

Risk bounded nonlinear robot motion planning with integrated perception & control ^{☆,☆☆}

Venkatraman Renganathan ^{a,*}, Sleiman Safaoui ^b, Aadi Kothari ^c, Benjamin Gravell ^b, Iman Shames ^d, Tyler Summers ^{b,*}

^a Department of Automatic Control, Lund University, Naturvetarvagen 18, Lund, 221 00, SE, Sweden

^b Eric Jonsson School of Engineering & Computer Science, The University of Texas at Dallas, 800 W Campbell Rd, Richardson, 75080, TX, USA

^c School of Engineering, Massachusetts Institute of Technology, Cambridge, 02139, MA, USA

^d CIICADA Lab, School of Engineering, The Australian National University, Acton, 0200, ACT, Australia

ARTICLE INFO

Article history:

Received 15 October 2021

Received in revised form 25 October 2022

Accepted 27 October 2022

Available online 28 October 2022

Keywords:

Risk-bounded motion planning

Distributional robustness

Integrated perception & planning

ABSTRACT

Robust autonomy stacks require tight integration of perception, motion planning, and control layers, but these layers often inadequately incorporate inherent perception and prediction uncertainties, either ignoring them altogether or making questionable assumptions of Gaussianity. Robots with nonlinear dynamics and complex sensing modalities operating in an uncertain environment demand more careful consideration of how uncertainties propagate across stack layers. We propose a framework to integrate perception, motion planning, and control by explicitly incorporating perception and prediction uncertainties into planning so that risks of constraint violation can be mitigated. Specifically, we use a nonlinear model predictive control based steering law coupled with a decorrelation scheme based Unscented Kalman Filter for state and environment estimation to propagate the robot state and environment uncertainties. Subsequently, we use distributionally robust risk constraints to limit the risk in the presence of these uncertainties. Finally, we present a layered autonomy stack consisting of a nonlinear steering-based distributionally robust motion planning module and a reference trajectory tracking module. Our numerical experiments with nonlinear robot models and an urban driving simulator show the effectiveness of our proposed approaches.

© 2022 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Safety is a critical issue for robotic and autonomous systems that must traverse through uncertain environments. More sophisticated motion planning and control algorithms are needed as environments become increasingly dynamic and uncertain to ensure safe and effective autonomous behavior. Safely deploying robots in such dynamic environments requires a systematic accounting of various risks both within and across layers in an autonomy stack from perception to motion

[☆] This paper is part of the Special Issue: "Risk-aware Autonomous Systems: Theory and Practice".

^{☆☆} This work was completed when V. Renganathan was a PhD student at The University of Texas at Dallas, USA.

* Corresponding authors.

E-mail addresses: venkat@control.lth.se (V. Renganathan), sleiman.safaoui@utdallas.edu (S. Safaoui), aadi@mit.edu (A. Kothari), benjamin.gravell@utdallas.edu (B. Gravell), iman.shames@anu.edu.au (I. Shames), tyler.summers@utdallas.edu (T. Summers).

<https://doi.org/10.1016/j.artint.2022.103812>

0004-3702/© 2022 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

planning and control. Many widely used motion planning algorithms have been developed in deterministic settings. However, since motion planning algorithms must be coupled with the outputs of inherently uncertain perception systems, there is a crucial need for more tightly coupled perception and planning frameworks that explicitly incorporate perception and prediction uncertainties.

Motion planning under uncertainty has been considered in several lines of recent research [1–7]. Many approaches make questionable assumptions of Gaussianity and utilize chance constraints, ostensibly to maintain computational tractability. However, this can cause significant miscalculations of risk, and the underlying risk metrics do not necessarily possess desirable coherence properties [8,9]. The emerging area of distributionally robust optimization (DRO) shows that stochastic uncertainty can be handled in much more sophisticated ways without necessarily sacrificing computational tractability [10]. These approaches allow modelers to explicitly incorporate inherent ambiguity in probability distributions, rather than making overly strong structural assumptions on the distribution.

Traditionally, the perception and planning components in a robot autonomy stack are loosely coupled in the sense that nominal estimates from the perception system may be used for planning, while inherent perception uncertainties are usually ignored. This paradigm is inherited, in part, from the classical separation of estimation and control in linear systems theory. However, in the presence of uncertainties and constraints, estimation and control should *not* be separated as there are needs and opportunities to explicitly incorporate perception uncertainties into planning, both to mitigate risks of constraint violation [1,11,3,12,7,13–16] and to actively plan paths that improve perception [17]. Algorithms for motion planning under uncertainty have begun to address mission safety, state and control constraints, and trajectory robustness using chance constraints, as in [3]. It was recently broadened in [12,18,19] using a more general framework of axiomatic risk theory and distributionally robust optimization. The present paper is aimed towards establishing a systematic framework for integrating the perception and control components using a coherent risk assessment with distributionally robust risk constraints.

Contributions: This manuscript is a significant extension of our previous works [18,20]. In this paper, we relax the assumptions from our previous works by considering both motion model and sensor models to be nonlinear with additive uncertainties and propose a unified framework aimed towards a tighter integration of perception and planning in autonomous robotic systems. Apart from considering a full fledged nonlinear output model, we differ from our previous works by considering correlated process and sensor noises and employing a distance metric tailored for car-like vehicles to effectively capture the non-holonomic constraints while building our motion plan using sampling based motion planning algorithms. Our main contributions in this paper are:

1. We propose a distributionally robust incremental sampling-based motion planning framework that explicitly and coherently incorporates perception and prediction uncertainties. Our solution approach called Nonlinear Risk Bounded RRT* (NRB-RRT*) (Algorithm 1), approximates asymptotically optimal risk-bounded trajectories.
2. We design output feedback policies and consider moment-based ambiguity sets of distributions to enforce probabilistic collision avoidance constraints under the worst-case distribution in the ambiguity set (Algorithm 2). That is, trajectories generated using our output feedback policy are validated using a probabilistic collision check to ensure satisfaction of state risk constraints.
3. We formulate distributionally robust collision avoidance risk constraints. We demonstrate via numerical simulation results that this gives a more sophisticated and coherent risk quantification compared to an approach that accounts for uncertainty using Gaussian assumptions, without increasing the computation complexity.
4. We demonstrate our proposed algorithms and approaches using a unicycle model and a bicycle model in an open urban driving simulator [21] with static and dynamic obstacles and show that risk bounded motion planning can be achieved effectively for nonlinear robotic systems.

The rest of the paper is organized as follows. The integrated environment state formulation using the nonlinear dynamical model of the robot and the obstacle and their uncertainty modeling is discussed in §2. Next, we describe the layers of a distributionally robust autonomy stack and formulate a general stochastic optimal control problem in §3. An output feedback based steering law with uncertainty propagation is presented in §4. In §5, we describe the planner module that uses the (NRB-RRT*) motion planning algorithm for planning a reference trajectory. Then, in §6, we discuss about the reformulation technique for the distributionally robust risk constraints that define obstacle avoidance. The simulation results are then presented in §7. Finally, the paper summary and future research directions are discussed in §8.

Notation

The set of real numbers and natural numbers are denoted by $\mathbb{R}$ and $\mathbb{N}$, respectively. The subset of natural numbers between and including a and b with $a < b$ is denoted by $[a : b]$. The operators $|\cdot|$ and $(\cdot)^c$ denote the set cardinality and the set complement of its argument, respectively. The operators $\oplus, \setminus$ denote the set translation and set subtraction respectively. An identity matrix in dimension n is denoted by I_n . For a non-zero vector $x \in \mathbb{R}^n$ and a matrix $P \in \mathbb{S}_{++}^n$ (set of positive definite matrices), let $\|x\|_P = \sqrt{x^T P x}$.

2. Description of the robot & the environment

Consider a robot operating in an environment, $\mathcal{X} \subset \mathbb{R}^n$ cluttered with obstacles whose locations and motion are uncertain. The robot is assumed to be equipped with sensors and a perception system that support safe navigation through the environment. In what follows, we present the necessary ingredients for presenting the results such as the models for the robot, the environment, and the perception system.

2.1. Robot & environment model

2.1.1. Robot model

The robot is modeled as a stochastic discrete-time dynamical system:

$$x_{t+1} = \mathbf{f}(x_t, u_t) + w_{r,t}, \quad (1)$$

where $x_t \in \mathbb{R}^n$, $u_t \in \mathbb{R}^m$ are the robot state and input respectively at time t and $\mathbf{f}: \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ is a nonlinear function that represents the robot dynamics. The initial condition x_0 is subject to an uncertainty model with the true distribution $\mathbb{P}_{x_0}$ belonging to a moment-based ambiguity set $\mathcal{P}^{x_0}$, i.e., $\mathbb{P}_{x_0} \in \mathcal{P}^{x_0}$ where

$$\mathcal{P}^{x_0} := \left\{ \mathbb{P}_{x_0} \mid \mathbb{E}[x_0] = \bar{x}_0, \mathbb{E}[(x_0 - \bar{x}_0)(x_0 - \bar{x}_0)^\top] = \Sigma_{x_0} \right\}, \quad (2)$$

and $\bar{x}_0$ and Σ_{x_0} are some known parameters of appropriate dimensions. The additive process noise $w_{r,t} \in \mathbb{R}^n$ is a zero-mean random vector independent and identically distributed across time according to some prescribed distribution $\mathbb{P}_{w_r}$ with covariance Σ_{w_r} .

2.1.2. Environment model

The environment $\mathcal{X}$ is assumed to be convex and represented by finite number of linear inequalities

$$\mathcal{X} := \{x_t \mid A_e x_t \leq b_e\} \subset \mathbb{R}^n, \quad (3)$$

where the matrices A_e, b_e are of appropriate dimensions. We collectively refer the set of obstacles in the environment to be avoided as $\mathcal{B}$ with $|\mathcal{B}| = F \in \mathbb{N}$. We represent the shape of an obstacle $i \in \mathcal{B}$ at a given initial time ($t_0 = 0$) by $\mathcal{O}_{i,0} \subset \mathbb{R}^n$. For simplicity, $\forall i \in \mathcal{B}$ the obstacles shapes $\mathcal{O}_{i,0}$ are assumed to be convex polytopes. That is, for $i \in \mathcal{B}$

$$\mathcal{O}_{i,0} := \{x_0 \mid A_i x_0 \leq b_{i,0}\} \subset \mathbb{R}^n, \quad (4)$$

with the matrices $A_i \in \mathbb{R}^{n \times n}$, $b_{i,t} \in \mathbb{R}^n$. All obstacles are assumed to have an uncertain motion with possibly noisy prediction. Then, the free space in the environment at time t and the set defining the space occupied by the obstacle $i \in \mathcal{B}$ at any time t denoted by $\mathcal{O}_{i,t}$ are respectively given by

$$\mathcal{X}_t^{\text{free}} := \mathcal{X} \setminus \bigcup_{i \in \mathcal{B}} \mathcal{O}_{i,t}, \quad \text{where,} \quad (5)$$

$$\mathcal{O}_{i,t} = \mathbf{R}_{i,t-1} \mathcal{O}_{i,t-1} \oplus \{\bar{r}_{i,t}\} \oplus \{r_{i,t-1}\}, \quad \forall t \geq 1. \quad (6)$$

Here, $\bar{r}_{i,t} \in \mathbb{R}^n$ represents a known nominal translation and $r_{i,t} \in \mathbb{R}^n$ is a zero-mean random vector that represents the uncertain translation (possibly an unknown location or a noisy prediction of the obstacle motion) of the obstacle $i \in \mathcal{B}$ at time t , and it is assumed to follow a known distribution $\mathbb{P}_{i,t}^r$ with covariance $\Sigma_{i,t}^r$. Further, $\mathbf{R}_{i,t-1} \in \mathbb{R}^{n \times n}$ denotes the product of rotation matrices that represents the rotation of the obstacle $i \in \mathcal{B}$ at time t where we used the definition that multiplying a matrix $\mathbf{R}_{i,t}$ to a set $\mathcal{O}_{i,t}$ is defined by $\mathbf{R}_{i,t} \mathcal{O}_{i,t} := \{\mathbf{R}_{i,t} \cdot (o - c_{o,i}) + c_{o,i} \mid o \in \mathcal{O}_{i,t}\}$ with $c_{o,i} \in \mathbb{R}^n$ denoting the centroid of obstacle set $\mathcal{O}_{i,t}$. Although uncertainty in obstacle rotations could also be built into our framework, for the ease of exposition in the remainder of this paper, we shall assume that either $\mathbf{R}_{i,t}$ is known or $\mathbf{R}_{i,t}$ being an identity transformation (no rotation) at all time steps t , so that there is uncertainty only in the translation. We denote by $\mathbf{X}_{i,t} \in \mathbb{R}^l$, the states of the obstacle $i \in \mathcal{B}$, which for instance could represent the position and velocity of the centroid of the obstacle at time t . Then, the evolution of the state of the obstacle $i \in \mathcal{B}$ can be written as

$$\mathbf{X}_{i,t+1} = \mathbf{g}_{i,t}(\mathbf{X}_{i,t}) + w_{\mathcal{O},i,t}. \quad (7)$$

