2021 60th IEEE Conference on Decision and Control (CDC)
December 13-15, 2021. Austin, Texas

Robust Motion Planning in the Presence of Estimation Uncertainty

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

2021 60th IEEE Conference on Decision and Control (CDC) | 978-1-6654-3659-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/CDC45484.2021.9683651

***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 estimation 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 simulations, that the induced robustness margins constitute a trade-off between conservatism and robustness for planning under estimation 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 configuration starting from an initial configuration while avoiding unsafe states (e.g., obstacles). Several planning and control methods have been proposed to address this problem under the assumption that the unsafe state space is known; see e.g., [2], [3].

In this paper, we consider the problem of robust motion planning in the presence of state estimation uncertainty. In particular, we consider the case where the goal is to control a linear system using a Kalman filter to reach a desired final state while avoiding known unsafe states. To account for estimation uncertainty in current and predicted future system states due to noisy sensors and Kalman filtering, we require the system's state to always respect risk-aware safety constraints. First, we formulate this motion planning problem as a stochastic optimal control problem that generates control policies that, as a consequence of Kalman filtering, rely on future sensor measurements. Due to this dependence on future measurements, which are not available initially, 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 estimation uncertainty may evolve under the execution of these controllers and a Kalman filter and *robust* as they guarantee that risk-aware safety constraints are met even when the predicted state estimates deviate from the realized 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: Motion planning under estimation uncertainty has been considered in [4]–[6] by using partially observable markov decision processes in belief space. As such approaches are often computationally intractable, sampling-based approximations have appeared in [7]–[9]. The authors in [10], [11] 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 [12], [13]. Common in these works is that the planning is not integrated with sensing and, therefore, these approaches are quite conservative as uncertainty may grow unbounded. Sampling-based approaches that exhibit robustness to process noise have been proposed in [14], [15], but again without considering sensing information. The authors in [16] have used an unscented transformation to estimate state distributions for nonlinear systems within an RRT* framework by considering tightened risk constraints. The authors in [17], [18] 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 and sampling-based approaches that integrate Kalman or particle filters have been proposed in [19]–[22]. These works have in common to treat future output measurements as random variables via its output measurement map, i.e., the map from states to observations. As a consequence, the estimated state becomes a random variable. This way, these

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 AFOSR grant FA9550-19-1-0265 (Assured Autonomy in Contested Environments) and the NSF CPS 2038873.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:57:19 UTC from IEEE Xplore. Restrictions apply.

methods lack robustness to uncertainty in the realized output measurements. We instead impose robustness margins around a predicted trajectory under a Kalman filter and nominal output measurements, and replan if needed.

Contribution: *First*, we formulate the motion planning problem under state estimation uncertainty and risk constraints as a stochastic optimal control problem and propose a tractable approximation that is based on nominal predictions and robustified constraints. *Second*, we propose a sampling-based algorithm towards solving the approximate problem and 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 margins within the robustified constraints directly affect the frequency of replanning and hence constitute a fundamental trade-off between conservatism and robustness.

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 [23].

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 X where $\omega \in \Omega$. As X is measurable, a probability space can be defined for X and probabilities can be assigned to events for 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. Risk measures allow for a risk assessment when the input of R is seen as a cost random variable. Commonly used risk measures are the expected value, the variance, or the conditional value-at-risk [24]. In [23, Appendix A], we summarize existing risk measures and their properties.

B. Stochastic Control System

Consider the discrete-time stochastic control system

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

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 [23, Appendix B]. 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 $Y_t := \begin{bmatrix} y(0)^T & \dots & y(t)^T \end{bmatrix}^T$. 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

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

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 [23, Appendix B].

