{R}^{nM}\to\mathbb{R}$ for $j\in\mathcal{J}:=\{1,\ldots,J\}$ that will encode such constraints. We aim to evaluate the function d_j based on the conditional state estimate $X(t|t)$. Since $X(t|t)$ is a random variable, the functions $d_j(X(t|t), O_2, \ldots, O_M)$ also become random variables. The problem that this paper addresses can be captured by the following *stochastic* optimal control problem:

$$\min_{H,U_H} \sum_{t=0}^{H} c(\mu(t|t)) \tag{3a}$$
$$\mu(t+1|t+1) = F_\mu(\mu(t|t), u(t), y(t+1)), \tag{3b}$$
$$\Sigma(t+1|t+1) = F_\Sigma(\Sigma(t|t)), \tag{3c}$$
$$R(\|X(H|H) - O_1\| - \kappa) \leq \gamma, \tag{3d}$$
$$R(-d_j(X(t|t), O_2, \ldots, O_M)) \leq \gamma_j, \forall j \in \mathcal{J} \tag{3e}$$

where $U_t := \begin{bmatrix} u(0)^T & \ldots & u(t)^T \end{bmatrix}^T$, $c : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ is a non-negative cost function, $\kappa > 0$ is an accuracy parameter for reaching O_1, and γ and γ_j are given risk thresholds.

B. The Approximate Stochastic Optimal Control Problem

The challenge in solving the optimization problem (3) lies in the dependence of the cost function (3a) and the constraints (3d) and (3e) on the realized output measurements. The mean $\mu(t|t)$ can not be calculated a priori due to the dependence on the realization $y(t)$ of the measurement as reflected in the dependence of F_μ on $y(t)$. Therefore, the optimization problem in (3) can not be solved a priori.

To address this challenge, we approximate the optimization problem (3) with an approximate stochastic optimal control problem by using $\mu(t|0)$ instead of $\mu(t|t)$ where we recall that $\mu(t|0)$ is computed using only the prediction step of the Kalman filter so that $\mu(t|0)$ is equivalent to the unconditional mean $\mu(t)$.[1] Similar ideas have been followed in [20], [21], [23], but without explicitly adding robustness margins to alleviate the lack of knowledge of Y_t during planning. To account for a potential mismatch between $\mu(t|0)$ and the in-the-field realization of $\mu(t|t)$, we introduce additional *robustness* margins.

In particular, we account for all realizations $y(t)$ that are such that $X(t|t)$ is 'ϵ-close' to the random variable $\hat{X}(t|t) \sim \mathcal{N}(\mu(t|0), \Sigma(t|t))$ that we use for planning.[2] To formalize the 'ϵ-closeness' notion, define the set

$$\mathcal{B}_\epsilon\big(\hat{X}(t|t)\big) := \big\{\mathcal{N}\big(\mu, \Sigma(t|t)\big) | \exists \mu \in \mathbb{R}^n, \|\mu - \mu(t|0)\|^2 \leq \epsilon\big\}$$

which contains all distributions with a covariance of $\Sigma(t|t)$ in an Euclidean ball of size ϵ around $\mu(t|0)$ where ϵ is a design robustness parameter. As it will be shown in Section V, the size of ϵ will determine the probability by which the constraints in (3) are satisfied. In particular, we let $\epsilon : \mathbb{R}^{n\times n} \times \mathbb{N}_{\geq 0} \to \mathbb{R}_{\geq 0}$ be a function that depends on A and t, i.e., $\epsilon(A, t)$. We will drop the dependence on A and t when it is clear from the context for ease of notation. Naturally, the discrepancy between $X(t|t)$ and $\hat{X}(t|t)$ increases with time so that a larger ϵ may be desired for larger t.

Thus, now we approximate the stochastic optimal control problem (3) with the following approximate stochastic optimal control problem:

$$\min_{H,U_H} \sum_{t=0}^{H} c(\mu(t|0)) \tag{4a}$$
$$\mu(t+1|0) = A\mu(t|0) + Bu(t) + \mu_W, \tag{4b}$$
$$\Sigma(t+1|t+1) = F_\Sigma(\Sigma(t|t)), \tag{4c}$$
$$\sup_{X \in \mathcal{B}_{\epsilon(A,H)}(\hat{X}(H|H))} R(\|X - O_1\| - \kappa) \leq \gamma, \tag{4d}$$
$$\sup_{X \in \mathcal{B}_{\epsilon(A,t)}(\hat{X}(t|t))} R(-d_j(X, O_2, \ldots, O_M)) \leq \gamma_j, \forall j \in \mathcal{J}. \tag{4e}$$

where $\hat{X}(t|t) \sim \mathcal{N}(\mu(t|0), \Sigma(t|t))$.

Problem 1: Given the system in (1) and the regions of interest O_i, determine a terminal horizon H and a sequence of control inputs U_H that solves (4).

Remark 3.1 (Approximation Gap): Note that a solution to (4) may not constitute a feasible solution to (3). The reason is that (3) relies on knowledge of measurements that will be taken in the future which is not the case in (4). Although, this is accounted by introducing the robustness parameter ϵ, the feasibility gap may still exist depending on the values of ϵ. In Section V, we show that the probability that a solution to (4) satisfies the constraints of (3) depends on ϵ. Moreover, to further account for this feasibility gap, in Section IV-C, we propose a re-planning framework that is triggered during the execution time when the robustness requirement $X(t|t) \in \mathcal{B}_{\epsilon(A,t)}(\hat{X}(t|t))$ is not met.

Remark 3.2 (Robustness): Note that in (4), although the main purpose of the robustness parameter ϵ is to account for uncertainty in the measurements that will be collected online, due to its generality it also provides robustness to other modeling uncertainties e.g., inaccurate model for the system, the process, or measurement noise.

Remark 3.3 (Sensor Model): The sensor model described in (1) can model e.g., a GPS-sensor, used for estimating the system's state, e.g., the position of a robot. By definition of this sensor model, it is linear and it does not interact with the environment. Nevertheless, more complex nonlinear sensor models that allow for environmental interaction can be considered (e.g., a range sensor). To account for such sensor models, an Extended Kalman filter (EKF) can be used. In this case, the constraints (4c) require linearization of the sensor model with respect to the hidden state; see e.g., [10].

IV. Robust Rapidly Exploring Random Tree (R-RRT*)

In the following three sections, we build upon the RRT* algorithm [3] and we present and propose a new robust sampling-based approach to solve (4) summarized in Algorithm 1. In particular, in Sections IV-A and IV-B we propose an offline planning algorithm that relies on building trees incrementally that simultaneously explore the reachable space of Gaussian distributions modeling the system's state. In Section IV-C, we propose a replanning algorithm for cases

[1] With respect to Remark 2.1, note that this approximation is equivalent to the innovation term in the Kalman filter update equations being always zero.

