

Probabilistic Feasibility for Nonlinear Systems with Non-Gaussian Uncertainty using RRT

Brandon Luders* and Jonathan P. How†

Aerospace Controls Laboratory

Massachusetts Institute of Technology, Cambridge, MA

{luders,jhow}@mit.edu

For motion planning problems involving many or unbounded forms of uncertainty, it may not be possible to identify a path guaranteed to be feasible, requiring consideration of the trade-off between planner conservatism and the risk of infeasibility. Recent work developed the chance constrained rapidly-exploring random tree (CC-RRT) algorithm, a real-time planning algorithm which can efficiently compute risk at each timestep in order to guarantee probabilistic feasibility. However, the results in that paper require the dual assumptions of a linear system and Gaussian uncertainty, two assumptions which are often not applicable to many real-life path planning scenarios. This paper presents several extensions to the CC-RRT framework which allow these assumptions to be relaxed. For nonlinear systems subject to Gaussian process noise, state distributions can be approximated as Gaussian by considering a linearization of the dynamics at each timestep; simulation results demonstrate the effectiveness of this approach for both open-loop and closed-loop dynamics. For systems subject to non-Gaussian uncertainty, we propose a particle-based representation of the uncertainty, and thus the state distributions; as the number of particles increases, the particles approach the true uncertainty. A key aspect of this approach relative to previous work is the consideration of probabilistic bounds on constraint satisfaction, both at every timestep and over the duration of entire paths.

I. Introduction

A major research focus area of current interest to the motion planning community is the development of verification and validation tools for autonomous systems, subject to complex forms of autonomy and uncertainty.⁴ While guaranteed safety is always ideal, often the quantity or degree of uncertainty is sufficiently high that a guaranteed-feasible path does not exist. In this case, chance constraints¹⁴ can provide an elegant way to model the trade-off between planner conservatism and the risk of path infeasibility.

The recently-proposed chance-constrained rapidly-exploring random trees (CC-RRT) algorithm leverages chance constraints² to guarantee probabilistic feasibility for complex motion planning problems, involving both process noise and environmental uncertainty.¹⁶ By using RRT,¹¹ the algorithm enjoys the computational benefits of sampling-based algorithms while explicitly incorporating uncertainty. Though this algorithm requires only a minimal computational increase, it relies on the assumptions of linear systems and Gaussian uncertainty, restricting the applicability of this approach. Furthermore, probabilistic feasibility can only be guaranteed at each *timestep*, when feasibility estimates over a *path* are often more useful.

This paper extends the CC-RRT algorithm to consider nonlinear systems and/or systems subject to non-Gaussian uncertainty. The resulting algorithms approximate the probabilistic feasibility metrics established for linear Gaussian systems, allowing for their use on more complex dynamics and uncertainty formulations, and can also provide an estimate of path probabilistic feasibility. A novel contribution of this approach is its applicability to both open-loop and closed-loop dynamics,⁹ as well as environmental uncertainty. Simulation results demonstrate the viability of these extensions in assessing probabilistic feasibility for complex dynamics and/or uncertainty.

*Research Assistant, Department of Aeronautics and Astronautics, MIT; Member AIAA

†Richard C. Maclaurin Professor of Aeronautics and Astronautics, MIT; Associate Fellow AIAA

For nonlinear systems subject to Gaussian process noise, state distributions can be approximated as Gaussian by considering a linearization of the dynamics at each timestep. Since the linearized dynamics depend on the current vehicle state, the previously considered offline CC-RRT formulation is no longer applicable, but online CC-RRT can still be used.¹⁶ For closed-loop RRT, the reference feedback law must also be linearized, as demonstrated for a skid-steered vehicle using pure pursuit steering.