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 $i \in \mathcal{M} := \{1, \dots, M\}$. The goal is to safely navigate the system to the goal region O_1 by means of the control input $u(t)$. Additionally, the system has satisfy spatial requirements, i.e., to avoid and/or be close to regions $O_2, \dots, O_M$. For instance, consider a robot that always has to avoid obstacles and be within communication range of a static wifi spots. Consider therefore J measurable functions $d_j : \mathbb{R}^{nM} \to \mathbb{R}$ for $j \in \mathcal{J} := \{1, \dots, J\}$ 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 $d_j(X(t|t), O_2, \dots, O_M)$ also become random variables. The problem that this paper addresses can be captured by the following *stochastic* optimal

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:57:19 UTC from IEEE Xplore. Restrictions apply.

control problem:

$$\min_{H,U_H} \sum_{t=0}^{H} c(\mu(t|t)) \tag{3a}$$

$$\text{s.t. } \mu(t+1|t+1) = F_\mu(\mu(t|t), u(t), y(t+1)), \forall t \in \mathcal{T}_H \tag{3b}$$

$$\Sigma(t+1|t+1) = F_\Sigma(\Sigma(t|t)), \forall t \in \mathcal{T}_H \tag{3c}$$

$$R(-d_j(X(t|t), O_2, \dots, O_M)) \leq \gamma_j, \forall j \in \mathcal{J}, \forall t \in \mathcal{T}_H \tag{3d}$$

$$R(\|X(H|H) - O_1\| - \kappa) \leq \gamma, \tag{3e}$$

where $U_t := \begin{bmatrix} u(0)^T & \dots & u(t)^T \end{bmatrix}^T$, $\mathcal{T}_H = \{0, \dots, H\}$, $c : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ is a cost function, $\kappa > 0$ indicates the accuracy 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 (3b), (3d), and (3e) on the realized output measurements $y(t)$. Therefore, (3) can not be solved a priori.

Unlike [19]–[22] where $y(t)$ is treated as a random variable, consequently making $\mu(t|t)$ itself a random variable, we approximate the optimization problem (3) with an approximate stochastic optimal control problem by using $\mu(t|0)$ instead of $\mu(t|t)$. 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 $\mu(t)$. To account for a potential mismatch between $\mu(t|0)$ and the realization of $\mu(t|t)$, we introduce additional *robustness* margins to alleviate the lack of knowledge of Y_t during planning.

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.[1] To formalize the 'ϵ-closeness' notion, define the set of distributions

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

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.

We approximate the stochastic optimal control problem (3) with the stochastic optimal control problem

$$\min_{H,U_H} \sum_{t=0}^{H} c(\mu(t|0)) \tag{4a}$$

$$\text{s.t. } \mu(t+1|0) = A\mu(t|0) + Bu(t) + \mu_W, \forall t \in \mathcal{T}_H \tag{4b}$$

$$\Sigma(t+1|t+1) = F_\Sigma(\Sigma(t|t)), \forall t \in \mathcal{T}_H \tag{4c}$$

$$\sup_{X \in \mathcal{B}_{\epsilon(A,t)}(\hat{X}(t|t))} R(-d_j(X, O_2, \dots, O_M)) \leq \gamma_j, \quad \forall j \in \mathcal{J}, \forall t \in \mathcal{T}_H \tag{4d}$$

$$\sup_{X \in \mathcal{B}_{\epsilon(A,H)}(\hat{X}(H|H))} R(\|X - O_1\| - \kappa) \leq \gamma, \tag{4e}$$

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 partially accounted for 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 models of the system, the process, or the 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., [25].

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

We build upon the RRT* algorithm [3] and present a new robust sampling-based approach to solve (4). 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 when $X(t|t) \notin \mathcal{B}_\epsilon(\hat{X}(t|t))$ during the online execution of the algorithm.

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

[1] 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.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:57:19 UTC from IEEE Xplore. Restrictions apply.

$v.\Sigma$, respectively, along with a time stamp $v.t$. We remark that each node will accept state distributions X that are such that $X \in \mathcal{B}_\epsilon\big(\mathcal{N}(v.\mu, v.\Sigma)\big)$. 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 at time $v.t$ and by means of u according to the dynamics in (2). 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). The tree is rooted at a node v_0 capturing the initial system 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 an incremental fashion, as detailed in Algorithm 1, 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 (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 (4e). 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 $H := v.t$ 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 follows next.

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, we use $\text{SAMPLE}(i)$ instead of $\text{SAMPLE}(i, \omega)$ and again omit the underlying sample space Ω.

Remark 4.1 (Sampling): RRT* algorithms typically draw samples directly from the obstacle-free space. This is not possible here due to the stochastic setup so that nodes consist of means and a covariances. These nodes need to be dynamically feasible according to (2) with respect to the parent node while satisfying the constraint (4d). We sample both means and covariances despite having no control over the covariance update equation (2b). This follows as we try to connect new nodes to existing nodes $v \in \mathcal{V}$ that may be different time hops away from the root node v_0.