[2] Closeness here is in terms of the 2nd Wasserstein distance between $X(t|t)$ and $\hat{X}(t|t)$. Note in particular that the 2nd Wasserstein distance between $X(t|t)$ and $\hat{X}(t|t)$ is equivalent to $\|\mu(t|0) - \mu(t|t)\|^2$ as both $X(t|t)$ and $\hat{X}(t|t)$ are Gaussian with the same covariance matrix.

when $X(t|t) \notin \mathcal{B}_\epsilon(\hat{X}(t|t))$ during the online execution of the algorithm. In Section V, we present the theoretical guarantees that our approach comes with.

A. Tree Definition

In what follows, we denote the constructed directed tree by $G := (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} \subseteq \mathbb{R}^n \times \mathbb{R}^{n\times n} \times \mathbb{N}_{\geq 0}$ is the set of nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V} \times \mathcal{U}$ denotes the set of edges. The set $\mathcal{V}$ collects nodes v consisting of a nominal mean and a nominal covariance matrix that are denoted by $v.\mu$ and $v.\Sigma$, respectively, along with a time stamp $v.t$. We remark that each node will accept state distributions X that are such that $X \in \mathcal{B}_\epsilon\big(\mathcal{N}(v.\mu, v.\Sigma)\big)$. Also, the set $\mathcal{E}$ collects edges from a node v to another node v' along with a control input u, if the state distribution in v' is reachable from v as per the constraints (4b)-(4c) and the control input u. Also, the cost of reaching a node v' with parent node v is

$$\text{Cost}(v') = \text{Cost}(v) + c(v'.\mu). \tag{5}$$

Observe that by applying (5) recursively, we get that $\text{Cost}(v')$ is the objective function in (4). Also, the tree is rooted at a node v_0 capturing the initial robot state, i.e., $v_0.\mu = \mu(0|0)$, $v_0.\Sigma = \Sigma(0|0)$, $v_0.t = 0$, and $\text{Cost}(v_0) = c(v_0.\mu)$. Construction of the tree occurs in incremental fashion where within a single iteration (i) a new node is sampled; (ii) the tree is extended towards this new sample, if possible, and (iii) a rewiring operation follows aiming to decrease the cost of existing tree nodes by leveraging the newly added node. After taking $N \geq 0$ samples [line 3, Alg. 1], where N is user-specified, Algorithm 1 terminates and returns a solution to Problem 4, if it has been found, i.e., a terminal horizon H and a sequence of control inputs U_H.

To extract such a solution, we first need to define the set $\mathcal{V}_G \subseteq \mathcal{V}$ that collects all nodes $v \in \mathcal{V}$ of the tree that satisfy the terminal constraint (4d). Then, among all nodes in $\mathcal{V}_G$, we select the node $v \in \mathcal{V}_G$ with the smallest cost $C(v)$. Then, the terminal horizon is H, and the control inputs U_H are recovered by computing the path in G that connects the selected node $v \in \mathcal{V}_G$ to the root v_0. Note that satisfaction of the remaining constraints in (4) is guaranteed by construction of the tree G. A detailed description of the steps required to construct the tree follow.

B. Incremental Construction of Tree

SAMPLE: At every iteration i of Algorithm 1, we first generate a sample from the space of Gaussian distributions modelling the system's uncertain state. To sample from the set of Gaussian distributions, we define the function $\text{SAMPLE} : \mathbb{N}_{\geq 0} \times \Omega \to \mathbb{R}^n \times S_{\geq 0}$ that generates independent and identically distributed samples, denoted by s_{rand}, of means μ and covariances Σ where $S_{\geq 0} \subseteq \mathbb{R}^{n\times n}$ is the set of positive semidefinite matrices [line 4, Alg. 1]. For simplicity, assume that the distribution of SAMPLE is uniform. For brevity, we use $\text{SAMPLE}(i)$ instead of $\text{SAMPLE}(i, \omega)$ and again omit the underlying sample space Ω.

Remark 4.1 (Sampling): Traditional RRT* algorithms draw samples directly from the obstacle-free space. Note that this is here not possible due to the stochastic setup where a node consists of a nominal mean and a covariance that need to be dynamically feasible according to (4b) and (4c) with respect to the parent node while satisfying the constraint (4e).

NEAREST: Next, given the sample $s_{\text{rand}} \in \mathbb{R}^n \times S_{\geq 0}$, among all nodes in the current tree structure, we pick the closest one to s_{rand}, denoted by v_{nearest} [line 5, Alg. 1]. To this end, we define the following function:

$$\text{NEAREST}(\mathcal{V}, \mathcal{E}, s_{\text{rand}}) := \underset{v\in\mathcal{V}}{\text{argmin}} \, \|\mathcal{N}(s_{\text{rand}}) - \mathcal{N}(v.\mu, v.\Sigma)\|$$

where $\|\cdot\|$ can be any norm between distributions, e.g., the 2nd Wasserstein distance.[3] The function $\text{NEAREST}(\mathcal{V}, \mathcal{E}, s)$ returns the node v_{nearest} in $\mathcal{V}$ that is closest to s_{rand}.

STEER: In order to steer the tree from a node v towards s [lines 6, 13, and 25, Alg. 1], we use the function $\text{STEER}(v, s, \eta) := (v^*, u^*)$, where $\eta \geq 0$ is a positive constant, and returns the closest to s dynamically feasible node $v^* := (\mu^*, \Sigma^*, v.t + 1)$ that is still η-close to the node v. In particular, the mean μ^* and the covariance Σ^* are required to be dynamically feasible as per constraints (4b)-(4c). The steer function aims to solve the following optimization problem:

$$\begin{aligned}
(\mu^*, \Sigma^*, u^*) := \underset{(\mu,\Sigma,u)\in\mathbb{R}^n\times S_{\geq 0}\times\mathbb{R}^m}{\text{argmin}} \; & \|\mathcal{N}(s) - \mathcal{N}(\mu, \Sigma)\| \\
\text{s.t. } & \|\mathcal{N}(v.\mu, v.\Sigma) - \mathcal{N}(\mu, \Sigma)\| \leq \eta \\
& \mu = Av.\mu + Bu + \mu_W \qquad (6) \\
& \Sigma = F_\Sigma(v.\Sigma) \qquad (7) \\
& u \in \mathcal{U}. \qquad (8)
\end{aligned}$$

CHECKCONSTRROB: We next check if the new node v_{new} is feasible i.e., if the constraint (4e) holds for all distributions accepted by v_{new} [line 7, Alg. 1]., i.e., if

$$\sup_{X\in\mathcal{B}_{\epsilon(A, v_{\text{new}}.t)}(\hat{X})} R(-d_j(X, O_2, \ldots, O_M)) \leq \gamma_j, \forall j \in \mathcal{J}. \tag{9}$$

where $\hat{X} \sim \mathcal{N}(v_{\text{new}}.\mu, v_{\text{new}}.\Sigma)$. Note that checking (9) may or may not pose computational burden in general depending on the function d_j as well as the risk measure R. While, in general, sampling-based approaches for checking the constraint are viable, e.g, [24], there also exist tractable reformulation, e.g., for the conditional value-at-risk [19] when the supremum is taken over Wasserstein distance balls.