The random vector $r_{i,t}$ that resulted in the set translation in (6) manifests itself as the zero-mean process noise that affects the states of the obstacle $i \in \mathcal{B}$ as $w_{\mathcal{O},i,t} \in \mathbb{R}^l$ and $w_{\mathcal{O},i,t}$ is assumed to follow a known distribution $\mathbb{P}_{w_{\mathcal{O},i}}$ with covariance $\Sigma_{w_{\mathcal{O},i}}$. Further, $\mathbf{g}_{i,t}: \mathbb{R}^l \rightarrow \mathbb{R}^l$ denotes the function (possibly nonlinear) that represents the dynamics of the obstacle state at time t . For instance, the obstacle can be assumed to travel at a constant velocity but with process noise. Further, we assume that the uncertainty in the motion of obstacle i is independent from every other obstacle $j \in \mathcal{B}$, which implies $\mathbb{E}[w_{\mathcal{O},i,t} w_{\mathcal{O},j,t}^\top] = 0, \forall i, j \in \mathcal{B}, i \neq j$. Then, the joint evolution of all obstacles can be written as

$$\mathbf{X}_{\mathcal{O},t+1} = \mathbf{g}(\mathbf{X}_{\mathcal{O},t}) + \mathbf{w}_{\mathcal{O},t}, \quad (8)$$

where $\mathbf{X}_{\mathcal{O},t}$, $\mathbf{g}(\cdot)$, and $\mathbf{w}_{\mathcal{O},t}$ represent the concatenated states, nonlinear dynamics and the process noises of all the F obstacles at time t respectively. The additive process noise $\mathbf{w}_{\mathcal{O},t} \in \mathbb{R}^{Fl}$ is a zero-mean random vector independent and identically distributed across time according to some prescribed distribution $\mathbb{P}_{\mathbf{w}_{\mathcal{O}}}$ with covariance $\Sigma_{\mathbf{w}_{\mathcal{O}}}$. We assume that the process noises affecting the robot and the obstacles $i \in \mathcal{B}$ are independent of each other. That is, $\mathbb{E}[\mathbf{w}_{r,t} \mathbf{w}_{\mathcal{O},i,t}^\top] = 0, \forall i \in \mathcal{B}$.

2.1.3. Integrated environmental dynamics

We concatenate both the robot's state and the obstacle states at time t to form the environmental state

$$\mathcal{Z}_t = \begin{bmatrix} \mathbf{x}_t \\ \mathbf{X}_{\mathcal{O},t} \end{bmatrix} \in \mathbb{R}^{n_z}, \quad n_z = n + Fl. \quad (9)$$

Then the dynamics of the environmental state is given by

$$\mathcal{Z}_{t+1} = \underbrace{\begin{bmatrix} \mathbf{f}(\mathbf{x}_t, u_t) \\ \mathbf{g}(\mathbf{X}_{\mathcal{O},t}) \end{bmatrix}}_{\tilde{\mathbf{f}}(\mathcal{Z}_t, u_t)} + \underbrace{\begin{bmatrix} \mathbf{w}_{r,t} \\ \mathbf{w}_{\mathcal{O},t} \end{bmatrix}}_{\mathbf{w}_t}. \quad (10)$$

The initial condition $\mathcal{Z}_0$ is subject to an uncertainty model with the true distribution $\mathbb{P}_{\mathcal{Z}_0}$ of $\mathcal{Z}_0$ belonging to a moment-based ambiguity set $\mathcal{P}^{\mathcal{Z}_0}$, i.e., $\mathbb{P}_{\mathcal{Z}_0} \in \mathcal{P}^{\mathcal{Z}_0}$ where

$$\mathcal{P}^{\mathcal{Z}_0} := \left\{ \mathbb{P}_{\mathcal{Z}_0} \mid \mathbb{E}[\mathcal{Z}_0] = \bar{\mathcal{Z}}_0, \mathbb{E}[(\mathcal{Z}_0 - \bar{\mathcal{Z}}_0)(\mathcal{Z}_0 - \bar{\mathcal{Z}}_0)^\top] = \Sigma_{\mathcal{Z}_0} \right\}. \quad (11)$$

Here, $\mathbf{w}_t \in \mathbb{R}^{n_z}$ is a zero-mean random vector independent and identically distributed across time with covariance $\Sigma_{\mathbf{w}} = \begin{bmatrix} \Sigma_{\mathbf{w}_r} & 0 \\ 0 & \Sigma_{\mathbf{w}_{\mathcal{O}}} \end{bmatrix}$. Even when the distributions of the uncertainties affecting the obstacle motion are assumed to be exactly known in advance, the nonlinear dynamics of the obstacle transform the distributions for the future obstacle states (which are part of the environmental state $\mathcal{Z}_t$). At any instant of time t , $\mathbf{x}_t$ can be extracted from the environmental state $\mathcal{Z}_t$ as

$$\mathbf{x}_t = \underbrace{\begin{bmatrix} I_n & \mathbf{0}_{n \times Fl} \end{bmatrix}}_{C_{xr}} \mathcal{Z}_t. \quad (12)$$

2.2. Sensor model

In an autonomous robot, the environmental state $\mathcal{Z}_t$ must be estimated with a perception system represented by a noisy on-board sensor measurements. We assume that a high-level perception system, such as Semantic SLAM described in [22] is in place and it processes high dimensional raw data $\Theta_t \in \mathbb{R}^{N_\Theta}$, $N_\Theta \in \mathbb{N}$ to recognize the robot and the obstacles to produce noisy joint measurements of their respective states. In particular, the perception system recognizes the robot and obstacles through their distinctive features through mappings $\Upsilon_x : \mathbb{R}^{N_\Theta} \rightarrow \mathbb{R}^n$, $\Upsilon_{\mathcal{O}} : \mathbb{R}^{N_\Theta} \rightarrow \mathbb{R}^l$, labels the entities accordingly and returns a noisy measurement about their position and orientation. We abstract this whole process and assume that the perception system produces noisy measurements of the robot and obstacle states using an assumed nonlinear output model

$$\mathbf{y}_t = \mathcal{S}(\mathcal{Z}_t) + \mathbf{v}_t, \quad (13)$$

where $\mathbf{y}_t \in \mathbb{R}^p$ is the output measurement and $\mathcal{S} : \mathbb{R}^{n_z} \rightarrow \mathbb{R}^p$ is a nonlinear function. The additive measurement noise $\mathbf{v}_t \in \mathbb{R}^p$ is a zero-mean random vector independent and identically distributed across time according to some prescribed distribution $\mathbb{P}_{\mathbf{v}}$ with covariance $\Sigma_{\mathbf{v}}$. Cross correlated noises for instance in target tracking systems are known to occur when both process noise and the sensor noise are dependent on the system state as mentioned in [23]. Similarly, there can be interactions between the uncertainties of the perception system and the process disturbance, due to coupling between robot and environmental states in the perception and the use of an output feedback control policy. Ignoring the cross correlated noises will generally lead to incorrect dynamics for the state covariance propagation, and subsequently this may impose undue risk. To model the interaction, we assume that the process noise $\mathbf{w}_t$ and the measurement noise $\mathbf{v}_t$ are cross correlated with each other. That is, the joint noise $\begin{bmatrix} \mathbf{w}_t \\ \mathbf{v}_t \end{bmatrix}$ has zero mean with covariance

$$\Sigma_{\mathbf{wv}} = \mathbb{E} \left[\begin{bmatrix} \mathbf{w}_t \\ \mathbf{v}_t \end{bmatrix} \begin{bmatrix} \mathbf{w}_t \\ \mathbf{v}_t \end{bmatrix}^\top \right] = \begin{bmatrix} \Sigma_{\mathbf{w}} & M \\ M^\top & \Sigma_{\mathbf{v}} \end{bmatrix} \succ 0, \quad (14)$$

where $M = \mathbb{E}[\mathbf{w}_t \mathbf{v}_t^\top] \in \mathbb{R}^{n_z \times p}$ denotes the rank-one cross-correlation matrix across all time steps t and the matrix $M \neq \mathbf{0}_{n_z \times p}$ is such that the joint noise covariance $\Sigma_{\mathbf{wv}}$ given by (14) is positive definite.

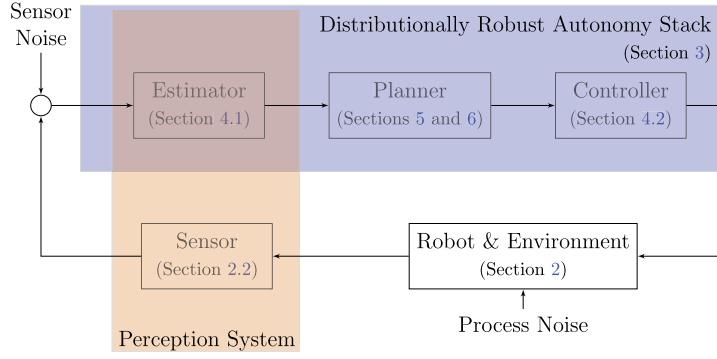


Fig. 1. A distributionally robust autonomy stack with integrated perception, planning, and control layers.

2.3. State & input constraints

The robot is nominally subject to constraints on the state and input of the form

$$x_t \in \mathcal{X}_t^{\text{free}}, \quad (15)$$

$$u_t \in \mathcal{U}, \quad (16)$$

for all $t = 0, \dots, T-1$, where $\mathcal{U} \subset \mathbb{R}^m$ is assumed to be convex polytope. Rewriting the description of the required sets using the robot inputs and environmental state, we see

$$\mathcal{U} = \{u_t \mid A_u u_t \leq b_u\}, \quad (17)$$

$$\mathcal{X} = \{z_t \mid A_0 C_{xr} z_t \leq b_0\}, \quad (18)$$

$$\mathcal{O}_{it} = \{z_t \mid A_i C_{xr} z_t \leq b_{it}\}, \quad i \in \mathcal{B} \quad (19)$$

where $b_u \in \mathbb{R}^{n_u}$, $b_0 \in \mathbb{R}^{n_e}$, $b_{it} \in \mathbb{R}^{n_{i_i}}$, and A_u , A_0 , and A_i are matrices of appropriate dimension. Suppose that n_{ob_i} and n_{env} be the number of constraints for obstacle $i \in \mathcal{B}$ and the environment $\mathcal{X}$ respectively. Then, the total number of constraints is denoted by

$$n_{total} = n_{env} + \sum_{i=1}^F n_{ob_i}. \quad (20)$$

3. Layers of a distributionally robust autonomy stack

The autonomy stack in an autonomous robot, as depicted in Fig. 1, can be partitioned into a hierarchy of i) a perception system that jointly estimates the robot and environmental states; ii) a motion planner which generates a reference trajectory through the environment, and iii) a feedback controller that tracks the reference trajectory online to mitigate the effects of disturbances. To emphasize our integrated approach, we first pose a general stochastic optimal control problem that incorporates all of these layers. Then we describe how the layers are integrated with explicit incorporation of uncertainties across layers in subsequent sections.

Given an initial state distribution $\mathcal{Z}_0 \sim \mathbb{P}_{\mathcal{Z}_0}$ and a set of final goal locations $\mathcal{X}_{goal} \subset \mathcal{X}$, we find a measurable output-and-input-history-dependent control policy $\pi = [\pi_0, \dots, \pi_{T-1}]$ with $u_t = \pi_t(y_{[0:t]}, u_{[0:t-1]})$ that moves the robot state mean to the goal set while respecting input constraints and distributionally robust collision risk constraints, while incorporating the ambiguities in the distributions described above. To this aim, we define the following distributionally robust constrained stochastic optimal control problem

$$\underset{\pi}{\text{minimize}} \quad \left[\sum_{t=0}^{T-1} \ell_t(\mathbb{E}[z_t], u_t) + \ell_T(\mathbb{E}[z_T]) \right] \quad (21a)$$

$$\text{subject to} \quad z_{t+1} = \tilde{f}(z_t, u_t) + w_t, \quad (21b)$$

$$y_t = \mathcal{S}(z_t) + v_t, \quad (21c)$$

$$z_0 \sim \mathbb{P}_{z_0} \in \mathcal{P}^{\mathcal{Z}}, \quad (21d)$$

$$w_t \sim \mathbb{P}_w(0, \Sigma_w), v_t \sim \mathbb{P}_v(0, \Sigma_v), \quad (21e)$$

$$u_t \in \mathcal{U} = \{u_t \mid A_u u_t \leq b_u\}, \quad (21f)$$

$$\sup_{\mathbb{P}_{Z_t} \in \mathcal{P}^Z} \mathcal{R}(C_{xr} Z_t \notin \mathcal{X}_t^{\text{free}}) \leq \alpha_t, \quad \forall t \in [0 : T], \quad (21g)$$

where $\mathcal{P}^Z$ is an ambiguity set of marginal state distributions, $\alpha_t \in (0, 0.5)$ is a user-prescribed stage risk parameter, and $\mathcal{R}(\cdot)$ is a risk measure that operates on the distribution of states and satisfies certain desirable axioms mentioned in [9]. The stage cost functions $\ell_t(\cdot)$ penalizes the robot's distance to the goal set and actuator effort, and are assumed to be expressed in terms of the environmental state mean $\mathbb{E}[Z_t]$, so that the state uncertainty appears only in the constraints.