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:

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

where $d(\cdot)$ is the 2nd Wasserstein distance.[2]

STEER: In order to steer the tree from a node v towards s [lines 6, 13, and 25, Alg. 1], we use the function $\text{STEER}(v, s, \eta) := (v_{\text{new}}, u_{\text{new}})$, where $\eta \geq 0$ is a positive constant, and returns the closest to s dynamically feasible node $v_{\text{new}} := (\mu_{\text{new}}, \Sigma_{\text{new}}, v.t + 1)$ that is still η-close to the node v. In particular, the mean μ_{new} and the covariance Σ_{new} are required to be dynamically feasible according to the dynamics in (2). The steer function aims to solve the optimization problem

$$\begin{aligned}(\mu_{\text{new}}, \Sigma_{\text{new}}, u_{\text{new}}) := \underset{(\mu,\Sigma,u) \in \mathbb{R}^n \times S_{\geq 0} \times \mathbb{R}^m}{\text{argmin}}\ & d(\mathcal{N}(s), \mathcal{N}(\mu, \Sigma)) \\ \text{s.t. } & d(\mathcal{N}(v.\mu, v.\Sigma), \mathcal{N}(\mu, \Sigma)) \leq \eta \\ & \mu = Av.\mu + Bu + \mu_W && \text{(6a)} \\ & \Sigma = F_\Sigma(v.\Sigma) && \text{(6b)} \\ & u \in \mathcal{U}. && \text{(6c)}\end{aligned}$$

The above optimization program can be rewritten as a convex optimization program if $\mathcal{U}$ is a convex set. First, note that $\Sigma = F_\Sigma(v.\Sigma)$ is fixed according to constraint (6b). For this Σ, the expressions involving $d(\cdot)$ then result in expressions that are convex in μ due to the use of the 2nd Wasserstein distance. Finally, (6a) and (6c) are convex in u if $\mathcal{U}$ is convex.

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

$$\sup_{X \in \mathcal{B}_{\epsilon(A, v_{\text{new}}.t)}(\hat{X})} R(-d_j(X, O_2, \ldots, O_M)) \leq \gamma_j, \forall j \in \mathcal{J}. \tag{7}$$

where $\hat{X} \sim \mathcal{N}(v_{\text{new}}.\mu, v_{\text{new}}.\Sigma)$. Checking (7) analytically may in general be hard. Sampling-based solutions as in [24] for the the conditional value-at-risk (CVaR) or [26] for logic specifications can be used, while there also exist sampling-based reformulations, again for the CVaR in [18, Sec. IV].

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]

$$\begin{aligned}&\text{NEAR}(\mathcal{V}, \mathcal{E}, s_{\text{new}}, \eta) := \\ &\{v \in \mathcal{V} | \|\mathcal{N}(s_{\text{new}}) - \mathcal{N}(v.\mu, v.\Sigma)\| \leq r(\mathcal{V})\}.\end{aligned} \tag{8}$$

where $r(\mathcal{V})$ is, for $\gamma > 0$, defined as

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

The set in (8) collects all nodes $v \in \mathcal{V}$ that are within at most a distance of r from s_{new} in terms of the norm $\|\cdot\|$.

[2]Particularly, we define $d(\mathcal{N}(\mu_1, \Sigma_1), \mathcal{N}(\mu_2, \Sigma_2)) := \|\mu_1 - \mu_2\|^2 + \text{trace}(\Sigma_1 + \Sigma_2 - 2(\Sigma_2^{\frac{1}{2}} \Sigma_1 \Sigma_2^{\frac{1}{2}})^{\frac{1}{2}})$.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:57:19 UTC from IEEE Xplore. Restrictions apply.

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 (4e). This is accomplished by the function GOALREACHED [line (21), Alg. 1]. This function resembles the function CHECKCONSTRROB, but it focuses on the terminal constraint (4e) instead of (4d). 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:

$$\sup_{X \in \mathcal{B}_{\epsilon(A, v_{\text{new}}.t)}(\hat{X})} R(\|X - O_1\| - \kappa) \leq \gamma. \tag{9}$$

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 v_{new}, we check if the cost of the nodes collected in the set (8) can decrease by rewiring them to v_{new}, as long as such transitions respect (4d) (or equivalently (7)) [lines 23-33, Alg. 1].