NEAR: If the transition from v_{nearest} to v_{new} is feasible, we check if there is any other candidate parent node for v_{new} that can incur to v_{new} a lower cost than the one when v_{nearest} is the parent node. The candidate parent nodes are selected from the following set [line 9, Alg. 1]

$$\begin{aligned}
&\text{NEAR}(\mathcal{V}, \mathcal{E}, s_{\text{new}}, \eta) := \\
&\{v \in V \,|\, \|\mathcal{N}(s_{\text{new}}) - \mathcal{N}(v.\mu, v.\Sigma)\| \leq r(\mathcal{V})\}.
\end{aligned} \tag{10}$$

[3] Alternatively to using a norm between distributions, one can define and use a norm on $\mathbb{R}^n \times S_{\geq 0}$. We emphasize that this choice does not affect the theoretical guarantees we provide in Section V, but it may affect the way the tree is guided to grow and hence affect computation times.

where the $r(\mathcal{V})$ is defined as follows:

$$r(\mathcal{V}) = \min\Big\{\gamma\Big(\frac{\log|\mathcal{V}|}{|\mathcal{V}|}\Big)^{\frac{1}{n}}, \eta\Big\}.$$

where $\gamma > 0$. In words, the set in (10) collects all nodes $v \in \mathcal{V}$ that are within at most a distance of r from s_{new} in terms of the the norm $\|\cdot\|$. Among all candidate parents, we pick the one that incurs the minimum cost while ensuring that transition from it towards v_{new} is feasible; the selected parent node is denoted by v_{new} [lines 9-18, Alg. 1]. Next, the sets of nodes and edges of the tree are accordingly updated [lines 19-20, Alg. 1].

GOALREACHED: Once a new node v_{new} is added to the tree, we check if its respective Gaussian distribution satisfies the terminal constraint (4d). This is accomplished by the function GOALREACHED [line (21), Alg. 1]. This function resembles the function CHECKCONSTRROB, but it focuses on the terminal constraint (4d) instead of (4e). Specifically, for a given node $v_{\text{new}} \in V$, the function GOALREACHED(v_{new}) checks robust satisfaction of the goal constraint, i.e., whether or not the following condition is met:

$$\sup_{X \in \mathcal{B}_{\epsilon(A, v_{\text{new}}.t)}(\hat{X})} R(\|X - O_1\| - \kappa) \leq \gamma.$$

where $\hat{X} \sim \mathcal{N}(v_{\text{new}}.\mu, v_{\text{new}}.\Sigma)$. If this condition is met, the set $\mathcal{V}_G$ is updated accordingly [line (22), Alg. 1].

Finally, given a new node u_{new}, we check if the cost of the nodes collected in the set (10) can decrease by rewiring them to v_{new}, as long as such transitions respect (4e) (or equivalently (9)) [lines 23-33, Alg. 1].

CHECKCONSTRREW: Note that the constraint in (9) is checked for the set of distributions $\mathcal{B}_{\epsilon(A, v_{\text{new}}.t)}(\hat{X})$ when a node $v_{\text{new}} \in \mathcal{V}$ is added to the tree. If now a node v_{near} is attempted to be rewired [line 26, Alg. 1], it may happen that $v_{\text{near}}.t$ changes so that we need to re-check (9) for v_{near} as well as for all leaf nodes of v_{near}. For a time stamp $t \in \mathbb{N}_{\geq 0}$, the function CHECKCONSTRREW(v_{near}, t) performs this step. Particularly, let $v_{\text{near}}, v_1, \ldots, v_L$ be the sequence of nodes that defines the tree starting from node v_{near} and ending in the leaf node v_L. Let t be the new time stamp of v_{near}. In order to check (9) for these nodes, we change the time stamps of these nodes to $v_{\text{near}}.t = t$ and $v_l.t = t + l$ for $l \in \{1, \ldots, L\}$. We then check if CHECKCONSTRROB(v_{near}) and CHECKCONSTRROB(v_l) hold for all $l \in \{1, \ldots, L\}$.

TIMESTAMPREW: In the case that a node $v_{\text{near}} \in \mathcal{V}$ is rewired, we need to update the time stamps of v_{near} and all its leaf nodes that we denote by $v_1, \ldots, v_L$. For a node v_{near} and a time stamp $t \in \mathbb{N}_{\geq 0}$, the function TIMESTAMPREW$(\mathcal{V}, v_{\text{near}}, t)$ changes the time stamps of these nodes to $v_{\text{near}} = t$ and $v_l.t = t + l$ for $l \in \{1, \ldots, L\}$ and outputs the modified set of nodes $\mathcal{V}$.

C. Real-time Execution and Replanning

Having constructed a tree by means of Algorithm 1, we can now find a control sequence U_H from $\mathcal{V}_G \subseteq \mathcal{V}$ as described in Section IV-A. This sequence will satisfy the constraints of the optimization problem (4) as will be formally shown in Section V. Towards solving the optimization problem (3), we propose an online execution and replanning scheme in Algorithm 2. A sequence U_H is initially calculated in lines 2-3 and executed in lines 4-10. Recall that the proposed R-RRT* in Algorithm 1 does not make use of Y_t and instead introduces robustness margins ϵ. There is hence a fundamental trade-off between the size of ϵ and the number of times $X(t|t) \notin \mathcal{B}_{\epsilon(A,t)}(\hat{X}(t|t))$. In these cases, we trigger replanning in lines (7)-(9).