Remark 1. We can have a more complex objective function than (21a). For instance, we could consider a distributionally robust risk measure related to the nominal state and terminal cost functions, with expectation taken with respect to the worst case distribution of the states and noises. Although such cost formulations are possible, they end up further increasing the complexity of the already existing harder optimization problem with distributionally robust risk constraint (which is an infinite dimensional constraint). Hence, to simplify the exposition we consider uncertainties only in the constraints.

Two key features distinguish our problem formulation. First, the state constraints are expressed as *distributionally robust risk constraints*, using the risk measure $\mathcal{R}(\cdot)$ and the ambiguity set $\mathcal{P}^Z$ whose construction will be detailed below. Since we will work with moment based ambiguity sets for the states, we give special consideration to the distributionally robust value-at-risk (DR-VaR) risk measure

$$\sup_{\mathbb{P}_{Z_t} \in \mathcal{P}^Z} \mathbb{P}_{Z_t}(C_{xr} Z_t \notin \mathcal{X}_t^{\text{free}}) \leq \alpha_t, \quad \forall t \in [0 : T], \quad (22)$$

which is a special case of (21g) with a tractable reformulation. This means that the nominal constraints $x_t \in \mathcal{X}_t^{\text{free}}$ are enforced with probability α_t at time t under the worst-case distribution in the ambiguity set and thus making them *distributionally robust chance constraints*.

Second, since information about the environmental state is obtained only from noisy measurements, we optimize over dynamic output feedback policies. This policy encompasses the full autonomy stack from perception to planning to control. However, the problem is infinite-dimensional, and moreover, the distributionally robust risk constraint in (22) is also infinite-dimensional and inherently non-convex due to the underlying obstacle avoidance constraints. Thus, solving (21) exactly is essentially impossible. Instead, we aim for an approximate solution. Our proposed solution framework, detailed below, integrates a dynamic nonlinear state estimator (the estimator block in Fig. 1), a nonlinear feedback controller (the controller block in Fig. 1), and a sampling-based kinodynamic motion planning under uncertainty algorithm (the planner block in Fig. 1). Specifically, in this work we propose using an Unscented Kalman filter as the nonlinear estimator, a nonlinear model predictive control as the nonlinear feedback controller, and a novel distributionally robust, kinodynamic variant of the RRT* called Nonlinear Risk Bounded RRT*(NRB-RRT*) as the planner.¹ This integration and explicit incorporation of state estimation uncertainty into the motion planning and control takes a step toward tighter integration of perception, planning, and control, which are nearly always separated in state-of-the-art robotic systems. In our work, both the planning and the tracking modules make use of the same output feedback approach for accomplishing their tasks, with the planning module using it to steer to a feasible sampled point while building the motion plan, and the tracking module using it to steer the mean of the robot to the next reference point in the reference trajectory.

4. Output feedback based steering law: Unscented Kalman Filter with nonlinear model predictive controller

Sampling based motion planning algorithms require a steering law to steer the robot from one pose to another pose in the free space. Since the environment state has nonlinear dynamics and must be estimated from noisy nonlinear output measurements (13), our proposed steering law $\pi = [\pi_0, \dots, \pi_{T-1}]$ with $u_t = \pi_t(y_{0:t}, u_{0:t-1})$ comprises a combination of nonlinear dynamic state estimator and a nonlinear feedback controller. We utilize the Unscented Kalman Filter (UKF) for joint estimation of robot and obstacle states. The UKF provides derivative-free estimation of the first and second moments that are more accurate than the Extended Kalman Filter while remaining computationally tractable. A nonlinear model predictive controller as described in [24], [25] is then utilized based on the state estimate from the UKF.

4.1. The Unscented Kalman Filter (UKF)

In this subsection, we elaborate about the Unscented Kalman Filter algorithm which serves as an estimator module for a perception system present in the autonomy stack shown in Fig. 1. Later in subsection 4.2, we will describe how the estimates from UKF are then used to realize an output feedback based controller.

¹ It is worth noting that other choices for estimator, controller, and planner subsystems can be incorporated in our proposed framework, nevertheless, in this work, for reasons that will be explained later, we make the aforementioned choices for these subsystems.

Noise Decorrelation. Before describing the UKF, we present a decorrelation scheme [26] to handle cross-correlation between process noise and sensor induced by the coupling between perception and control layers from (14). The scheme uses a pseudo state process equation to reconstruct a corresponding pseudo process noise, which is no longer correlated with the measurement noise. It is evident from (13) that $y_t - \mathcal{S}(\mathcal{Z}_t) - v_t = 0$. Then, the environmental dynamics given in (10) can be rearranged as follows

$$\mathcal{Z}_{t+1} = \tilde{f}(\mathcal{Z}_t, u_t) + w_t + \mathcal{H}(y_t - \mathcal{S}(\mathcal{Z}_t) - v_t) \quad (23)$$

$$= \underbrace{\tilde{f}(\mathcal{Z}_t, u_t) - \mathcal{H}\mathcal{S}(\mathcal{Z}_t) + \mathcal{H}y_t}_{f^*(\mathcal{Z}_t, u_t)} + \underbrace{w_t - \mathcal{H}v_t}_{w_t^*}, \quad (24)$$

where $f^*(\mathcal{Z}_t, u_t)$, w_t^* are the corresponding pseudo process dynamics and pseudo process noise respectively. The term $\mathcal{H}y_t$ is considered as a known input and is approximated with $\mathcal{H}\hat{y}_t$. The pseudo gain term $\mathcal{H}$ is a design parameter chosen such that the cross-correlation between the pseudo process noise and the sensor noise is made zero. That is,

$$\mathbb{E}[w_t^* v_t^\top] = 0 \implies \mathbb{E}[(w_t - \mathcal{H}v_t)v_t^\top] = M - \mathcal{H}\Sigma_v = 0, \implies \mathcal{H} = M\Sigma_v^{-1}.$$

The mean and the covariance of the pseudo process noise are then given by

$$\mathbb{E}[w_t^*] = \mathbb{E}[w_t - \mathcal{H}v_t] = \mathbb{E}[w_t] - \mathcal{H}\mathbb{E}[v_t] = 0, \quad (25)$$

$$\begin{aligned} \Sigma_w^* &= \mathbb{E}[w_t^* w_t^{*\top}] = \mathbb{E}[(w_t - \mathcal{H}v_t)(w_t - \mathcal{H}v_t)^\top] \\ &= \Sigma_w - M\mathcal{H}^\top - \mathcal{H}M^\top + \mathcal{H}\Sigma_v\mathcal{H}^\top \\ &= \Sigma_w - M\Sigma_v^{-1}M^\top - \mathcal{H}\Sigma_v\mathcal{H}^\top + \mathcal{H}\Sigma_v\mathcal{H}^\top \\ &= \Sigma_w - M\Sigma_v^{-1}M^\top. \end{aligned} \quad (26)$$

Then, applying Schur complement on (14), it is easy to observe that Σ_w^* is positive definite as well:

$$\begin{bmatrix} \Sigma_w^* & M \\ M^\top & \Sigma_v \end{bmatrix} \succ 0 \iff \Sigma_v \succ 0, \underbrace{\Sigma_w^* - M\Sigma_v^{-1}M^\top}_{\Sigma_w^*} \succ 0.$$

With the new pseudo process equation $f^*(\cdot)$ given by (24) and the measurement update given by (13), the standard UKF can be implemented as the new process noise w_t^* is no longer cross-correlated with the sensor noise v_t .

Remark 2. The complexity of computing the Σ_w^* using (26) is $O(\max(p, n_z)^3)$. If the measurement function is known a priori, this must be done just once. In case, we need to recompute it by adding a new measurement function (e.g., when a new landmark is observed we add a new row), then it can be done incrementally via the Sherman-Morrison-Woodbury matrix inversion formula. Such a detailed exposition is out of the scope of this paper and hence we assume the measurement function to be fixed.

The Unscented Kalman Filter. The UKF is a popular tool for nonlinear state estimation. It uses the so-called Unscented Transformation (UT) in both the propagation and update step, yielding derivative-free Kalman filtering for nonlinear systems. For Gaussian inputs, the moment estimates from UT are accurate up to the third order approximation and for the case of non-Gaussian, the approximations are accurate to at least the second-order [27]. An ensemble of $2n_z + 1$ samples called the *sigma points* following the Van Der Merwe algorithm is generated deterministically as follows:

$$\chi_0 = \hat{\mathcal{Z}}_{t-1} \quad (27)$$

$$\chi_i = \hat{\mathcal{Z}}_{t-1} + \left[\sqrt{(n_z + \lambda)\hat{\Sigma}_{\mathcal{Z}_{t-1}}} \right]_i, \quad i = [1 : n_z] \quad (28)$$

$$\chi_{i+n} = \hat{\mathcal{Z}}_{t-1} - \left[\sqrt{(n_z + \lambda)\hat{\Sigma}_{\mathcal{Z}_{t-1}}} \right]_i, \quad i = [1 : n_z], \quad (29)$$

where $\hat{\mathcal{Z}}_{t-1}$, $\hat{\Sigma}_{\mathcal{Z}_{t-1}}$ denote the posterior UKF estimate of mean and covariance of the environmental state at $t - 1$ and $\left[\sqrt{(n_z + \lambda)\hat{\Sigma}_{\mathcal{Z}_{t-1}}} \right]_i$ is the i^{th} row or column of the matrix square root $\sqrt{(n_z + \lambda)\hat{\Sigma}_{\mathcal{Z}_{t-1}}}$. The weights of the sigma points in the calculation of propagated mean and covariance are

$$W_0^{(m)} = \frac{\lambda}{n_z + \lambda}, \quad W_0^{(c)} = \frac{\lambda}{n_z + \lambda} + 1 - \alpha_u^2 + \beta_u, \quad (30)$$

$$W_i^{(m)} = W_i^{(c)} = \frac{\lambda}{2(n_z + \lambda)}, \quad i = 1, 2, \dots, 2n_z. \quad (31)$$

The scaling parameter $\lambda = \alpha_u^2(n + \kappa) - n_z$, where $\alpha_u, \beta_u, \kappa$ are used to tune the unscented transformation. Then, every sigma point is propagated through non-linear state update equation to yield an ensemble of sigma points capturing the *a priori* statistics of $\mathcal{Z}_t$ namely $\mathcal{Z}_t^-, \Sigma_{\mathcal{Z}_t}^-$. In the update step, we use the transformed set of sigma points described above and propagate them using the given nonlinear measurement model. Finally, the *a posteriori* state $\hat{\mathcal{Z}}_t$ and covariance $\hat{\Sigma}_{\mathcal{Z}_t}$ are computed using the output residual and the obtained Unscented Kalman filter gain, $\mathcal{L}_t$ as follows

$$\xi_i = f^*(\chi_i, u_t), \quad i = 0, 1, \dots, 2n_z, \quad (32)$$

$$\mathcal{Z}_t^- = \sum_{i=0}^{2n_z} W_i^{(m)} \xi_i, \quad (33)$$

$$\Sigma_{\mathcal{Z}_t}^- = \sum_{i=0}^{2n_z} W_i^{(c)} (\xi_i - \mathcal{Z}_t^-)(\xi_i - \mathcal{Z}_t^-)^\top + \Sigma_w^* \quad (34)$$

$$\Theta_i = \mathcal{S}(\xi_i), \quad i = 0, 1, \dots, 2n_z, \quad (35)$$

$$\mu_\Theta = \sum_{i=0}^{2n_z} W_i^{(m)} \Theta_i, \quad (36)$$

$$\Sigma_\Theta = \sum_{i=0}^{2n_z} W_i^{(c)} (\Theta_i - \mu_\Theta)(\Theta_i - \mu_\Theta)^\top + \Sigma_v \quad (37)$$

$$\mathcal{L}_t = \left[\sum_{i=0}^{2n_z} W_i^{(c)} (\xi_i - \mathcal{Z}_t^-)(\Theta_i - \mu_\Theta)^\top \right] \Sigma_\Theta^{-1} \quad (38)$$

$$\hat{\mathcal{Z}}_t = \mathcal{Z}_t^- + \mathcal{L}_t(y_t - \mu_\Theta) \quad (39)$$

$$\hat{\Sigma}_{\mathcal{Z}_t} = \Sigma_{\mathcal{Z}_t}^- - \mathcal{L}_t \Sigma_\Theta \mathcal{L}_t^\top. \quad (40)$$

Remark 3. While the values of the scaling parameters $\lambda, \alpha_u, \beta_u, \kappa$ that yield third-order approximation accuracy for Gaussian state distributions are well known, there are still no known guidelines to choose the best values for the above parameters when the state distributions are non-Gaussian. With the environmental state distribution being generally non-Gaussian and along with the moments calculation being done using finite number of deterministic samples result in the obtained moments being only approximations of the true moments of the non-Gaussian state distributions. This motivates our use of distributionally robust risk constraints to explicitly account for this non-Gaussianity. The introduction of nonlinear output model along with the decorrelation scheme renders our current output feedback based steering approach significantly different from our previous work [20]. The complexity (costliest operation) of UKF lies in the matrix square-root operation used in the (28) and (29). This is the price that the UKF pays for being in the middle ground between the Extended Kalman Filter (EKF) and the Particle Filter (PF), as it avoids both expensive Jacobian computation of the EKF and the PF's need for large numbers of particles to approximate the moments. An incremental UKF update as in [28] is possible, but is out of the scope of this paper. This calls for a design of better nonlinear filtering techniques in future to improve risk assessments for collision with obstacles.