For systems subject to non-Gaussian uncertainty, the above approach may not approximate the state distributions sufficiently well. In this case, we propose a particle-based representation of the uncertainty, and thus the state distributions; as the number of particles increases, the particles approach the true uncertainty. Probabilistic feasibility is then measured in terms of the number of feasible particles at each node, compared to all particles considered at that node or along the path. While similar approaches have been proposed for chance constraints¹ and sampling-based algorithms,¹⁷ our algorithm enjoys several key advantages for assessing probabilistic feasibility. By not clustering particles, the one-to-one mapping between inputs and nodes is maintained. A fixed number of particles are propagated at each node, by probabilistically resampling the feasible particles to replace any particles found infeasible. In this manner, high-resolution feasibility estimates are maintained for every node and path. In the paper, both uniform resampling of particles and weighted resampling based on the likelihood of each particle are considered; while the former is both faster and numerically stable, the latter better represents the true likelihood of feasibility at each node, with fewer samples. Finally, two estimates of probability are assessed at each node: the likelihood of feasibility at each timestep, and the overall likelihood of feasibility over the entire path. This extends the existing chance constraint framework, which is restricted to considering likelihood at each timestep, while enabling the use of either component as part of hard or soft constraints on the system. Simulation results demonstrate the ability of this approach to efficiently impose probabilistic constraints for challenging uncertainty environments.

This paper is organized as follows. Sections II and III set up the preliminaries for this paper, with the former establishing the problem statement and the latter reviewing the background work for this paper, chance constraints and the CC-RRT algorithm. The difference between offline CC-RRT and online CC-RRT is also reviewed, as the former will not always be applicable when the assumptions of a linear system and/or Gaussian uncertainty are removed. Section IV considers the implications of allowing nonlinear dynamics, while Section V introduces a particle-based framework which handles the possibility of non-Gaussian uncertainty. Section VI presents some representative simulation results, while Section VII offers concluding remarks and suggestions of future work.

A. Related Work

This paper considers extensions of the CC-RRT algorithm to achieve probabilistic robustness to two main types of uncertainty, often labeled model uncertainty and environmental sensing uncertainty.¹³ Many approaches have been considered to address the problem of motion planning under uncertainty,^{12,24} several of which are discussed below.

One large class of approaches seek to identify optimal policies in uncertain, possibly partially-observable environments, such as POMDPs^{5,23} and the Gaussian overlap A* framework of [6]. These optimal-policy approaches tend to scale poorly for problems with high-dimensional configuration spaces, hindering their ability to model complex dynamics and constraints. However, we call particular attention to the work of Prentice and Roy,²² which constructs a roadmap in the belief state for a linear system subject to Gaussian noise. The belief roadmap is primarily used to identify paths which maintain strong localization, rather than avoid uncertain obstacles. Though the propagation of the system mean and covariance is quite similar to the formulation proposed in the original CC-RRT work,¹⁶ the roadmap dictates the use of kinematic planning.

Much existing work on probabilistic uncertainty has focused on environmental sensing uncertainty, using techniques such as potential fields¹⁸ and probabilistic roadmaps,³ often representing obstacles with uncertain shapes.^{7,19} However, these approaches are not applicable to nonholonomic systems, which generally cannot track the piecewise linear roadmap paths, and require a significant preprocessing phase for roadmap construction. The CC-RRT algorithm extended in this work requires little to no preprocessing and generates dynamically feasible paths by construction. More recent work has focused on the problem of randomized planning subject to configuration uncertainty. Pepy and Lambert²¹ addresses the ego-sensing problem through use of an extended Kalman filter and simulated localization feedback, assuming a perfect system model. Pepy *et al.*²⁰ seeks guaranteed robust feasibility for a nonlinear system subject to bounded state uncertainty. However, the approach uses a conservative box-shaped “wrapper” to approximate and bound the reachable set at each node. However, the use of an extended Kalman filter is closely related to the

linearization techniques applied here to remove the assumption of linear dynamics.