Algorithm 1 Robust RRT*: Tree Expansion

1: **procedure** $(\mathcal{V}, \mathcal{V}_G, \mathcal{E})$=RRT*$(\mathcal{V}_0, \mathcal{E}_0, N)$
2: $\quad \mathcal{V} \leftarrow \mathcal{V}_0$, $\mathcal{V}_G \leftarrow \emptyset$, $\mathcal{E} \leftarrow \mathcal{E}_0$
3: $\quad$ **for** $i = 1, \ldots, N$ **do**
4: $\quad\quad s_{\text{rand}} \leftarrow$ SAMPLE(i)
5: $\quad\quad v_{\text{nearest}} \leftarrow$ NEAREST$(\mathcal{V}, \mathcal{E}, s_{\text{rand}})$
6: $\quad\quad (v_{\text{new}}, u_{\text{new}}) \leftarrow$ STEER$(v_{\text{nearest}}, s_{\text{rand}}, \eta)$
7: $\quad\quad$ **if** CHECKCONSTRROB(v_{new}) **then**
8: $\quad\quad\quad s_{\text{new}} = (v_{\text{new}}.\mu, v_{\text{new}}.\Sigma)$
9: $\quad\quad\quad \mathcal{V}_{\text{near}} \leftarrow$ NEAR$(\mathcal{V}, \mathcal{E}, s_{\text{new}}, \eta)$
10: $\quad\quad\quad v_{\text{min}} \leftarrow v_{\text{nearest}}$
11: $\quad\quad\quad c_{\text{min}} \leftarrow$ COST$(v_{\text{nearest}}) + c(v_{\text{new}}.\mu)$
12: $\quad\quad\quad$ **for** $v_{\text{near}} \in V_{\text{near}}$ **do**
13: $\quad\quad\quad\quad (v'_{\text{new}}, u'_{\text{new}}) \leftarrow$ STEER$(v_{\text{near}}, s_{\text{rand}}, \eta)$
14: $\quad\quad\quad\quad$ **if** $\wedge$COST$(v_{\text{near}}) + c(v'_{\text{new}}.\hat{\mu}) < c_{\text{min}}$ **then**
15: $\quad\quad\quad\quad\quad v_{\text{min}} \leftarrow v_{\text{near}}$
16: $\quad\quad\quad\quad\quad c_{\text{min}} \leftarrow$ COST$(v_{\text{near}}) + c(v'_{\text{new}}.\hat{\mu})$
17: $\quad\quad\quad\quad\quad v_{\text{new}} \leftarrow v'_{\text{new}}$
18: $\quad\quad\quad\quad\quad u_{\text{new}} \leftarrow u'_{\text{new}}$
19: $\quad\quad\quad \mathcal{V} \leftarrow \mathcal{V} \cup \{v_{\text{new}}\}$
20: $\quad\quad\quad \mathcal{E} \leftarrow \mathcal{E} \cup \{(v_{\text{min}}, v_{\text{new}}), u_{\text{min}}\}$
21: $\quad\quad\quad$ **if** GOALREACHED(v_{new}) **then**
22: $\quad\quad\quad\quad \mathcal{V}_G \leftarrow \mathcal{V}_G \cup \{v_{\text{new}}\}$
23: $\quad\quad\quad$ **for** $v_{\text{near}} \in V_{\text{near}} \setminus \{v_{\text{new}}\}$ **do**
24: $\quad\quad\quad\quad s_{\text{near}} \leftarrow (v_{\text{near}}.\mu, v_{\text{near}}.\Sigma)$
25: $\quad\quad\quad\quad (v'_{\text{new}}, u'_{\text{new}}) \leftarrow$ STEER$(v_{\text{new}}, s_{\text{near}}, \eta)$
26: $\quad\quad\quad\quad$ **if** $v_{\text{near}}.\mu = v'_{\text{new}}.\mu \wedge v_{\text{near}}.\Sigma = v'_{\text{new}}.\Sigma$
$\hookrightarrow \wedge$COST$(v_{\text{new}}) + c(v'_{\text{new}}.\hat{\mu}) <$ COST(v_{near})
$\hookrightarrow \wedge$CHECKCONSTRREW$(v_{\text{near}}, v'_{\text{new}}.t)$ **then**
27: $\quad\quad\quad\quad\quad \mathcal{V} \leftarrow$ TIMESTAMPREW$(\mathcal{V}, v_{\text{near}}, v'_{\text{new}}.t)$
28: $\quad\quad\quad\quad\quad e_{\text{parent}} \leftarrow$ PARENT(v_{near})
29: $\quad\quad\quad\quad\quad \mathcal{V} \leftarrow (\mathcal{V} \setminus \{v_{\text{near}}\}) \cup \{v'_{\text{new}}\}$
30: $\quad\quad\quad\quad\quad \mathcal{V}_G \leftarrow \mathcal{V}_G \setminus \{v_{\text{near}}\}$
31: $\quad\quad\quad\quad\quad$ **if** GOALREACHED(v'_{new}) **then**
32: $\quad\quad\quad\quad\quad\quad \mathcal{V}_G \leftarrow \mathcal{V}_G \cup \{v'_{\text{new}}\}$
33: $\quad\quad\quad\quad\quad \mathcal{E} \leftarrow (\mathcal{E} \setminus \{e_{\text{parent}}\}) \cup \{(v_{\text{new}}, v'_{\text{new}}), u'_{\text{new}}\}$

V. THEORETICAL GUARANTEES OF R-RRT*

Let us first show soundness of our proposed method with respect to the constraints (4d) and (4e).

Theorem 5.1 (Constraint Satisfaction of (4)*):* Let the tree $(\mathcal{V}, \mathcal{V}_G, \mathcal{E}) = $ RRT*$(v_0, \emptyset, N)$ be obtained from

Algorithm 2 Robust RRT*: Real-time Execution

```
t ← 0
(𝒱, 𝒱_G, ℰ) ← RRT*(v_0, ∅, N)
Find control sequence U_H from 𝒱_G ⊆ 𝒱
while R(‖X(H|H) − O_1‖ − δ) > γ) do
    Collect measurement y(t)
    Apply u(t)
    if X(t|t) ∉ ℬ_{ε(A,t)}(X̂(t|t)) then
        (𝒱, 𝒱_G, ℰ) ← RRT*(v_{t+1}, ∅, N)
        Find control sequence U_H from 𝒱_G ⊆ 𝒱
    t ← t + 1
```

Algorithm 1 for some $N \in \mathbb{N}_{\geq 0}$. Let $v_0, \ldots, v_H$ be a path in $(\mathcal{V}, \mathcal{E})$ with $v_H \in \mathcal{V}_G$ and let U_H be the associated control sequence, i.e., $(v_t, v_{t+1}, u(t)) \in \mathcal{E}$ for all $t \in \{0, \ldots, H-1\}$. Then it holds that the constraints (4d) and (4e) are satisfied.