4.2. Nonlinear model predictive control (NMPC) law

In this subsection, we elaborate about the feedback controller that represents an important module in the autonomy stack shown in Fig. 1. Since the robot dynamics is both nonlinear and stochastic, we will be describing an output feedback based nonlinear model predictive control algorithm to steer the robots in the considered setting. Specifically, the steering law for steering the robots is realized through a multiple shooting-based nonlinear model predictive controller [24], [25] that uses the state estimate obtained through an Unscented Kalman filter at each time step. An illustration of the output feedback is shown in Fig. 2. The error at a time step t is

$$e_t = C_{xr} \hat{\mathcal{Z}}_t - x_s = \hat{x}_t - x_s, \quad (41)$$

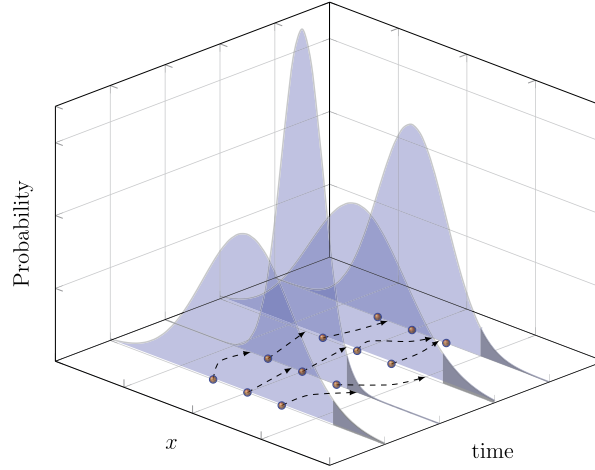


Fig. 2. An output feedback based controller using NMPC and UKF is illustrated here in $\mathbb{R}$. The sigma points and the distribution of the robot states at different time steps are depicted with orange dots and shaded blue area respectively. The NMPC-steered trajectory from each sigma point is shown in dotted black lines. At each time step t , the distributionally robust budget risk constraint is shown using darkly shaded color whose area is at most α_t and the corresponding sigma points are steered using the NMPC control law. The distribution at time $t + 1$ is described using the approximated moments calculated using those transformed sigma points. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

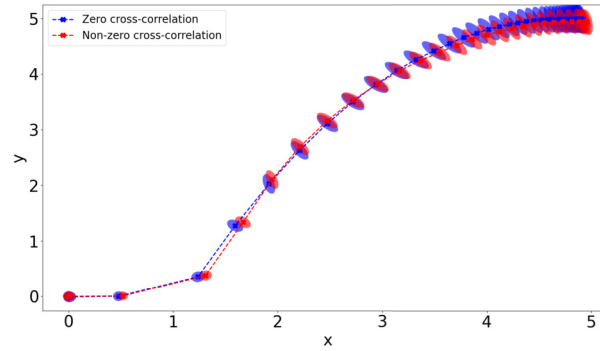


Fig. 3. This plot shows the simulated trajectories of a bicycle robot as in (53) using NMPC control law coupled with UKF based state estimator. The trajectories with $M = 0_{6 \times 4}$ and with $M = 10^{-3} \begin{bmatrix} M_1 & 0 \\ 0 & M_1 \end{bmatrix}$, $M_1 = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$ have different moments for the system states and hence will have different total trajectory cost and different risk evaluations.

where x_s represents a sample in the free space to be steered to. At every time instance t , the nonlinear model predictive controller repeatedly solves the following optimization problem in a receding horizon fashion with prediction horizon N_t to find a control input sequence as follows

$$\begin{aligned}
 u_t^\dagger = \underset{\{u_k\}_{k=t}^{t+N_t-1}}{\operatorname{argmin}} \quad & \sum_{k=t}^{t+N_t-1} \left(\|e_k\|_Q^2 + \|u_k\|_R^2 \right) + \|e_{k+N_t}\|_Q^2, \\
 \text{subject to} \quad & \hat{x}_{k+1} = \mathbf{f}(\hat{x}_k, u_k), \\
 & \hat{x}_k \in \mathcal{X}, \quad u_k \in \mathcal{U},
 \end{aligned} \tag{42}$$

where only the first control input in the optimal sequence $u_t^\dagger$ is applied at time t and the horizon is shifted to $t + 1$ for the problem to be solved again. Here, Q, R denote the state penalty and control penalty matrices respectively. To illustrate the need for considering the cross-correlation between the noises, we simulated a bicycle robot using the NMPC control law and the UKF with and without the cross-correlated noises. It is clear from Fig. 3 that the cost of steering from a source to the destination differs between the two cases as the state moments are different at all time steps due to the cross-correlation between the noises. This is a crucial observation for risk based planning as the risk of obstacle collision, for a given trajectory, is computed using the resulting moments which in turn determines the safety of the path.

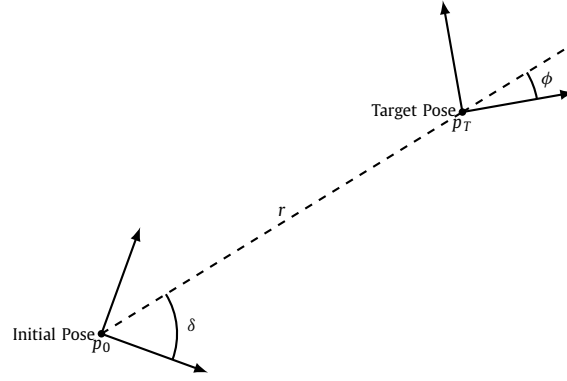


Fig. 4. The problem of steering a non-holonomic robot from an initial pose p_0 to a target pose p_T with different orientation is shown here.

5. Sampling-based motion planning algorithm

The planner module of an autonomy stack as shown in the Fig. 1 is responsible for planning a path from a given source to a desired destination adhering to all motion planning specifications. Typically, the planner module solves a motion planning problem as in (21) to design the reference trajectory from source to destination. However, due to the non-convex collision avoidance constraints and presence of uncertainties, it is difficult to exactly solve the motion planning problem subject to the distributionally robust risk constraints. Hence, we resort to a sampling based motion planner, which incrementally constructs a motion plan from the source to the destination for a robot by sampling a point in the obstacle free space and connecting the point to the tree of motion plans if the trajectory connecting the points is collision-free. We propose to use a distributionally robust, kinodynamic variant of the RRT* motion planning algorithm with dynamic output feedback policies. Our proposed algorithm, called Nonlinear Risk Bounded RRT* (NRB-RRT*), grows trees of state and state estimate *distributions*, rather than merely trees of states, and incorporates distributionally robust risk constraints to build risk-bounded state trajectories and feedback policies. The NRB-RRT* tree expansion procedure, inspired by the chance constrained RRT* algorithm developed in [4], is presented in Algorithm 1. The NRB-RRT* tree is denoted by $\mathcal{T}$, consisting of $|\mathcal{T}|$ nodes. Each node N of the tree $\mathcal{T}$ consists of a sequence of state distributions, characterized by a distribution mean $\hat{x}$ and covariance D . A sequence of means and covariance matrices is denoted by $\bar{\sigma}$ and $\bar{\Pi}$, respectively. The final mean and covariance of a node's sequence are denoted by $x[N]$ and $D[N]$, respectively. For the state distribution sequence $(\bar{\sigma}, \bar{\Pi})$, the notation $\Delta J(\bar{\sigma}, \bar{\Pi})$ denotes the cost of that sequence. If $(\bar{\sigma}, \bar{\Pi})$ denotes the trajectory of node N with parent N_{parent} , then we denote by $J[N]$, the entire path cost from the starting state to the terminal state of node N , constructed recursively as

$$J[N] = J[N_{parent}] + \Delta J(\bar{\sigma}, \bar{\Pi}). \quad (43)$$

5.1. Approximating optimal cost-to-go

In order to efficiently explore the reachable set of the dynamics and increase the likelihood of generating collision-free trajectories, authors in [29] described the limitations of the standard Euclidean distance metric and rather advocated the use of an optimal “cost-to-go” metric, that takes into account a dynamics and control-related quantities to steer the robot. When the dynamics are linear and obstacles are ignored, the optimal “cost-to-go” between any two nodes in the tree can be computed using dynamic programming. However, with nonlinear dynamics and potentially non-holonomic constraints, cost-to-go can be tailored to the specific robot dynamics. Inspired by [30], we present an approximate cost-to-go for some special robot representations and car-like dynamics. The initial pose of a robot, p_0 can be defined using a tuple (x, y, ψ) , where (x, y) are the positions in global coordinate frame and ψ being its orientation with respect to the positive x axis. We use an egocentric polar coordinate system to describe the relative location of the target pose p_T observed from the robot with radial distance r along the line of sight vector from the robot to the target, orientation of the target ϕ , and orientation of the robot δ , where angles are measured from the line of sight vector as shown in the Fig. 4.

Definition 1. The non-holonomic directed distance from an initial pose p_0 to a target pose p_T , with non-holonomic constraints is defined as

$$\mathfrak{D}(p_0, p_T) = \sqrt{r^2 + k_\phi^2 \phi^2} + k_\delta |\delta|. \quad (44)$$

Here, $k_\phi > 0$ is a constant that represent the weight of ϕ with respect to the radial distance r and $k_\delta > 0$ is another constant that emphasizes the weight on $|\delta|$ as δ can take both positive and negative values. The non-holonomic distance in (44) is asymmetric meaning that $\mathfrak{D}(p_0, p_T)$ is not equal to $\mathfrak{D}(p_T, p_0)$.