CHECKCONSTRREW: Note that the constraint in (7) 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 (7) 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 (7) 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*$(\mathcal{V}_0, \mathcal{V}_{G,0}, \mathcal{E}_0, N)$
2: $\mathcal{V} \leftarrow \mathcal{V}_0$, $\mathcal{V}_G \leftarrow \mathcal{V}_{G,0}$, $\mathcal{E} \leftarrow \mathcal{E}_0$
3: **for** $i = 1, \ldots, N$ **do**
4: $s_{\text{rand}} \leftarrow$ SAMPLE(i)
5: $v_{\text{nearest}} \leftarrow$ NEAREST$(\mathcal{V}, \mathcal{E}, s_{\text{rand}})$
6: $(v_{\text{new}}, u_{\text{new}}) \leftarrow$ STEER$(v_{\text{nearest}}, s_{\text{rand}}, \eta)$
7: **if** CHECKCONSTRROB(v_{new}) **then**
8: $s_{\text{new}} = (v_{\text{new}}.\mu, v_{\text{new}}.\Sigma)$
9: $\mathcal{V}_{\text{near}} \leftarrow$ NEAR$(\mathcal{V}, \mathcal{E}, s_{\text{new}}, \eta)$
10: $v_{\min} \leftarrow v_{\text{nearest}}$
11: $c_{\min} \leftarrow$ COST$(v_{\text{nearest}}) + c(v_{\text{new}}.\mu)$
12: **for** $v_{\text{near}} \in V_{\text{near}}$ **do**
13: $(v'_{\text{new}}, u'_{\text{new}}) \leftarrow$ STEER$(v_{\text{near}}, s_{\text{rand}}, \eta)$
14: **if** COST$(v_{\text{near}}) + c(v'_{\text{new}}.\hat{\mu}) < c_{\min}$ **then**
15: $v_{\min} \leftarrow v_{\text{near}}$
16: $c_{\min} \leftarrow$ COST$(v_{\text{near}}) + c(v'_{\text{new}}.\hat{\mu})$
17: $v_{\text{new}} \leftarrow v'_{\text{new}}$
18: $u_{\text{new}} \leftarrow u'_{\text{new}}$
19: $\mathcal{V} \leftarrow \mathcal{V} \cup \{v_{\text{new}}\}$
20: $\mathcal{E} \leftarrow \mathcal{E} \cup \{(v_{\min}, v_{\text{new}}), u_{\min}\}$
21: **if** GOALREACHED(v_{new}) **then**
22: $\mathcal{V}_G \leftarrow \mathcal{V}_G \cup \{v_{\text{new}}\}$
23: **for** $v_{\text{near}} \in V_{\text{near}} \setminus \{v_{\text{new}}\}$ **do**
24: $s_{\text{near}} \leftarrow (v_{\text{near}}.\mu, v_{\text{near}}.\Sigma)$
25: $(v'_{\text{new}}, u'_{\text{new}}) \leftarrow$ STEER$(v_{\text{new}}, s_{\text{near}}, \eta)$
26: **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: $\mathcal{V} \leftarrow$ TIMESTAMPREW$(\mathcal{V}, v_{\text{near}}, v'_{\text{new}}.t)$
28: $e_{\text{parent}} \leftarrow$ PARENT(v_{near})
29: $\mathcal{V} \leftarrow (\mathcal{V} \setminus \{v_{\text{near}}\}) \cup \{v'_{\text{new}}\}$
30: $\mathcal{V}_G \leftarrow \mathcal{V}_G \setminus \{v_{\text{near}}\}$
31: **if** GOALREACHED(v'_{new}) **then**
32: $\mathcal{V}_G \leftarrow \mathcal{V}_G \cup \{v'_{\text{new}}\}$
33: $\mathcal{E} \leftarrow (\mathcal{E} \setminus \{e_{\text{parent}}\}) \cup \{(v_{\text{new}}, v'_{\text{new}}), u'_{\text{new}}\}$

V. THEORETICAL GUARANTEES OF R-RRT*

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

Theorem 5.1 (Constraint Satisfaction of (4)*):* Let the tree $(\mathcal{V}, \mathcal{V}_G, \mathcal{E}) = \text{RRT}^*(v_0, \emptyset, \emptyset, N)$ 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 it holds that the constraints (4b)-(4e) are satisfied.