Proof: Note first that every node v_t is such that

$$\sup_{X \in \mathcal{B}_{\epsilon(A, v_t.t)}(\hat{X}_t)} R(-d_j(X, O_2, \ldots, O_M)) \leq \gamma_j, \forall j \in \mathcal{J} \tag{11}$$

where $\hat{X}_t \sim \mathcal{N}(v_t.\mu, v_t.\Sigma)$. This follows because every node v_{new} that is added to the tree in line 19 of Algorithm 1 is checked for (11) via CHECKCONSTRROB in line 7. Also note that the function CHECKCONSTRREW in line 26 ensures that after rewiring each node still ensures (11). Since the STEER function, which is called in lines 6, 13, and 25 of Algorithm 1, ensures that the constraints (6), (7), and (8) hold, it consequently follows that (4d) and (4e) are satisfied. ■

Let us next show that Algorithm 1 has the property that the cost of each node decreases as we keep growing the tree.

Theorem 5.2 (Non-increasing Cost Function (4a)*):* Let the tree $(\mathcal{V}, \mathcal{V}_G, \mathcal{E}) = RRT^*(v_0, \emptyset, N)$ be obtained from Algorithm 1 for some $N \in \mathbb{N}_{\geq 0}$. If we extend this tree by calling $(\mathcal{V}', \mathcal{E}') = \text{RRT}^*(\mathcal{V}, \mathcal{E}, 1)$, then it holds that $\text{COST}(v') \leq \text{COST}(v)$ for each $v \in \mathcal{V}$ and $v' \in \mathcal{V}'$ with $v.\mu = v'.\mu$ and $v.\Sigma = v'.\Sigma$. *Proof:* The proof follows by construction of the rewiring in Algorithm 1 as nodes are rewired only if their cost decreases after rewiring as per line [26, Alg. 1]. ■

Remark 1 (Optimality): Note that proving global asymptotic optimality guarantees, as in case of the standard RRT* [2], is an open problem as our tree and the rewiring depend on time.

Let us next analyze in what way our solution to the optimization problem (4) relates to solving the optimization problem (3). Let us first state a straightforward corollary.

Corollary 5.3 (Constraint Satisfaction of (3)*):* Let the tree $(\mathcal{V}, \mathcal{V}_G, \mathcal{E}) = \text{RRT}^*(v_0, \emptyset, N)$ be obtained from Algorithm 1 for some $N \in \mathbb{N}_{\geq 0}$. Let $v_0, \ldots, v_H$ be a path in $(\mathcal{V}, \mathcal{E})$ with $v_H \in \mathcal{V}_G$ and let U_H be the associated control sequence, i.e., $(v_t, v_{t+1}, u(t)) \in \mathcal{E}$ for all $t \in \{0, \ldots, H-1\}$. Let $\hat{X}(t|t) \sim \mathcal{N}(\mu(t|0), \Sigma(t|t))$ be a random variable and let the realized disturbance $y(t)$ be such that

$$X(t|t) \in \mathcal{B}_{\epsilon(A,t)}(\hat{X}(t|t))$$

for all $t \in \{1, \ldots, H\}$. Then it holds that (3d) and (3e) hold.
Proof: Follows as a consequence of Theorem 5.1. ■

By the previous result, note that larger ϵ will increase the probability of satisfying the constraints of (3) as it is more likely to satisfy $X(t|t) \in \mathcal{B}_{\epsilon(A,t)}(\hat{X}(t|t))$. For a trajectory of length $H \in \mathbb{N}_{\geq 0}$, let us next quantify the probability $\delta \in [0,1]$ such that $X(t|t) \in \mathcal{B}_{\epsilon(A,t)}(\hat{X}(t|t))$ for all $t \in \{1, \ldots, H\}$. Recall that $X(t|t) \in \mathcal{N}(\mu(t|t), \Sigma(t|t))$. From (2), note that $\mu(t|t)$ is a random variable with a mean and a covariance when $y(t)$ is treated instead as a random variable $Y(t)$. Let us denote this difference by explicitly using $M(t|t)$ instead of $\mu(t|t)$ and note that $M(t|t)$ is such that

$$M(t+1|t+1) = F_\mu(M(t|t), u(t), Y(t+1))$$
$$M(t+1|t+1) = F_\mu(M(t|t), u(t), CX(t+1) + V(t+1))$$

Note that $M(t|t)$ is linear in $X(t)$ and $V(t)$, which both follow a Gaussian distribution, so that $M(t|t)$ again follows a Gaussian distribution. We can then calculate

$$\begin{aligned} \delta &:= P(X(t|t) \in \mathcal{B}_{\epsilon(A,t)}(\hat{X}(t|t)), t \in \{1, \ldots, H\}) \\ &= P(\|M(t|t) - \mu(t|0)\|^2 \leq \epsilon(A,t), t \in \{1, \ldots, H\}) \end{aligned}$$

as the probability that $X(t|t) \in \mathcal{B}_{\epsilon(A,t)}(\hat{X}(t|t))$.

Corollary 5.4: Let the tree $(\mathcal{V}, \mathcal{V}_G, \mathcal{E}) = RRT^*(v_0, \emptyset, N)$ be obtained from Algorithm 1 for some $N \in \mathbb{N}_{\geq 0}$. Let $v_0, \ldots, v_H$ be a path in $(\mathcal{V}, \mathcal{E})$ with $v_H \in \mathcal{V}_G$ and let U_H be the associated control sequence, i.e., $(v_t, v_{t+1}, u(t)) \in \mathcal{E}$ for all $t \in \{0, \ldots, H-1\}$. Then with a probability δ the constraints (3d) and (3e) hold. *Proof:* Follows as a consequence of Theorem 5.1 and Corollary 5.3. ■

Remark 5.5: Our proposed method hence allows to make statements such as "with a probability of δ, the constraints (3d) and (3e) will hold". The probability δ naturally increases with the size of ϵ, which increases conservatism.

Next, we analyze the effect of ϵ and provide some insights on how to select ϵ. Given two nodes $v, v' \in \mathcal{V}$, recall that the node v encodes the set of distributions $X \in \mathcal{B}_{\epsilon(A,v.t)}(v.\mu, v.\Sigma)$. If there exists an edge between v and v', i.e., $(v, v', u) \in \mathcal{E}$, a favorable property would be that there exists a dynamically feasible transition from each distribution in $\mathcal{B}_{\epsilon(A,v.t)}(v.\mu, v.\Sigma)$ into $\mathcal{B}_{\epsilon(A,v'.t)}(v'.\mu, v'.\Sigma)$ when applying the same u. This means that for each $X \in \mathcal{B}_{\epsilon(A,v.t)}(v.\mu, v.\Sigma)$, it has to hold that

$$\|AE(X) + Bu + \mu_W - v'.\mu\|^2 \leq \epsilon(A, v'.t) \tag{12}$$

where $E(X)$ is the expected value of X. We next show under which conditions (12) holds.