5.2. Outline of NRB-RRT* algorithm

In the first step, using the `Sample(·)` function, a random pose x_{rand} is sampled from the free space $\mathcal{X}_t^{\text{free}}$. Then the tree node, $N_{nearest}$ that is nearest to the sampled pose is selected using the `NearestNode(·)` function (line 3 of Algorithm 1) which uses the “cost-to-go” distance metric defined in (44). Attempts are then made to steer the robot from the nearest tree node to the random sample using the `Steer(·)` function that employs the steering law explained in Section 4 (line 4 of Algorithm 1). The control policy obtained is then used to propagate the state mean and covariance, and the entire trajectory $(\bar{\sigma}, \bar{\Pi})$ is returned by the `Steer(·)` function. Using the `DR-Feasible(·)` function, each state distribution in the trajectory is checked for distributionally robust probabilistic constraint satisfaction discussed in the next section. Further, the line connecting subsequent state distributions in the trajectory is also checked for collision with the obstacle sets $\mathcal{O}_{it}, i \in \mathcal{B}$. An outline of the `DR-Feasible` subroutine is shown in Algorithm 2. If the entire trajectory $(\bar{\sigma}, \bar{\Pi})$ is probabilistically feasible, a new node N_{min} with that distribution sequence $(\bar{\sigma}, \bar{\Pi})$ is created (line 7 of Algorithm 1) but not yet added to $\mathcal{T}$. Instead, nearby nodes are identified for possible connections via the `NearNodes(·)` function (line 8 of Algorithm 1), which returns a subset of nodes $\mathcal{N}_{near} \subseteq \mathcal{T}$, if they are within a search radius r ensuring probabilistic asymptotic optimality guarantees specified in [31]. The radius r is given by

$$r = \min \left\{ \gamma \left(\frac{\log(|\mathcal{N}_t|)}{\mathcal{N}_t} \right)^{\frac{1}{d+1}}, \mu_{max} \right\}, \quad (45)$$

where $\mathcal{N}_t$ refers to the number of nodes in the tree at time t , the positive scalar μ_{max} is the maximum radius specified by the user, and γ refers to the planning constant based on the d dimensional environment and is selected using [31, Theorem 1]. Then we seek to identify the lowest-cost, probabilistically feasible connection from the $\mathcal{N}_{near}$ nodes to x_{rand} (lines 10-14 of Algorithm 1). For each possible connection, a distribution sequence is simulated via the steering law (line 11 of Algorithm 1). If the resulting sequence is probabilistically feasible, and the cost of that node represented as $c_{rand} = J[N_{near}] + \Delta J(\bar{\sigma}, \bar{\Pi})$, is lower than the cost of N_{min} denote by $J[N_{min}]$, then a new node with this sequence replaces N_{min} (line 14 of Algorithm 1). The lowest-cost node is ultimately added to $\mathcal{T}$ (line 15 of Algorithm 1). Finally, edges are rewired based on attempted connections from the new node N_{min} to nearby nodes $\mathcal{N}_{near}$ (lines 17-22 of Algorithm 1), ancestors excluded (line 17 of Algorithm 1) which are found using `Ancestors(·)` function. A distribution sequence is simulated via the steering law from N_{min} to the terminal state of each nearby node $N_{near} \in \mathcal{N}_{near}$ (line 18 of Algorithm 1). If the resulting sequence is probabilistically feasible, and the cost of that node c_{min} is lower than the cost of N_{near} given by $J[N_{near}]$ (line 19 of Algorithm 1), then a new node with this distribution sequence replaces N_{near} (lines 21-22 of Algorithm 1). The tree expansion procedure is then repeated until a node from the goal set is added to the tree. At that point, a distributionally robust feasible trajectory is obtained from the tree root to $\mathcal{X}_{goal}$.

Algorithm 1 NRB-RRT*-tree expansion procedure.

```

1: Inputs: Current Tree  $\mathcal{T}$ , time step  $t$ 
2:  $x_{rand} \leftarrow \text{Sample}(\mathcal{X}_t^{\text{free}})$ 
3:  $N_{nearest} \leftarrow \text{NearestNode}(x_{rand}, \mathcal{T})$ 
4:  $(\bar{\sigma}, \bar{\Pi}) \leftarrow \text{Steer}(\hat{x}[N_{nearest}], D[N_{nearest}], x_{rand})$ 
5: // Check if sequence  $(\bar{\sigma}, \bar{\Pi})$  is DR-Feasible
6: if DR-Feasible( $\bar{\sigma}, \bar{\Pi}$ ) then
7:   Create node  $N_{min}\{\bar{\sigma}, \bar{\Pi}\}$ 
8:    $\mathcal{N}_{near} \leftarrow \text{NearNodes}(\mathcal{T}, x_{rand}, |\mathcal{T}|)$ 
9:   // Connect via a minimum-cost path
10:  for each  $N_{near} \in \mathcal{N}_{near} \setminus N_{nearest}$  do
11:     $(\bar{\sigma}, \bar{\Pi}) \leftarrow \text{Steer}(\hat{x}[N_{near}], D[N_{near}], x_{rand})$ 
12:     $c_{rand} \leftarrow J[N_{near}] + \Delta J(\bar{\sigma}, \bar{\Pi})$ 
13:    if DR-Feasible( $\bar{\sigma}, \bar{\Pi}$ ) &  $c_{rand} < J[N_{min}]$  then
14:      Replace  $N_{min}$  with  $N_{min}\{\bar{\sigma}, \bar{\Pi}\}$ 
15:  Add  $N_{min}$  to  $\mathcal{T}$ 
16:  // Re-Wire the Tree
17:  for each  $N_{near} \in \mathcal{N}_{near} \setminus \text{Ancestors}(N_{min})$  do
18:     $(\bar{\sigma}, \bar{\Pi}) \leftarrow \text{Steer}(\hat{x}[N_{min}], D[N_{min}], \hat{x}[N_{near}])$ 
19:     $c_{min} \leftarrow J[N_{min}] + \Delta J(\bar{\sigma}, \bar{\Pi})$ 
20:    if DR-Feasible( $\bar{\sigma}, \bar{\Pi}$ ) &  $c_{min} < J[N_{near}]$  then
21:      Delete  $N_{near}$  from  $\mathcal{T}$ 
22:      Add new node  $N_{new}\{\bar{\sigma}, \bar{\Pi}\}$  to  $\mathcal{T}$ 

```

6. Distributionally robust collision check

It is necessary to ensure that the total risk of the trajectory returned by the planner module of the autonomy stack in Fig. 1 does not exceed the given total risk budget and thereby agree with the motion planning specifications. In this section,

Algorithm 2 DR-Feasible subroutine.

```

Input:  $\mathbf{T}$ —time step distribution sequence  $(\bar{\sigma}, \bar{\Pi})$ 
2: for  $t = 1$  to  $\mathbf{T}$  do
     $(\hat{x}_t, D_{\hat{x}_t}) \leftarrow t^{th}$  element in  $(\bar{\sigma}, \bar{\Pi})$  sequence
4:    $\mathbb{L} \leftarrow$  Line connecting position block of  $\hat{x}_{t-1}$  to  $\hat{x}_t$ 
    for each  $i \in \mathcal{B}$  do
6:       if  $(\hat{x}_t, D_{\hat{x}_t})$  dissatisfies (50) or  $\mathbb{L} \cap \mathcal{O}_{it} \neq \emptyset$  then
           Return false
8: Return true

```

we evaluate the safety of a trajectory returned by the planner module using distributionally robust risk constraints and formally present a risk treatment to ensure the safety of the whole mission plan.

6.1. Moment-based ambiguity set to model uncertainty

Unlike most stochastic motion planning algorithms that often assume a functional form (often Gaussian) for probability distributions to model uncertainties, we will focus here on uncertainty modeling using moment-based ambiguity sets. Since the initial environmental state $\mathcal{Z}_0$ is not assumed to be Gaussian, then neither are the environmental state distributions $\mathbb{P}_{\mathcal{Z}_t}$ at any point of future time t . Hence, we define an ambiguity set for the whole environmental state by

$$\mathcal{P}^{\mathcal{Z}_t} = \left\{ \mathbb{P}_{\mathcal{Z}_t} \mid \mathbb{E}[\mathcal{Z}_t] = \hat{\mathcal{Z}}_t, \mathbb{E}[(\mathcal{Z}_t - \hat{\mathcal{Z}}_t)(\mathcal{Z}_t - \hat{\mathcal{Z}}_t)^\top] = \Sigma_{\mathcal{Z}_t} \right\}. \quad (46)$$

6.2. Risk treatment for trajectory safety

To study the stage risk constraints (21) for the safety of the planned reference trajectory, we follow the steps similar to the one described in [20] which was inspired by the risk treatment developed by authors in [3] and [4]. Since, the robot dynamics are nonlinear, we generalize the risk treatment developed by authors in [3] to handle future states with arbitrary distributions. Let $\mathfrak{P}_{Safe}$ denote the event that plan $\mathfrak{P}$ succeeds and $\mathfrak{P}_{Fail}$ as the complementary event (i.e. failure). Given a confidence $\beta \in (0, 0.5]$ and a total time horizon T , we consider the specification that a plan succeeds with high probability or equivalently that it fails with low probability. That is,

$$\mathbb{P}(\mathfrak{P}_{Safe}) \geq 1 - \beta \iff \mathbb{P}(\mathfrak{P}_{Fail}) \leq \beta. \quad (47)$$

Failure of the total plan requires at least one stage risk constraints to be violated. Then, $\forall t \in [0 : T]$,

$$\begin{aligned} \inf_{\mathbb{P}_{\mathcal{Z}_t} \in \mathcal{P}^{\mathcal{Z}}} \mathbb{P}_{\mathcal{Z}_t}(C_{xr} \mathcal{Z}_t \in \mathcal{X}_t^{\text{free}}) &\geq 1 - \alpha_t \\ \Leftrightarrow \sup_{\mathbb{P}_{\mathcal{Z}_t} \in \mathcal{P}^{\mathcal{Z}}} \mathbb{P}_{\mathcal{Z}_t}(C_{xr} \mathcal{Z}_t \notin \mathcal{X}_t^{\text{free}}) &\leq \alpha_t. \end{aligned}$$

Applying Boole's law, probability of the success event can be bounded as

$$\begin{aligned} \mathbb{P}(\mathfrak{P}_{Safe}) &= 1 - \mathbb{P}(\mathfrak{P}_{Fail}) = 1 - \mathbb{P}_{\mathcal{Z}_t} \left(\bigcup_{t=0}^T C_{xr} \mathcal{Z}_t \notin \mathcal{X}_t^{\text{free}} \right) \\ &\geq 1 - \sup_{\mathbb{P}_{\mathcal{Z}_t} \in \mathcal{P}^{\mathcal{Z}}} \mathbb{P}_{\mathcal{Z}_t} \left(\bigcup_{t=0}^T C_{xr} \mathcal{Z}_t \notin \mathcal{X}_t^{\text{free}} \right) \\ &\geq 1 - \sum_{t=0}^T \sup_{\mathbb{P}_{\mathcal{Z}_t} \in \mathcal{P}^{\mathcal{Z}}} \mathbb{P}_{\mathcal{Z}_t}(C_{xr} \mathcal{Z}_t \notin \mathcal{X}_t^{\text{free}}) \\ &\geq 1 - \underbrace{\sum_{t=0}^T \alpha_t}_{:=\beta} \end{aligned}$$

If the stage risks $\forall t \in [0 : T]$ are equal, meaning $\alpha_t = \alpha$, then $\beta = (T + 1)\alpha$. Furthermore, if the stage risk $\alpha_t = \alpha$ is equally distributed over all n_{total} constraints, then, a risk bound for a single constraint is $\frac{\alpha}{n_{total}}$. Subsequently, the corresponding risk bound for an obstacle $i \in \mathcal{B}$ and the environment $\mathcal{X}$ would be $\frac{\alpha n_{obj}}{n_{total}}$ and $\frac{\alpha n_{env}}{n_{total}}$ respectively. Usually the choice of T affects the design of the stage risk α_t . It is typical to start with a conservative upper bound T_{max} (usually an informed estimate

obtained using a trajectory optimization technique) and perform an uniform risk allocation to design a stage risk α_t . If a reference path happens to be tracked by the robot in an earlier time say $T' < T_{max}$, then we can refine our choice of T_{max} using T' and re-run the risk allocation. It is possible to have a spatio-temporal based risk allocation and it is currently being pursued as a future work.

6.3. Distributionally robust collision check

The non-convex obstacle avoidance constraints for obstacle $i \in \mathcal{B}$ can be expressed as the disjunction

$$\neg(A_i C_{xr} \mathcal{Z}_t \leq b_{it}) \Leftrightarrow \bigvee_{j=1}^{n_i} (a_{ij}^\top C_{xr} \mathcal{Z}_t \geq a_{ij}^\top c_{ijt}), \quad (48)$$

where $\vee$ denotes disjunction (logical OR operator) and $\neg$ denotes negation (logical NOT operator) and $c_{ijt} = \hat{c}_{ijt} + r_{i,t}$ is a point nominally on the j th constraint of the i th obstacle whose covariance is the same as that of $r_{i,t}$. The control law returned by the steering function should also satisfy the state constraints which are expressed as distributionally robust chance constraints. The nominal state constraints, $C_{xr} \mathcal{Z}_t \in \mathcal{X}_t^{\text{free}}$, are required to be satisfied with probability $1 - \alpha$, under the worst case probability distribution in the ambiguity set. Let the mean and covariance of the uncertain obstacle motion be defined using $\mathbf{E}[c_{ijt}] = \hat{c}_{ijt}$ and $\mathbf{E}[(c_{ijt} - \hat{c}_{ijt})(c_{ijt} - \hat{c}_{ijt})^\top] = \Sigma_{jt}^c$. Following [32], under the moment-based ambiguity set defined by (46), a constraint on the worst-case probability of violating the j^{th} constraint of obstacle $i \in \mathcal{B}$

$$\sup_{\mathbb{P}_{\mathcal{Z}_t} \in \mathcal{P}^{\mathcal{Z}}} \mathbb{P}_{\mathcal{Z}_t}(a_{ij}^\top C_{xr} \mathcal{Z}_t \geq a_{ij}^\top c_{ijt}) \leq \alpha_i \quad (49)$$

is equivalent to the linear constraint on the state mean $\hat{\mathcal{Z}}_t$

$$a_{ij}^\top C_{xr} \hat{\mathcal{Z}}_t \geq a_{ij}^\top \hat{c}_{ijt} + \sqrt{\frac{1 - \alpha_i}{\alpha_i}} \left\| (D_{\hat{x}_t} + \Sigma_{jt}^c)^{\frac{1}{2}} a_{ij} \right\|_2, \quad (50)$$

where $D_{\hat{x}_t} = C_{xr} \Sigma_{\mathcal{Z}_t} C_{xr}^\top$ and α_i is the user prescribed risk parameter for obstacle $i \in \mathcal{B}$. Obstacle risks are allocated such that their sum does not exceed the total constraint risk α (as described in subsection 6.2). The scaling constant $\sqrt{\frac{1 - \alpha_i}{\alpha_i}}$ in the deterministic tightening of the nominal constraint in (50) is larger than the one obtained with a Gaussian assumption, leading to a stronger tightening that reflects the weaker assumptions about the uncertainty distributions.