Proof: Note first that every node v_t is such that

$$\sup_{X \in \mathcal{B}_{\epsilon(A, v_t.t)}(\hat{X}_t)} R(-d_j(X, O_2, \ldots, O_M)) \leq \gamma_j, \forall j \in \mathcal{J} \tag{10}$$

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:57:19 UTC from IEEE Xplore. Restrictions apply.

Algorithm 2 Robust RRT*: Real-time Execution

1: $t \leftarrow 0$
2: $(\mathcal{V}, \mathcal{V}_G, \mathcal{E}) \leftarrow \text{RRT}^*(v_0, \emptyset, \emptyset, N)$
3: Find control sequence U_H from $\mathcal{V}_G \subseteq \mathcal{V}$
4: **while** $R(\|X(t|t) - O_1\| - \kappa) > \gamma)$ **do**
5: Collect measurement $y(t)$
6: Apply $u(t)$
7: **if** $X(t|t) \notin \mathcal{B}_{\epsilon(A,t)}(\hat{X}(t|t))$ **then**
8: $(\mathcal{V}, \mathcal{V}_G, \mathcal{E}) \leftarrow \text{RRT}^*(v_{t+1}, \emptyset, \emptyset, N)$
9: Find control sequence U_H from $\mathcal{V}_G \subseteq \mathcal{V}$
10: $t \leftarrow t + 1$

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 (10) via CHECKCONSTRROB in line 7. Also note that the function CHECKCONSTRREW in line 26 ensures that after rewiring each node $v.t$ still ensures (10). Similarly, the node v_H satisfies the goal constraint

$$\sup_{X \in \mathcal{B}_{\epsilon(A,v_H.t)}(\hat{X}_H)} R(\|\hat{X} - O_1\| - \kappa) \leq \gamma \tag{11}$$

due to the function GOALREACHED in line 21. Since the STEER function, called in lines 6, 13, and 25 of Alg. 1, ensures that the constraints (6a), (6b), and (6c) hold, it consequently follows that (4b)-(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 $(\mathcal{V}, \mathcal{V}_G, \mathcal{E}) = RRT^*(v_0, \emptyset, \emptyset, N)$ be obtained from Algorithm 1 for some $N \in \mathbb{N}_{\geq 0}$. If we extend this tree by calling $(\mathcal{V}', \mathcal{V}'_G, \mathcal{E}') = \text{RRT}^*(\mathcal{V}, \mathcal{V}_G, \mathcal{E}, 1)$, 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, as in case of RRT* [2], is an open problem as our incremental tree construction depends 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 $(\mathcal{V}, \mathcal{V}_G, \mathcal{E}) = \text{RRT}^*(v_0, \emptyset, \emptyset, N)$ 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 by the proof of Theorem 5.1, and in particular due to the satisfaction of (10) and (11). ■

Note that larger ϵ will increase the probability of satisfying the constraints of (3) as it is more likely that $X(t|t) \in \mathcal{B}_{\epsilon(A,t)}(\hat{X}(t|t))$. 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 $Y(t)$. Denote this random variable by $M(t|t)$ and note that

$$\begin{aligned} M(t+1|t+1) &= F_\mu(M(t|t), u(t), Y(t+1)) \\ &= F_\mu(M(t|t), u(t), CX(t+1) + V(t+1)) \end{aligned}$$

We have 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 $(\mathcal{V}, \mathcal{V}_G, \mathcal{E}) = RRT^*(v_0, \emptyset, \emptyset, N)$ 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 by 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.