Theorem 5.6: Let $\epsilon(A,t) = \|\|A\|\|^t \zeta$ for some $\zeta > 0$. Then for a transition $(v, v', u) \in \mathcal{E}$, there exists a dynamically feasible transition from each distribution in $\mathcal{B}_{\epsilon(A,v.t)}(v.\mu, v.\Sigma)$ into $\mathcal{B}_{\epsilon(A,v'.t)}(v'.\mu, v'.\Sigma)$, i.e., (12) holds.
Proof: Note that the control input u is such that the transition $(v.\mu, v.\Sigma)$ into $(v'.\mu, v'.\Sigma)$ is dynamically feasible, i.e., such that $v'.\mu = Av.\mu + Bu + \mu_W$. Now, for any $(\mu, \Sigma) \in$

$\mathcal{B}_{\epsilon(A,v.t)}(v.\mu, v.\Sigma)$ let $\mu' := A\mu + Bu + \mu_W$. We now have that $\|v'.\mu - \mu'\| = \|A(v.\mu - \mu)\| \leq \|\|A\|\| \|v.\mu - \mu\| \leq \|\|A\|\| \epsilon(A, v.t) = \|\|A\|\| \|\|A\|\|^{v.t}\zeta = \|\|A\|\|^{v'.t}\zeta = \epsilon(A, v'.t)$. ■

VI. Simulation Studies

We next demonstrate our proposed robust RRT* algorithm. In Section VI-A, we define the stochastic system dynamics as per (1) that we will use throughout this section along with the environment in which the system operates. In Section VI-B, we illustrate the effect of the robustness parameter ϵ on the path design. Specifically, we illustrate the aforementioned trade-off between robustness and conservatism, and we show that as ϵ decreases, the re-planning frequency decreases too (see Alg. 2). In this way, we illustrate how ϵ can be used as a design parameter to do planning in between very optimistic and very conservative planning.

A. System Dynamics & Environment

Throughout this section, we consider a stochastic system as in (1) defined by the following matrices

$$A = \mathcal{I}_2 \otimes \begin{bmatrix} 1 & 0.5 \\ 0 & 1 \end{bmatrix} \; B = \mathcal{I}_2 \otimes \begin{bmatrix} 0.125 \\ 0.5 \end{bmatrix} \; C = \mathcal{I}_2 \otimes \begin{bmatrix} 1 & 0 \end{bmatrix}$$

where $\mathcal{I}_2 \in \mathbb{R}^{2\times 2}$ is the identity matrix and $\otimes$ is the Kronecker product. The state $x := (x_1, v_1, x_2, v_2)$ consists of position and velocity in the first and second coordinate. This system describes discretized two-dimensional double integrator dynamics with a sampling time of 0.5 s, e.g., a service robot navigating through an obstacle cluttered environment. The process and measurement noise are such that $W(t) \sim \mathcal{N}\Big(\mathcal{O}_4, \mathcal{I}_2 \otimes \begin{bmatrix} 0.5 & 0.1 \\ 0. & 0.5 \end{bmatrix}\Big)$ and $V(t) \sim \mathcal{N}\big(\mathcal{O}_2, \mathcal{I}_2 \otimes 0.4\big)$ where $\mathcal{O}_i$ denotes an i-dimensional vector containing zeros. The robot operates in the environment shown in Figure 1a, where the initial and goal location of the robot are $X(0) := (0, 0, 0, 0)$ and $O_1 = (0, 0, 0, 30)$, respectively. Observe also that there are three corridors for the robot to traverse through as defined by the obstacles $O_2, \ldots, O_7$ that are indicated by the black circles. Importantly, note that these corridors have different width. We consider $J = 6$ and let each function be $d_j(X, O_{j+1}) := \|X - O_{j+1}\|$ for each $j \in \{1, \ldots, 6\}$. Furthermore, we select $\gamma = \gamma_j = -0.5$ and $\kappa := 0.5$ and use the conditional value-at-risk as the risk measure R.

B. Effect of Robustness on Path Design

First, let us remark that a comparison with a version of a stochastic RRT* that does not incorporate measurements, such as for instance in [11], was not possible as the planning problem did not find a feasible solution after $N := 700$ iterations of sampling new nodes. The reason here is that the unconditional covariance matrix $\Sigma(t)$, which is used for planning, grows unbounded.

For our proposed R-RRT*, we select $N := 700$ and first consider constant ϵ of different sizes. In Figs. 1a-1d we show the result when no replanning is considered, i.e., Algorithm 2 is run without lines 7-9 so that the open-loop policy is executed and for $\epsilon = 0, 0.5, 3.5, 5.5$. It can be observed that increasing ϵ naturally results in selecting the the path that allows safer distance to the obstacles at the expense of having a larger cost over the planned path. In Figs. 1e and 1f, Algorithm 2 is run with replanning as indicated by the red crosses. It can be observed that smaller ϵ result in more frequent replanning. In Fig. 2a, we show the grown trees for $N := 700$. Finally, in Figs. 2b-2c we used time-varying epsilon, i.e, $\epsilon(A, t)$.

VII. Conclusions and Future Work

We considered the robust motion planning problem in the presence of state uncertainty. In particular, we proposed a novel sampling-based approach that introduces robustness margins into the offline planning to account for uncertainty in the state estimates based on a Kalman filter. We complement the robust offline planning with an online replanning scheme and show an inherent trade-off in the size of the robustness margin and the frequency of replanning.

References

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

[2] S. Karaman and E. Frazzoli, "Optimal kinodynamic motion planning using incremental sampling-based methods," in *49th IEEE Conference on Decision and Control (CDC)*, Atlanta, GA, 2010, pp. 7681–7687.

[3] ——, "Sampling-based algorithms for optimal motion planning," *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.

[4] H. Kurniawati, T. Bandyopadhyay, and N. M. Patrikalakis, "Global motion planning under uncertain motion, sensing, and environment map," *Autonomous Robots*, vol. 33, no. 3, pp. 255–272, 2012.

[5] A.-A. Agha-Mohammadi, S. Chakravorty, and N. M. Amato, "Firm: Sampling-based feedback motion-planning under motion uncertainty and imperfect measurements," *The International Journal of Robotics Research*, vol. 33, no. 2, pp. 268–304, 2014.