7. Simulation results

In this section, we demonstrate our proposed framework using simulations with both a unicycle and a bicycle robot dynamics. Specifically, we demonstrate the bicycle dynamics in an urban driving simulator [21]. The robot is assumed to be moving in a bounded and cluttered environment. While the proposed framework can handle both dynamic and uncertain obstacles, we assume the obstacles to be static and deterministic ($w_{\mathcal{O}_t} = 0, \forall t$) for simplicity, so that all uncertainty comes from the unknown initial state, robot process disturbance, and measurement noise. That is, we assume that $g_i(x) = 0, \forall i \in \mathcal{B}$.

7.1. Unicycle model based simulation

7.1.1. Robot motion model

We consider the problem of navigating a robot with unicycle dynamics from an initial state to a final set of states. The nonlinear robot and the static obstacle dynamics as shown in Fig. 5 are given by

$$\begin{bmatrix} x_{t+1} \\ y_{t+1} \\ \theta_{t+1} \\ x_{obs,t+1} \\ y_{obs,t+1} \end{bmatrix} = \underbrace{\begin{bmatrix} x_t \\ y_t \\ \theta_t \\ x_{obs,t} \\ y_{obs,t} \end{bmatrix}}_{\mathcal{Z}_t} + \Delta t \begin{bmatrix} v_t \cos(\theta_t) \\ v_t \sin(\theta_t) \\ \omega_t \\ 0 \\ 0 \end{bmatrix} + \Delta t w_t \quad (51)$$

where $x_t, y_t \in \mathbb{R}$ are the horizontal and vertical positions of the robot, $\theta_t \in \mathbb{R}$ is the heading of the robot relative to the x -axis with $v_t, \omega_t \in \mathbb{R}$ being the linear and angular velocity control inputs, and w_t denoting the disturbances at time t . Further, $x_{obs,t}, y_{obs,t} \in \mathbb{R}$ denote the horizontal and vertical positions of the obstacles at time t . The discretization time step was selected to be $\Delta t = 0.2$ sec.

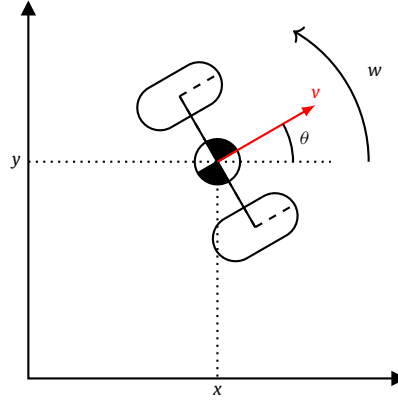


Fig. 5. A unicycle robot operating in $\mathbb{R}^2$ is shown here.

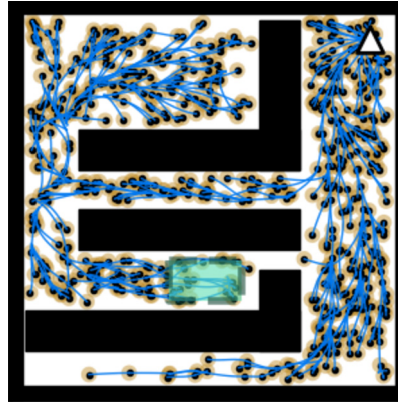


Fig. 6. A risk-bounded motion plan of a unicycle robot operating in $\mathbb{R}^2$ computed using the NRB-RRT* algorithm with stage-risk budget $\alpha_t = 10^{-4}$ and the corresponding 1σ uncertainty covariance ellipses for its positions are shown here in yellow. The white triangle and the green rectangle respectively denote the start position and the goal region.

7.1.2. Measurement model

We assume that the unicycle robot is equipped with a sensor with measurement noise. Further we consider independent sensor and process noises for simplicity. The measurement model is given by

$$z_t = \mathcal{S}(\mathcal{Z}_t) + v_t = \mathcal{Z}_t + v_t. \quad (52)$$

7.1.3. Discussion of results

The motion plan using NRB-RRT* algorithm was generated on a machine with an Intel Core i7 CPU and 8 GB of RAM. The trajectory tracking Monte Carlo simulations were performed on a machine with a Ryzen 7 2700X and 64 GB of RAM. The nonlinear MPC problem in (42) is modeled with CasADi Opti and solved with IPOPT solver [33]. We demonstrate NRB-RRT* in a obstacle cluttered environment as shown in Fig. 6. The environment consisted of a root node (white triangle), a goal area (dashed green rectangle), and a 10×10 environment with rectangular obstacles for its sides (black boundary). The robot is assumed to occupy a single point. The input bounds are ± 0.5 units/sec for linear velocity and $\pm \pi$ rad/sec for angular velocity. The NRB-RRT* steering horizon is $N = 30$ and the NMPC planning horizon is $N_{ll} = 10$. The planner control cost matrix was $R = \text{diag}([1, 1])$. The tracking cost matrices were $Q = \text{diag}([100, 100, 10])$ and $R = \text{diag}([1, 1])$ for all $t = [0 : T - 1]$, and $Q_T = 10Q$. We used a high-level plan risk bound of $\beta = 0.1$. It is divided equally across the time steps $T_{max} = 1000$ and among the obstacle constraints. The process noise distribution $\mathbb{P}_w$ and the sensor noise distribution $\mathbb{P}_v$ were taken as a multivariate Laplace distribution with zero mean and covariance $\Sigma_w = \Sigma_v = 10^{-7} I_n$. We employed NRB-RRT* algorithm with just the euclidean distance metric for generating the motion plan. Clearly, the NRB-RRT* tree as shown in Fig. 6 avoided the unsafe gap to the top-right of the goal region in process of reaching the goal as it was deemed to be too risky.

To evaluate the effectiveness of the NRB-RRT* algorithm, we performed Monte-Carlo simulation using the reference trajectory obtained after 1000 iterations of NRB-RRT* algorithm. Particularly, at each independent trials out of the total 1000 trials, we realized both the process noise w_t and the sensor noise v_t from either a multivariate Laplacian distribution

Table 1

Performance metrics from 1000 independent Monte Carlo trials with different covariance matrices $\Sigma_w = \Sigma_v$ for noises corresponding to two different distributions are tabulated here. It is evident the multivariate Laplacian being an heavy-tailed distribution results in more collisions than its counterpart multivariate Gaussian.

Noise Covariance $\Sigma_w = \Sigma_v$	Multivariate Laplacian		Multivariate Gaussian	
	# of collisions	Average Run Time (s)	# of collisions	Average Run Time (s)
$10^{-7}I_n$	0	4.2117	0	4.2043
$10^{-3}I_n$	551	3.5119	495	3.7024
$10^{-2}I_n$	1000	0.8964	1000	0.9310

Table 2

Performance metrics from 1000 independent Monte Carlo trials with and without risk assessment for a given covariance matrix $\Sigma_w = \Sigma_v$ of noises corresponding to multivariate Laplacian distribution are tabulated here. It is evident from Figs. 10, 11, 12 and the number of collisions that taking risk into consideration results in a longer but safer path while the approach without risk consideration results in a short but a risky path.

Noise Covariance $\Sigma_w = \Sigma_v$	# of collisions	
	With Risk	Without Risk
$10^{-7}I_n$	0	0
$10^{-5}I_n$	0	360

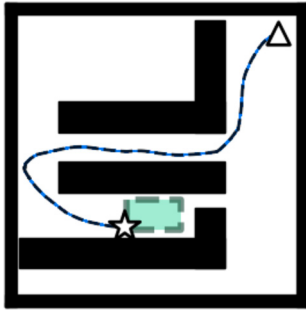


Fig. 7. Results of 1000 independent Monte-Carlo trials using zero-mean noises w_t, v_t sampled from multivariate Laplacian distribution with covariance matrices $\Sigma_w = \Sigma_v = 10^{-7}I_n$ are shown here. With noises being minimal, success rate is 100%. The white triangle, star and green rectangle denote the starting position, goal position and the goal region respectively.

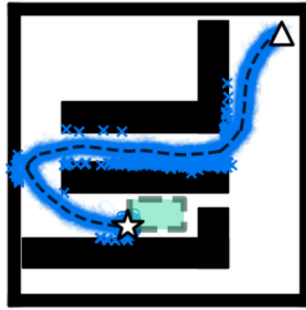


Fig. 8. Results of 1000 independent Monte-Carlo trials using zero-mean noises w_t, v_t sampled from multivariate Laplacian distribution with covariance matrices $\Sigma_w = \Sigma_v = 10^{-3}I_n$ are shown here. With noises being stronger, success rate falls to 44.9%. The white triangle, star and green rectangle denote the starting position, goal position and the goal region respectively.



Fig. 9. Results of 1000 independent Monte-Carlo trials using zero-mean noises w_t, v_t sampled from multivariate Laplacian distribution with covariance matrices $\Sigma_w = \Sigma_v = 10^{-2}I_n$ are shown here. With noises being strong, success rate falls to 0%. The white triangle, star and green rectangle denote the starting position, goal position and the goal region respectively.

or a multivariate Gaussian distribution with the same mean and covariance. The results of the Monte-Carlo trials using multivariate Laplacian noises with covariance matrices ($\Sigma_w = \Sigma_v$) for different noise levels ($10^{-7}I_n, 10^{-3}I_n, 10^{-2}I_n$) are shown in Figs. 7, 8, 9 respectively (Table 1). Analogous results using multivariate Gaussian distribution based noises for same noise levels were also obtained. The Monte-Carlo trials subjected to noises from both distributions with different noise levels reveal that as the noise levels were smaller ($\Sigma_w = \Sigma_v = 10^{-7}I_n$), the feedback control layer had good tracking of the reference trajectory provided by the planning layer. However, when the noises got stronger ($\Sigma_w = \Sigma_v = 10^{-3}I_n$), more collisions/failure occurred and finally with even stronger noises, ($\Sigma_w = \Sigma_v = 10^{-2}I_n$), no trial was capable to track the whole trajectory without collision. This demonstrates that as the planner in the autonomy stack plans a reference trajectory with a conservative noise setting, the controller can tolerate noises up to a certain level after which it cannot reach the goal set without collisions. A similar analysis with and without taking the risk into account is shown in Figs. 10, 11 and 12 and the results are tabulated in Table 2. Note that when the noises in the tracking phase get realized with the same moments that was assumed in the planning phase, then our proposed approach ensures that the worst case probability of failure in the tracking phase is bounded by $\beta = 0.1$. In the next subsection, we simulate a car-like robot having bicycle dynamics with a nonlinear measurement model and sensor noise and demonstrate our approach using an urban driving simulator.

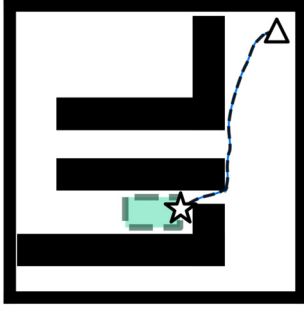


Fig. 10. The reference path (shown in dashed) generated by the path planning module from source to goal goes through the risky narrow opening when no risk is taken into account during the planning. With zero-mean noises w_t, v_t sampled from multivariate Laplacian distribution with covariance matrices $\Sigma_w = \Sigma_v = 10^{-7}I_n$, tracking of 1000 independent Monte-Carlo trials are overlayed in blue. With no collisions, the worst case failure probability is less than $\beta = 0.1$.

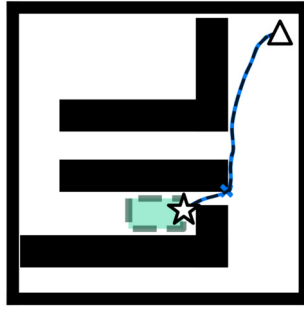


Fig. 11. Results of 1000 independent Monte-Carlo trials using zero-mean noises w_t, v_t sampled from multivariate Laplacian distribution with covariance matrices $\Sigma_w = \Sigma_v = 10^{-5}I_n$ are shown here. The success rate falls down to 64% as the reference path goes through the risky narrow opening when no risk is considered. Though the noise is very small, it was enough to cause 36% of the trajectories to collide and thereby showing the need for including the risk while planning.