We now suggest how to potentially 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 desirable property is that there exists a dynamically feasible transition from each distribution $X \in \mathcal{B}_{\epsilon(A,v.t)}(v.\mu, v.\Sigma)$ such that $AE(X) + Bu + \mu_W \in \mathcal{B}_{\epsilon(A,v'.t)}(v'.\mu, v'.\Sigma)$ to not trigger replanning too frequently. Note that $E(X)$ is the expected value of X. Hence, for each $X \in \mathcal{B}_{\epsilon(A,v.t)}(v.\mu, v.\Sigma)$,

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

has to hold. We show conditions under which (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 $v'.\mu = Av.\mu + Bu + \mu_W$. For any $\mathcal{N}(\mu, \Sigma) \in \mathcal{B}_{\epsilon(A,v.t)}(v.\mu, v.\Sigma)$, 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

In Section VI-A, we define the stochastic system dynamics as per (1) along with the environment. In Section VI-B, we illustrate the effect of the robustness parameter ϵ. Specifically, we illustrate the trade-off between robustness and conservatism, its effect on the re-planning frequency in

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:57:19 UTC from IEEE Xplore. Restrictions apply.

Alg. 2, and hence how ϵ can be used as a design parameter in between very optimistic and very conservative planning.

A. System Dynamics & Environment

Consider the stochastic system as in (1) defined by

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

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 $W(t) \sim \mathcal{N}\Big(\mathcal{O}_4, \mathcal{I}_2 \otimes \begin{bmatrix} 0.5 & 0.1 \\ 0.1 & 0.5 \end{bmatrix}\Big)$ 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 $O_1 = (0, 0, 30, 0)$, 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 $J := 6$ risk constraints and let each function be $d_j(X, O_{j+1}) := \|X - O_{j+1}\|$ for $j \in \{1, \ldots, 6\}$. We select $\gamma := \gamma_j := -0.5$ and $\kappa := 0.5$ and use the conditional value-at-risk $R(\cdot) := CVaR_\beta(\cdot)$ at risk level $\beta := 0.95$. To check (7) and (9) for a node $v \in \mathcal{V}$, we sample 10 Gaussian distributions $X \in \mathcal{B}_{\epsilon(A,v.t)}(\hat{X})$ where we recall that $\hat{X} \sim \mathcal{N}(v.\mu, v.\Sigma)$. Then, we check for each X if $R(-d_j(X, O_2, \ldots, O_M)) \leq \gamma_j, \forall j \in \mathcal{J}$ and $R(\|X - O_1\| - \kappa) \leq \gamma$ hold, respectively, using 500 samples from X and a sample average approximation [24]. For ϵ of 0, 0.5, 3.5, and 5.5 (see next section), we observed average run times, i.e., until a satisfying path was found by Algorithm 1, of 114, 127, 309, and 288 s. We observed that checking (7) and (9) is computationally expensive, and remark that we plan to derive efficient reformulations in the future.

B. Effect of Robustness on Path Design

We select $N := 700$ and 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 for choices of $\epsilon = 0, 0.5, 3.5, 5.5$. It can be observed that increasing ϵ naturally results in selecting the path that allows safer distance to the obstacles at the expense of having a larger cost over the planned path. Note also that for $\epsilon = 0$ the robot collides with one of the obstacles. 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 in green for $\epsilon = 5.5$. In Figs. 2b-2c we used time-varying epsilon, i.e, $\epsilon(A, t)$ to account for growing estimation uncertainty as time increases.

Finally, let us remark that a comparison with a version of a stochastic RRT* that does not incorporate measurements, e.g., as in [10], was no feasible solution was found 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.

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. Future work includes integration of perception and feedback control.

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 *Proc. Conf. Decis. Control*, Atlanta, GA, Dec. 2010, pp. 7681–7687.

[3] ——, "Sampling-based algorithms for optimal motion planning," *Int. Journal Robot. 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] P. Cai, Y. Luo, A. Saxena, D. Hsu, and W. S. Lee, "Lets-drive: Driving in a crowd by learning from tree search," in *Proc. Robot.: Science Syst.*, Freiburg, Germany, June 2019.

[6] K. Sun, B. Schlotfeldt, G. J. Pappas, and V. Kumar, "Stochastic motion planning under partial observability for mobile robots with continuous range measurements," *IEEE Trans. Robot.*, vol. 37, no. 3, pp. 979–995, 2020.

[7] A.-A. Agha-Mohammadi, S. Chakravorty, and N. M. Amato, "FIRM: Sampling-based feedback motion-planning under motion uncertainty and imperfect measurements," *Int. Journal Robot. Research*, vol. 33, no. 2, pp. 268–304, 2014.