[6] C.-I. Vasile, K. Leahy, E. Cristofalo, A. Jones, M. Schwager, and C. Belta, "Control in belief space with temporal logic specifications," in *IEEE 55th Conference on Decision and Control (CDC)*, Las Vegas, NV, December 2016, pp. 7419–7424.

[7] K. Leahy, E. Cristofalo, C.-I. Vasile, A. Jones, E. Montijano, M. Schwager, and C. Belta, "Control in belief space with temporal logic specifications using vision-based localization," *The International Journal of Robotics Research*, vol. 38, no. 6, pp. 702–722, 2019.

[8] P. Cai, Y. Luo, A. Saxena, D. Hsu, and W. S. Lee, "Lets-drive: Driving in a crowd by learning from tree search," *Robotics: Science and Systems*, 2019.

[9] K. Sun, B. Schlotfeldt, G. J. Pappas, and V. Kumar, "Stochastic motion planning under partial observability for mobile robots with continuous range measurements," *IEEE Transactions on Robotics*, 2020.

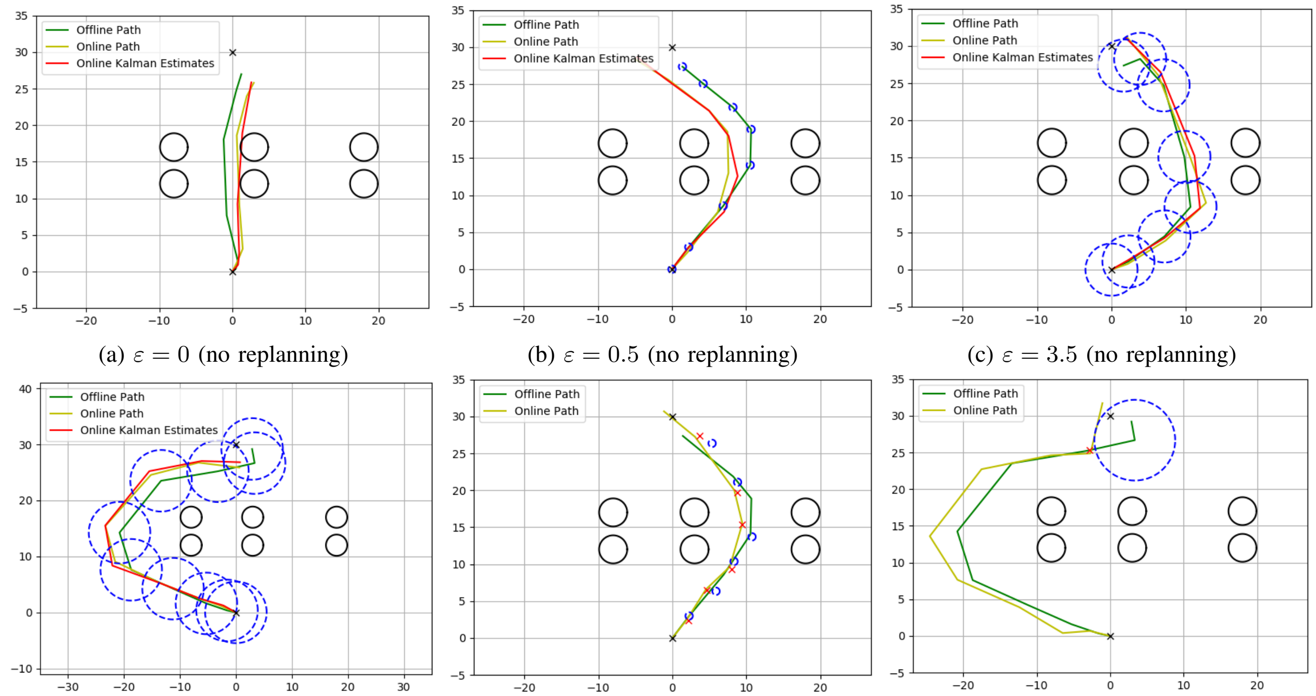
[10] N. Atanasov, J. Le Ny, K. Daniilidis, and G. J. Pappas, "Information acquisition with sensing robots: Algorithms and error bounds," in *IEEE International Conference on Robotics and Automation*, Hong Kong, China, 2014, pp. 6447–6454. [Online]. Available: https://ieeexplore.ieee.org/document/6907811

[11] B. Luders, M. Kothari, and J. How, "Chance constrained rrt for probabilistic robustness to environmental uncertainty," in *AIAA guidance, navigation, and control conference*, 2010, p. 8160.

[12] B. D. Luders, S. Karaman, and J. P. How, "Robust sampling-based motion planning with asymptotic optimality guarantees," in *AIAA Guidance, Navigation, and Control (GNC) Conference*, 2013, p. 5097.

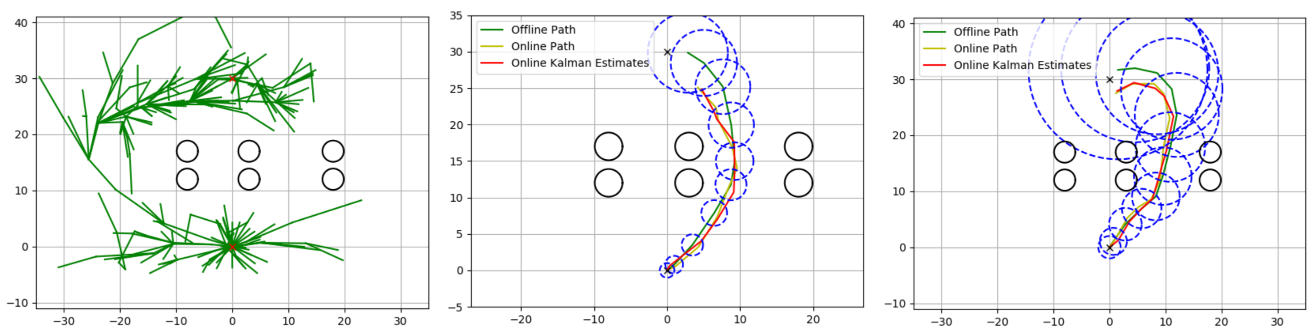
[13] M. Kothari and I. Postlethwaite, "A probabilistically robust path planning algorithm for uavs using rapidly-exploring random trees," *Journal of Intelligent & Robotic Systems*, vol. 71, no. 2, pp. 231–253, 2013.