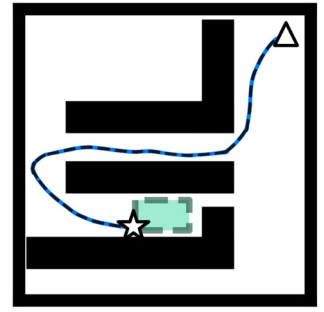


Fig. 12. Results of 1000 independent Monte-Carlo trials using zero-mean noises w_t, v_t sampled from multivariate Laplacian distribution with covariance matrices $\Sigma_w = \Sigma_v = 10^{-5}I_n$ are shown here. The success rate is still 100% for the reference path that uses the risk-based formulation. Unlike Fig. 11, the reference path with risk consideration avoids narrow openings to remain safe. The added conservatism of the reference path is partially a byproduct of the selection of T_{max} .

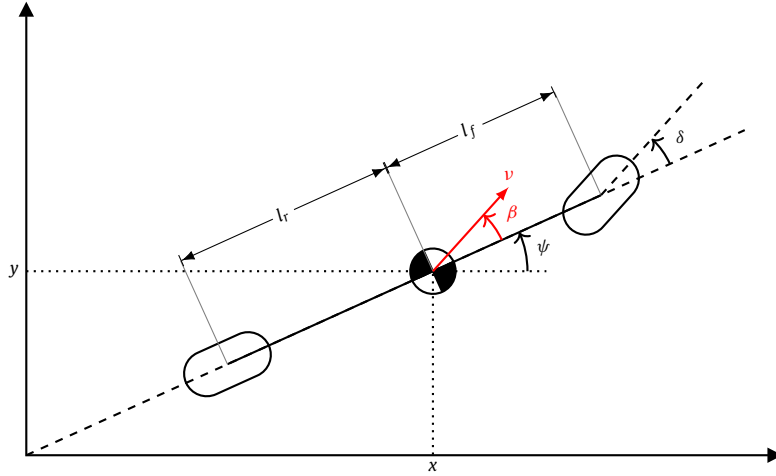


Fig. 13. A bicycle model [34] depicting the motion of a car-like vehicle is shown here.

7.2. Bicycle model based simulation

We consider a robotic car vehicle modeled using bicycle dynamics traversing in an uncertain environment with an obstacle. For the ease of exposition, we consider only one obstacle. To demonstrate our proposed approach, we perform a simulation with static obstacle first and then subsequently we present the results with a dynamic obstacle.

7.2.1. Robot motion model

The kinematics of the robot modeled using bicycle dynamics as shown in Fig. 13 and the static obstacle dynamics are given by

$$\begin{bmatrix} x_{t+1} \\ y_{t+1} \\ \psi_{t+1} \\ v_{t+1} \\ x_{obs,t+1} \\ y_{obs,t+1} \end{bmatrix} = \underbrace{\begin{bmatrix} x_t \\ y_t \\ \psi_t \\ v_t \\ x_{obs,t} \\ y_{obs,t} \end{bmatrix}}_{\mathcal{Z}_t} + \Delta t \begin{bmatrix} v_t \cos(\psi_t) \\ v_t \sin(\psi_t) \\ \frac{v_t}{L} \tan(\delta_t) \\ a_t \\ 0 \\ 0 \end{bmatrix} + \Delta t w_t, \quad (53)$$

where (x_t, y_t) denote the position of center of mass of the robot in the simulation environment, and ψ_t its orientation at time t . The pair $(x_{obs,t}, y_{obs,t})$ denote the position of center of mass of the obstacle at time t . The discretization time step was selected to be $\Delta t = 0.2$. The length of the vehicle wheelbase is denoted by L and $L = l_f + l_r$, where l_f, l_r denote the distance from the center of mass to the front and rear wheels respectively. The angle of the current velocity v_t of the center of mass with respect to the longitudinal axis of the car is denoted by β_t . The control inputs (a_t, δ_t) respectively denote the linear acceleration and the steering angle of the robot and acceleration of the center of mass is assumed to be in the same direction as the velocity v_t . We assume the robot to start from the origin with an orientation $(\psi_0 = 270^\circ$ in Carla). The pose of the robot is sampled uniformly within the bounds of the feasible environment in $\mathbb{R}^2$ whose boundaries are not treated probabilistically. We chose to simulate our model using the Audi E-Tron vehicle in the simulator whose parameter values are $L = 2.9m$. For the dynamic obstacle based simulations, we assumed the obstacle vehicle to move in a straight-line (along the y direction) with two different constant velocities of $v_{obs} \in \{[0, 0.10]^T, [0, 0.20]^T\}$.

7.2.2. Measurement model

We assume that the robot is equipped with a radar sensor which provides a noisy bearing and range to multiple known obstacle locations in the landscape. Thus the measurement model is given by

$$z_t = \mathcal{S}(x_t, y_t, \theta^*, x_{obs,t}, y_{obs,t}) = \begin{bmatrix} \sqrt{(x_t - x_{goal})^2 + (y_t - y_{goal})^2} \\ \tan^{-1}\left(\frac{y_t - y_{goal}}{x_t - x_{goal}}\right) - \psi_t \\ \cos(\theta^*)x_{obs,t} \\ \sin(\theta^*)y_{obs,t} \end{bmatrix} + v_t, \quad (54)$$

where the point (x_{goal}, y_{goal}) denotes the location of a landmark inside the goal region $\mathcal{X}_{goal}$ and $\theta^* = 0.01$ radian denotes a known distortion of a sensor that estimates the position of obstacles. The initial state is assumed to be of zero mean with covariance matrix $\hat{\Sigma}_{\mathcal{Z}_0} = 0.001I_{n_z}$. The sensor noise and process noise covariance matrices were $\Sigma_v = 0.001I_p$ and $\Sigma_w = \text{diag}(0.001, 0.001, 0.001, 0.001, 0, 0)$. For simplicity, we assumed the cross-correlation matrix to be zero ($M = 0$). While executing the Algorithm 1, the equation (44) is used to approximate the cost-to-go quantity with $k_\phi = 1.2$ and $k_\delta = 3$ respectively in the NearestNode($\cdot$) and NearNodes($\cdot$) modules and for the rest, the cost-to-go is approximated using the steering law that employs nonlinear model predictive control. The search radius used in the Algorithm 1 is obtained with $\gamma = 30$. For simplicity, the environmental boundaries are not treated probabilistically. Since the motion model given by (53) and the measurement model given by (54) are nonlinear, this particular set up demands using Unscented Kalman Filter for state estimation and a nonlinear model predictive control for steering. To generate the sigma points and weigh them accordingly, we use $\alpha_u = 1, \beta = 2, \kappa = 3 - n$. Further, we define the error $e_t = C_{xr} \hat{\mathcal{Z}}_t - x_s$ for $i = 0, \dots, T$, with $T = 1000$. A dynamic output feedback policy u_t that minimizes the cost function

$$J = \sum_{k=t}^{t+N-1} \left(e_k^\top Q e_k + u_k^\top R u_k \right) + e_{k+N}^\top Q e_{k+N}, \quad (55)$$

is computed using nonlinear model predictive control to steer the robot from a tree node state x_t to a random feasible sample pose x_s with prediction horizon $N = 50$. The control inputs are constrained as $|a| \leq 3ms^{-2}$ and $|\delta| \leq 70^\circ$. The nonlinear model predictive control law is obtained through the CasADi toolbox [33] that employs the IPOPT large-scale nonlinear optimization solver. The state and control penalty matrices namely $Q = 100\text{diag}([2, 2, 1, 5, 1, 1])$, $R = 0.01I_m$ are used to penalize the state and control deviations respectively. The distributionally robust state constraints are enforced over a mission horizon length $T = 50$ and with probabilistic satisfaction parameter $\alpha = 0.05$. For all the simulations, the NRB-RRT* algorithm with the distributionally robust collision check or Gaussian collision check procedure is run for 200 iterations, with $2\text{-}\sigma$ position uncertainty ellipses from the covariance matrix being drawn at the end of each trajectory.

7.2.3. Results and discussion

Static Obstacle Simulation: The NRB-RRT* tree expansion procedure was run until at most 2 nodes from the goal region were added to the tree successfully and the resulting reference trajectory from the starting pose to the goal is shown in Fig. 14. The corresponding 2D projection is shown in the left side of Fig. 15 and distributionally robust risk constraints generated more conservative trajectories around the obstacles, by explicitly incorporating the uncertainty in the state due to the initial localization, system dynamics, and measurement uncertainties in the form of ambiguity sets. It produces trajectories that satisfy the chance constraints under the worst-case distribution in the ambiguity sets. Clearly, the feasible set is smaller with the distributionally robust constraints. On the other hand, a similar reference trajectory was generated using Gaussian chance constraints assuming system uncertainties are Gaussian. The reference trajectories generated using DR risk constraints (with UKF and EKF), and Gaussian chance constraints are shown in the center of Fig. 15. The distributionally robust trajectories with a more sophisticated and coherent quantification of risk, are generated with the same computational complexity as with Gaussian chance constraints and exhibit conservatism to account for state distribution ambiguity. Though the result with Extended Kalman Filter based state estimation also produces collision free trajectories, the result inherently

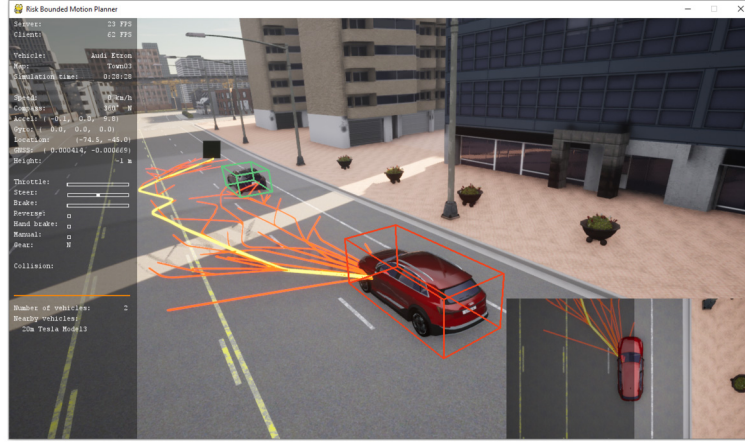


Fig. 14. Our proposed approach is demonstrated using an urban driving simulator [21]. An NRB-RRT* tree (in orange color) as a result of 200 iterations and the resulting reference path (in yellow color) from the start to the goal region are shown here. The NRB-RRT* algorithm employed distributionally robust chance constraints with $\alpha = 0.05$.

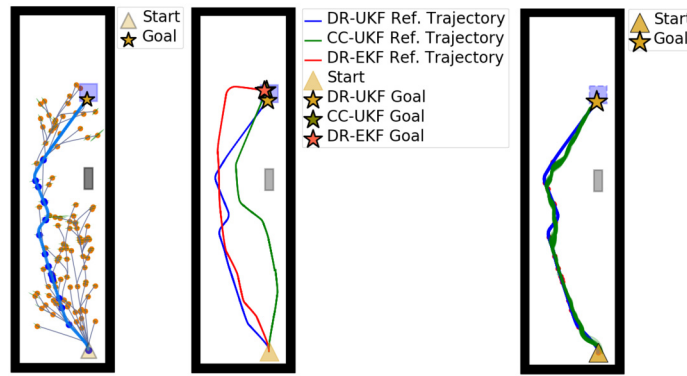


Fig. 15. A 2-D projection of the urban driving environment containing the robot starting pose, goal region (blue shaded rectangle), the obstacle (black shaded rectangle), constructed motion plan with distributionally robust chance constraints (in blue color) after executing 200 iterations of NRB-RRT* algorithm with the two-standard deviation covariance ellipses and the generated reference path from the source to the goal are shown on the plot to the left. A comparison of NRB-RRT* algorithm with distributionally robust (DR) (in blue color (UKF) and red color (EKF)) & Gaussian chance constraints (CC) (in green color (UKF)) are shown on the middle. Clearly, the added conservatism greater than that of the chance constrained counterpart that comes due to the distributional robustness would account for any arbitrary unknown state distribution satisfying the given moments. The results of 100 independent Monte Carlo trials of DR-UKF trajectory tracking with noises drawn from multivariate Gaussian distributions are shown on the right.

suffers from the (first order) approximations due to linearization and can possibly lead to significant miscalculation of risk. To corroborate the results generated by the high-level motion planner, a Monte-Carlo simulation of 100 independent trials involving noises sampled from multivariate Gaussian distribution was conducted and the results are shown in the right side of Fig. 15. It is evident that given the conservative noise assumption and risk quantification performed using the distributionally robust approach, the realized sampled paths from the source to destination had little deviations around the reference trajectory. This demonstrates our proposed approach that with a high-level motion plan that accounts for uncertainty, the low level online tracking controller has more room to maneuver and respond to noise realizations. Given a noise level and limited number of NRB-RRT* iterations, the chance constrained version might lead to a less conservative, shorter yet riskier path than the path obtained from the distributionally robust version that while possibly longer, remains comparatively safer. That is, conservatism is inherently traded off with risk guarantees under the worst case uncertainty distribution. Note that the tight probability of collision can be shown only with infinite number of NRB-RRT* iterations once the least possible horizon T_{max} is fixed a priori.