[8] C.-I. Vasile, K. Leahy, E. Cristofalo, A. Jones, M. Schwager, and C. Belta, "Control in belief space with temporal logic specifications," in *Proc. Conf. Decis Control*, Las Vegas, NV, Dec. 2016, pp. 7419–7424.

[9] 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," *Int. Journal Robot. Research*, vol. 38, no. 6, pp. 702–722, 2019.

[10] B. D. Luders, M. Kothari, and J. How, "Chance constrained RRT for probabilistic robustness to environmental uncertainty," in *Proc. Conf. AIAA guid., nav., control*, Toronto, Canada, Aug. 2010, p. 8160.

[11] B. D. Luders, S. Karaman, and J. P. How, "Robust sampling-based motion planning with asymptotic optimality guarantees," in *Proc. Conf. AIAA guid., nav., control*, Boston, MA, Aug. 2013, p. 5097.

[12] M. Kothari and I. Postlethwaite, "A probabilistically robust path planning algorithm for uavs using rapidly-exploring random trees," *Intel. & Robot. Syst.*, vol. 71, no. 2, pp. 231–253, 2013.

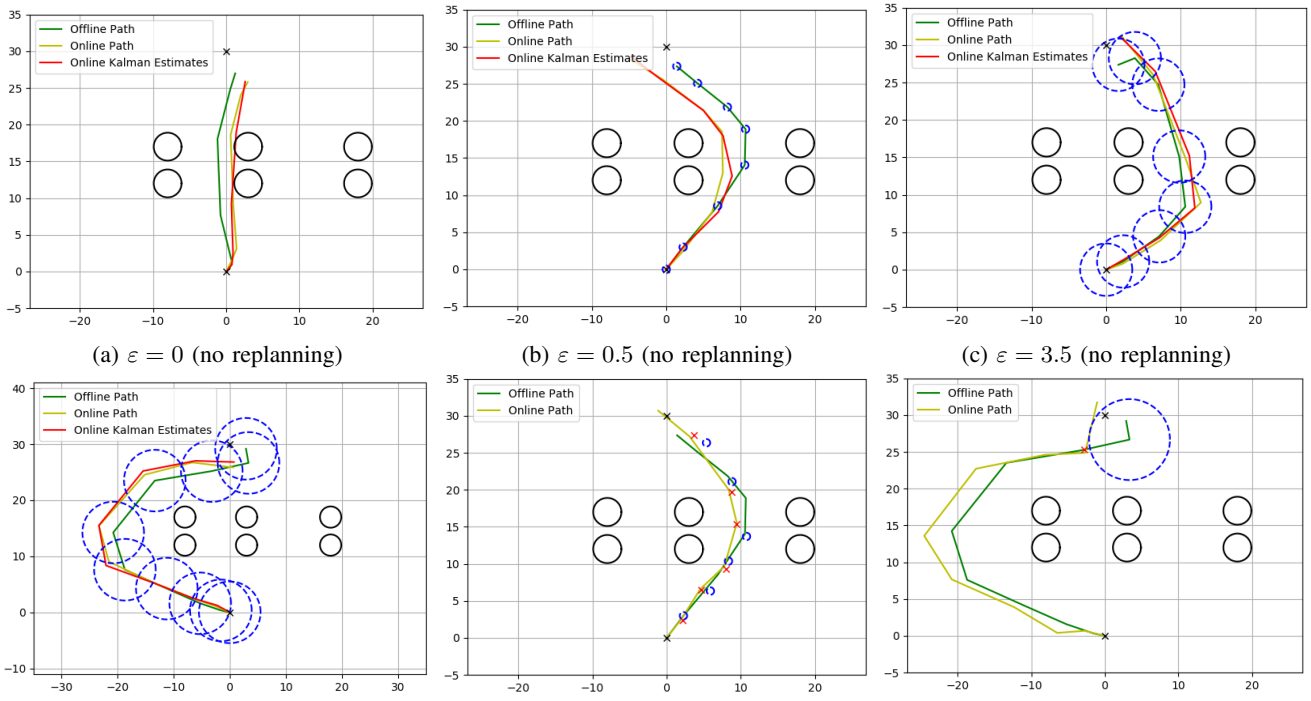
[13] G. S. Aoude, B. D. Luders, J. M. Joseph, N. Roy, and J. P. How, "Probabilistically safe motion planning to avoid dynamic obstacles with uncertain motion patterns," *Autonomous Robots*, vol. 35, no. 1, pp. 51–76, 2013.