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



(a) $\varepsilon = 0$ (no replanning)
(b) $\varepsilon = 0.5$ (no replanning)
(c) $\varepsilon = 3.5$ (no replanning)
(d) $\varepsilon = 5.5$ (no replanning)
(e) $\varepsilon = 0$ (with replanning)
(f) $\varepsilon = 5.5$ (with replanning)

Fig. 1: Simulation results with constant ϵ. (a)-(d) show the open-loop policy, while (e)-(f) show the replanning as in 2.The blue dashed circles indicate the robustness balls around the offline paths. The blue dashed circles indicate the epsilon balls of the open loop trajectory. The green line is unconditional path , i.e., offline as obtained from Algorithm 1, the red line is the Kalman estimate observed during executing 2, and the yellow line is the realized path of the robot.



(a) $\varepsilon = 5.5$
(b) $\epsilon(A, t) = \|A\|^t \zeta$ and $\zeta = 1$
(c) $\epsilon(A, t) = \|A\|^t \zeta$ and $\zeta = 2$

Fig. 2: (a) shows a grown tree of states for a robustness $\epsilon = 5.5$, while (b)-(c) show results for time-varying $\epsilon(A, t)$.

[15] B. D. Luders, S. Karaman, E. Frazzoli, and J. P. How, "Bounds on tracking error using closed-loop rapidly-exploring random trees," in *Proceedings of the 2010 American Control Conference*. IEEE, 2010, pp. 5406–5412.

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

[17] S. Safaoui, B. J. Gravell, V. Renganathan, and T. H. Summers, "Risk-averse rrt* planning with nonlinear steering and tracking controllers for nonlinear robotic systems under uncertainty," *arXiv preprint arXiv:2103.05572*, 2021.

[18] M. Schuurmans, A. Katriniok, H. E. Tseng, and P. Patrinos, "Learning-based risk-averse model predictive control for adaptive cruise control with stochastic driver models," *arXiv preprint arXiv:2005.02646*, 2020.

[19] J. Coulson, J. Lygeros, and F. Dörfler, "Distributionally robust chance constrained data-enabled predictive control," *arXiv preprint arXiv:2006.01702*, 2020.

[20] J. Van Den Berg, P. Abbeel, and K. Goldberg, "Lqg-mp: Optimized path planning for robots with motion uncertainty and imperfect state information," *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 895–913, 2011.

[21] V. Renganathan, I. Shames, and T. H. Summers, "Towards integrated perception and motion planning with distributionally robust risk constraints," *arXiv preprint arXiv:2002.02928*, 2020.

[22] K. Berntorp and S. Di Cairano, "Particle filtering for online motion planning with task specifications," in *2016 American Control Conference (ACC)*. IEEE, 2016, pp. 2123–2128.

[23] A. Bry and N. Roy, "Rapidly-exploring random belief trees for motion planning under uncertainty," in *IEEE International Conference on Robotics and Automation*, 2011, pp. 723–730.

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

APPENDIX I
RISK MEASURES

We next present some desireable properties that a risk measure may have. Let therefore $Z, Z' \in \mathfrak{F}(\Omega, \mathbb{R})$ be cost random variables. A risk measure is *coherent* if the following four properties are satisfied.

1. Monotonicity: If $Z(\omega) \leq Z'(\omega)$ for all $\omega \in \Omega$, it holds that $R(Z) \leq R(Z')$.

2. Translation Invariance: Let $c \in \mathbb{R}$. It holds that $R(Z + c) = R(Z) + c$.

3. Positive Homogeneity: Let $c \in \mathbb{R}_{\geq 0}$. It holds that $R(cZ) = R(Z)$.

4. Subadditivity: It holds that $R(Z + Z') \leq R(Z) + R(Z')$.

If the risk measure additionally satisfies the following two properties, then it is called a distortion risk measure.

5. Comonotone Additivity: If $(Z(\omega) - Z(\omega'))(Z'(\omega) - Z'(\omega')) \geq 0$ for all $\omega, \omega' \in \Omega$ (namely, Z and Z' are commotone), it holds that $R(Z + Z') = R(Z) + R(Z')$.

6. Law Invariance: If Z and Z' are identically distributed, then $R(Z) = R(Z')$.

Common examples of popular risk measures are the expected value $\mathrm{E}(Z)$ (risk neutral) and the worst-case $\operatorname{ess\,sup}_{\omega \in \Omega} Z(\omega)$ as well as:

- Mean-Variance: $\mathrm{E}(Z) + \lambda \mathrm{Var}(Z)$ where $c > 0$.
- Value at Risk (VaR) at level $\beta \in (0, 1)$: $VaR_\beta(Z) := \inf\{\alpha \in \mathbb{R} | F_Z(\alpha) \geq \beta\}$.
- Conditional Value at Risk (CVaR) at level $\beta \in (0, 1)$: $CVaR_\beta(Z) := E(Z | Z > VaR_\beta(Z))$.

Many risk measures are not coherent and can lead to a misjudgement of risk, e.g., the mean-variance is not monotone and the value at risk (which is closely related to chance constraints as often used in optimization) does not satisfy the subadditivity property.

APPENDIX II
STATE ESTIMATION

The random variable $X(t)$ of the stochastic control system in (1) is defined by the unconditional mean $\mu(t) := E[X(t)]$ and the unconditional covariance matrix $\Sigma(t) := E[(X(t) - \mu(t))(X(t) - \mu(t))^T]$ can recursively be calculated as

$$\begin{aligned}\mu(t+1) &= A\mu(t) + Bu(t) + \mu_W, \\ \Sigma(t+1) &= A\Sigma(t)A^T + \Sigma_W.\end{aligned}$$

The random variable $X(t|s)$ of the stochastic control system in (1) is defined by the conditional mean $\mu(t|s) := E[X(t)|Y_s] = E[X(t|s)]$ and the conditional covariance matrix $\Sigma(t|s) := E[(X(t|s) - \mu(t|s))(X(t|s) - \mu(t|s))^T]$. These can again recursively be calculated by means of the Kalman filter using the prediction equations

$$\begin{aligned}\mu(t+1|t) &:= A\mu(t|t) + Bu(t) + \mu_W \\ \Sigma(t+1|t) &:= A\Sigma(t|t)A^T + \Sigma_W\end{aligned}$$

and the update equations

$$\mu(t|t) = \mu(t|t-1) + K(t)(y(t) - C\mu(t|t-1) - \mu_V) \tag{13}$$

$$\Sigma(t|t) = \Sigma(t|t-1) - K(t)C\Sigma(t|t-1) \tag{14}$$

and the optimal Kalman gain

$$K(t) := \Sigma(t|t-1)C^T(C\Sigma(t|t-1)C^T + \Sigma_V)^{-1}.$$

The prediction and update equations together with the Kalman gain $K(t)$ define the functions F_μ and F_Σ.