Dynamic Obstacle Simulation: The simulation results with the dynamic obstacle are presented in Fig. 16. Two separate motion plans were incrementally built with the obstacle moving in a straight line with constant velocity of $v_{obs} \in \{[0, 0.10]^T, [0, 0.20]^T\}$. Since the reference trajectory was built by taking into account the motion of the dynamic obstacle, we observe that overtaking or following the obstacle vehicle depends upon both the risk budget and the magnitude of the obstacle velocity. This shows that our framework is capable of handling dynamic obstacles by keeping track of the time-stamp of the obstacle motion and performing the probabilistic collision check with the time-varying obstacle location.

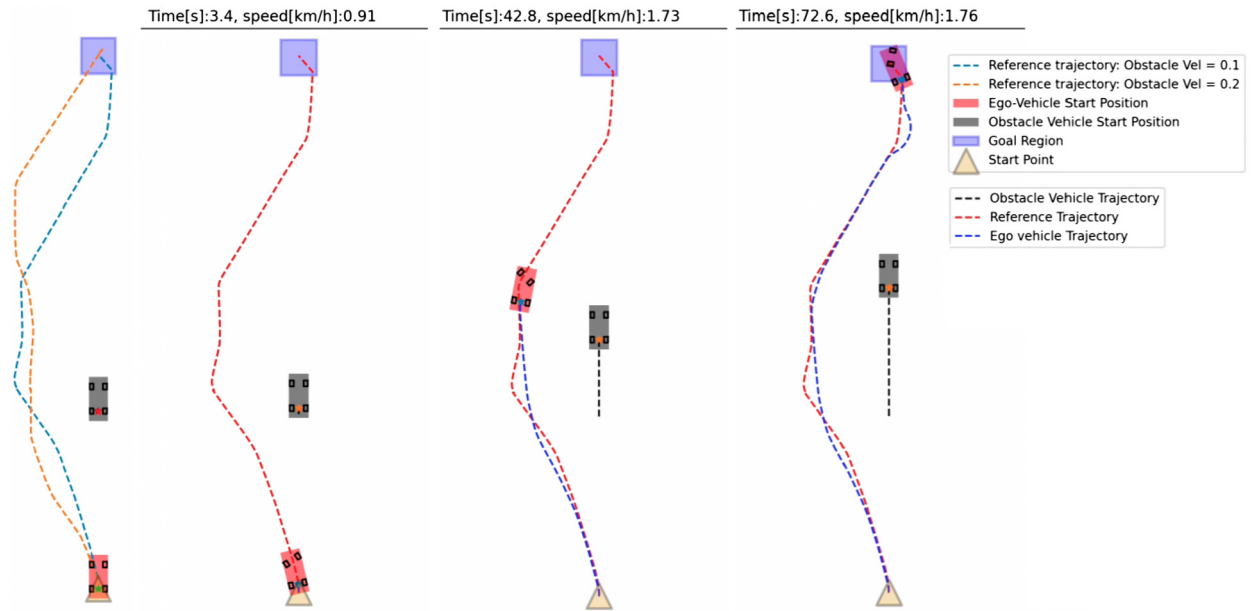


Fig. 16. The constructed reference path with distributionally robust chance constraints using NRB-RRT* algorithm from the source to the goal projected onto $\mathbb{R}^2$ with two different constant velocities for the dynamic obstacle is shown on the left plot. The next three plots on the right show snap shots of the robot overtaking an obstacle with velocity $v_{obs} = [0, 0.1]^T$.

The video of the both static and dynamic obstacle simulations is made available at <https://youtu.be/YyquJ1Hh-6A> and the simulation code is available at https://github.com/TSummersLab/Risk_Bounded_Nonlinear_Robot_Motion_Planning.

8. Conclusion

In this paper, we presented a framework aimed towards tighter integration of perception and planning in nonlinear robotic systems. Uncertainties in perception and motion prediction are explicitly accounted for through distributionally robust risk constraints. Using a dynamic output feedback based control policy realized using a nonlinear MPC and an Unscented Kalman filter, a new algorithm called NRB-RRT* is shown to produce risk bounded trajectories with systematic risk assessment. Future research involves using deep learning based perception system with semantic SLAM combined with our approach. We will also explore optimal spatio-temporal risk allocation for obstacles and environmental constraints which can result less conservative motion plans for robots operating in cluttered environments. Another direction involves studying propagation of state distributions of nonlinear systems with higher order moments and using them to estimate the risk.

Supplementary material

Video of the simulation is available at <https://youtu.be/YyquJ1Hh-6A> and the simulation code is available at https://github.com/TSummersLab/Risk_Bounded_Nonlinear_Robot_Motion_Planning.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgement

This work is partially supported by Defence Science and Technology Group, through agreement MyIP: ID10266 entitled “**Hierarchical Verification of Autonomy Architectures**”, the Australian Government, via grant AUSMURIB000001 associated with ONR MURI grant N00014-19-1-2571, the United States Air Force Office of Scientific Research under award number FA2386-19-1-4073, and the National Science Foundation under grant ECCS-2047040.

References

- [1] L. Blackmore, H. Li, B. Williams, A probabilistic approach to optimal robust path planning with obstacles, in: 2006 American Control Conference, IEEE, 2006, 7 pp.
- [2] A.-A. Agha-Mohammadi, S. Chakravorty, N.M. Amato, Firm: sampling-based feedback motion-planning under motion uncertainty and imperfect measurements, *Int. J. Robot. Res.* 33 (2) (2014) 268–304.
- [3] B. Luders, M. Kothari, J. How, Chance constrained RRT for probabilistic robustness to environmental uncertainty, in: AIAA Guidance, Navigation, and Control Conference, 2010, p. 8160.
- [4] B.D. Luders, S. Karaman, J.P. How, Robust sampling-based motion planning with asymptotic optimality guarantees, in: AIAA Guidance, Navigation, and Control (GNC) Conference, 2013, p. 5097.
- [5] L. Blackmore, M. Ono, B.C. Williams, Chance-constrained optimal path planning with obstacles, *IEEE Trans. Robot.* 27 (6) (2011) 1080–1094.
- [6] W. Liu, M.H. Ang, Incremental sampling-based algorithm for risk-aware planning under motion uncertainty, in: 2014 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2014, pp. 2051–2058.
- [7] H. Zhu, J. Alonso-Mora, Chance-constrained collision avoidance for MAVs in dynamic environments, *IEEE Robot. Autom. Lett.* 4 (2) (2019) 776–783.
- [8] R.T. Rockafellar, Coherent approaches to risk in optimization under uncertainty, in: *OR Tools and Applications: Glimpses of Future Technologies*, Informa, 2007, pp. 38–61.
- [9] A. Majumdar, M. Pavone, How should a robot assess risk? Towards an axiomatic theory of risk in robotics, in: *Robotics Research*, Springer, 2020, pp. 75–84.
- [10] J. Goh, M. Sim, Distributionally robust optimization and its tractable approximations, *Oper. Res.* 58 (4-part-1) (2010) 902–917.
- [11] P. Florence, J. Carter, R. Tedrake, Integrated perception and control at high speed: evaluating collision avoidance maneuvers without maps, in: *Workshop on the Algorithmic Foundations of Robotics (WAFR)*, 2016.
- [12] 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.
- [13] S. Thiébaux, B. Williams, et al., RAO*: an algorithm for chance-constrained POMDP's, in: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, 2016.
- [14] J. Pilipovsky, P. Tsiotras, Chance-constrained optimal covariance steering with iterative risk allocation, in: 2021 American Control Conference (ACC), IEEE, 2021, pp. 2011–2016.
- [15] A. Bry, N. Roy, Rapidly-exploring random belief trees for motion planning under uncertainty, in: 2011 IEEE International Conference on Robotics and Automation, IEEE, 2011, pp. 723–730.
- [16] Y. Wang, A.A.R. Newaz, J.D. Hernández, S. Chaudhuri, L.E. Kavraki, Online partial conditional plan synthesis for POMDPs with safe-reachability objectives: methods and experiments, *IEEE Trans. Autom. Sci. Eng.* 18 (3) (2021) 932–945.
- [17] G. Costante, C. Forster, J. Delmerico, P. Valigi, D. Scaramuzza, Perception-aware path planning, *arXiv preprint*, arXiv:1605.04151, 2016.
- [18] V. Renganathan, I. Shames, T.H. Summers, Towards integrated perception and motion planning with distributionally robust risk constraints, in: 21st IFAC World Congress, *IFAC-PapersOnLine* 53 (2) (2020) 15530–15536, <https://doi.org/10.1016/j.ifacol.2020.12.2396>.
- [19] A. Hakobyan, I. Yang, Wasserstein distributionally robust motion control for collision avoidance using conditional value-at-risk, *IEEE Trans. Robot.* (2021).
- [20] S. Safaoui, B.J. Gravell, V. Renganathan, T.H. Summers, Risk-averse RRT* planning with nonlinear steering and tracking controllers for nonlinear robotic systems under uncertainty, in: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2021, pp. 3681–3688.
- [21] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, V. Koltun, Carla: an open urban driving simulator, in: *Conference on Robot Learning*, in: PMLR, 2017, pp. 1–16.
- [22] N. Sünderhauf, T.T. Pham, Y. Latif, M. Milford, I. Reid, Meaningful maps with object-oriented semantic mapping, in: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2017, pp. 5079–5085.
- [23] S. Wang, G. Qi, L. Wang, High degree cubature Kalman filters for nonlinear systems with correlated noises, in: 2015 7th International Conference on Intelligent Human-Machine Systems and Cybernetics, vol. 1, IEEE, 2015, pp. 57–61.
- [24] D. Gu, H. Hu, A stabilizing receding horizon regulator for nonholonomic mobile robots, *IEEE Trans. Robot.* 21 (5) (2005) 1022–1028.
- [25] R. Findeisen, L. Imsland, F. Allgower, B.A. Foss, State and output feedback nonlinear model predictive control: an overview, *Eur. J. Control* 9 (2) (2003) 190–206, <https://doi.org/10.3166/ejc.9.190-206>.
- [26] G. Chang, Marginal unscented Kalman filter for cross-correlated process and observation noise at the same epoch, *IET Radar Sonar Navig.* 8 (1) (2014) 54–64.
- [27] E.A. Wan, R. Van Der Merwe, The unscented Kalman filter for nonlinear estimation, in: *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No. 00EX373)*, IEEE, 2000, pp. 153–158.
- [28] X.-S. Trinh, D. Ngoduy, M. Keyvan-Ekbatani, B. Robertson, Incremental unscented Kalman filter for real-time traffic estimation on motorways using multi-source data, *Transportmetrica A: Transp. Sci.* (2021) 1–27.
- [29] E. Frazzoli, M.A. Dahleh, E. Feron, Real-time motion planning for agile autonomous vehicles, *J. Guid. Control Dyn.* 25 (1) (2002) 116–129.
- [30] J.J. Park, B. Kuipers, Feedback motion planning via non-holonomic RRT* for mobile robots, in: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2015, pp. 4035–4040.
- [31] K. Solovey, L. Janson, E. Schmerling, E. Frazzoli, M. Pavone, Revisiting the asymptotic optimality of RRT*, in: 2020 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2020, pp. 2189–2195.
- [32] G.C. Calafiore, L. El Ghaoui, On distributionally robust chance-constrained linear programs, *J. Optim. Theory Appl.* 130 (1) (Dec. 2006).
- [33] J.A.E. Andersson, J. Gillis, G. Horn, J.B. Rawlings, M. Diehl, CasADi – a software framework for nonlinear optimization and optimal control, *Math. Program. Comput.* 11 (1) (2019) 1–36, <https://doi.org/10.1007/s12532-018-0139-4>.
- [34] J. Kong, M. Pfeiffer, G. Schildbach, F. Borrelli, Kinematic and dynamic vehicle models for autonomous driving control design, in: 2015 IEEE Intelligent Vehicles Symposium (IV), IEEE, 2015, pp. 1094–1099.