[14] B. D. Luders, S. Karaman, E. Frazzoli, and J. P. How, "Bounds on tracking error using closed-loop rapidly-exploring random trees," in *Proc. Am. Control Conf.*, Baltimore, Maryland, June 2010, pp. 5406–5412.

[15] T. Summers, "Distributionally robust sampling-based motion planning under uncertainty," in *Proc. Conf. Intel. Robots Syst.*, Madrid, Spain, October 2018, pp. 6518–6523.

[16] 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.

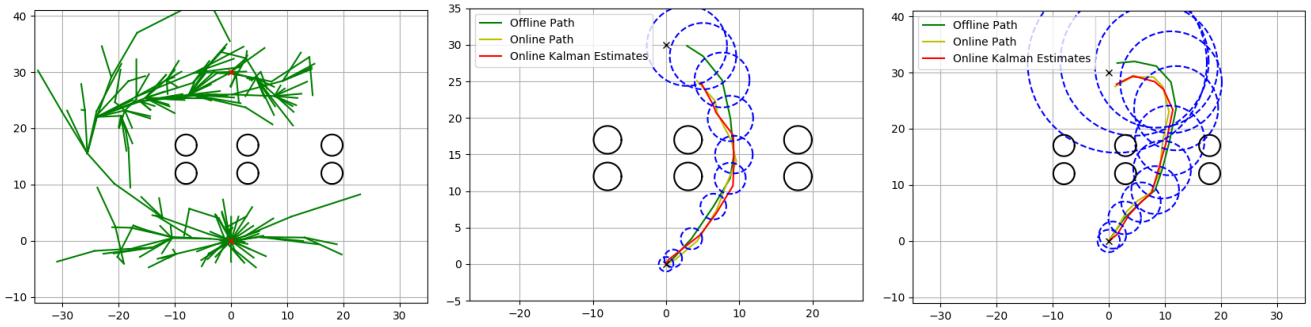
Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:57:19 UTC from IEEE Xplore. Restrictions apply.



(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 according to Algorithm 2. The blue dashed circles indicate the epsilon balls for some time steps around the offline paths. The green line is the nominal mean $\mu(t|0)$, i.e., the path as obtained from Algorithm 1, the red line is the realized Kalman estimate $\mu(t|t)$ observed during execution of Algorithm 2, and the yellow line is the realized path $x(t)$ 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 $\epsilon(A,t)$.

[17] 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," in *Proc. 21st IFAC World Congress*, Berlin, Germany, July 2020, pp. 15 128–15 133.

[18] J. Coulson, J. Lygeros, and F. Dörfler, "Distributionally robust chance constrained data-enabled predictive control," *IEEE Transactions on Automatic Control*, 2021.

[19] V. Renganathan, I. Shames, and T. H. Summers, "Towards integrated perception and motion planning with distributionally robust risk constraints," in *Proc. 21st IFAC World Congress*, Berlin, Germany, July 2020, pp. 15 530–15 536.

[20] K. Berntorp and S. Di Cairano, "Particle filtering for online motion planning with task specifications," in *Proc. Am. Control Conf.*, Boston, MA, July 2016, pp. 2123–2128.

[21] A. Bry and N. Roy, "Rapidly-exploring random belief trees for motion planning under uncertainty," in *Proc. Conf. Robot. Autom.*, Shanghai, China, May 2011, pp. 723–730.

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

[23] L. Lindemann, M. Cleaveland, Y. Kantaros, and G. J. Pappas, "Robust motion planning in the presence of estimation uncertainty," *arXiv preprint arXiv:2108.11983*, 2021.

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

[25] N. Atanasov, J. Le Ny, K. Daniilidis, and G. J. Pappas, "Information acquisition with sensing robots: Algorithms and error bounds," in *Proc. Conf. Robot. Autom.*, Hong Kong, China, May 2014, pp. 6447–6454.

[26] L. Lindemann, N. Matni, and G. J. Pappas, "STL robustness risk over discrete-time stochastic processes," *arXiv preprint arXiv:2104.01503*, 2021.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:57:19 UTC from IEEE Xplore. Restrictions apply.