Finally, several statistical RRT approaches have been proposed to model vehicles traveling on uneven terrain. The well-known particle-based approach of Melchior and Simmons¹⁷ samples each tree branch multiple times, using clustering to create nodes. The approach considered in this paper for non-Gaussian uncertainty differs from this approach in a few key ways. By not clustering particles, a one-to-one mapping is maintained between inputs and nodes, simplifying the control process. Furthermore, the approach considered here probabilistically resamples feasible particles to replace any particles found infeasible; thus, without resampling, the resolution on the probabilistic feasibility is uniform across all nodes. Furthermore, our work considers two assessments of probabilistic feasibility - at each timestep and over entire paths - both as soft constraints and hard constraints (Section II). We also note an algorithm which identifies a finite-series approximation of the uncertainty propagation, in order to reduce model complexity and the resulting number of simulations needed per node.⁸

II. Problem Statement

Consider the discrete-time system with process noise and uncertain localization,

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

$$x_0 \sim \mathbb{P}(x_0), \quad (2)$$

$$w_t \sim \mathbb{P}(w_t), \quad (3)$$

where $x_t \in \mathbb{R}^{n_x}$ is the state vector, $u_t \in \mathbb{R}^{n_u}$ is the input vector, and $w_t \in \mathbb{R}^{n_x}$ is a disturbance vector acting on the system. Here $\mathbb{P}(a)$ represents a random variable whose probability distribution is known. The disturbance w_t is unknown at current and future time steps, but has the known probability distribution (3).

The system itself is subject to two forms of uncertainty. Equation (2) represents uncertainty in the initial state x_0 , corresponding to an uncertain initial localization. Equation (3) represents a possibly non-Gaussian process noise, in the form of independent and identically distributed random variables w_t ($\mathbb{P}(w_t) \equiv \mathbb{P}(w) \forall t$). This noise may correspond to model uncertainty, external disturbances, and/or other factors.

The objective of this paper is to obtain results on probabilistic feasibility which are valid (or approximately valid) even if the dynamics remain nonlinear, and/or the uncertainty remains non-Gaussian. However, there will be times where one or both assumptions (linear dynamics, Gaussian uncertainty) remain in place, such as when introducing the existing CC-RRT algorithm in Section III. Under the assumption of linear dynamics, (1) takes the linear time-invariant form

$$x_{t+1} = Ax_t + Bu_t + w_t. \quad (4)$$

Under the assumption of Gaussian uncertainty, (2)-(3) take the form

$$x_0 \sim \mathcal{N}(\hat{x}_0, P_{x_0}), \quad (5)$$

$$w_t \sim \mathcal{N}(0, P_{w_t}), \quad (6)$$

where $\mathcal{N}(\hat{a}, P_a)$ represents a random variable whose probability distribution is Gaussian with mean \hat{a} and covariance P_a .

There are also constraints acting on the system state and input. These constraints are assumed to take the form

$$x_t \in \mathcal{X}_t \equiv \mathcal{X} - \mathcal{X}_{t1} - \dots - \mathcal{X}_{tB}, \quad (7)$$

$$u_t \in \mathcal{U}, \quad (8)$$

where $\mathcal{X}, \mathcal{X}_{t1}, \dots, \mathcal{X}_{tB} \subset \mathbb{R}^{n_x}$ are convex polyhedra, $\mathcal{U} \subset \mathbb{R}^{n_u}$, and the $-$ operator denotes set subtraction. The set \mathcal{X} defines a set of time-invariant convex constraints acting on the state, while $\mathcal{X}_{t1}, \dots, \mathcal{X}_{tB}$ represent B convex, polytopic obstacles to be avoided. The time dependence of \mathcal{X}_t allows the inclusion of both static and dynamic obstacles. For each obstacle, the shape and orientation are assumed to be known, while the placement is uncertain. This is represented as

$$\mathcal{X}_{tj} = \mathcal{X}_j^0 + c_{tj}, \quad \forall j \in \mathbb{Z}_{1,B}, \quad \forall t \quad (9)$$

$$c_{tj} \sim \mathbb{P}(c_{tj}) \quad \forall j \in \mathbb{Z}_{1,B}, \quad \forall t \quad (10)$$

where the $+$ operator denotes set translation and $\mathbb{Z}_{a,b}$ represents the set of integers between a and b inclusive. In this model, $\mathcal{X}_j^0 \subset \mathbb{R}^{n_x}$ is a convex polyhedron of known, fixed shape, while $c_{tj} \in \mathbb{R}^{n_x}$ is a possibly time-varying translation. This can be used to represent dynamic obstacles whose future state distributions are known, such as other vehicles with known intentions. Note that if the obstacle uncertainty is also assumed to be Gaussian, then (10) takes the form

$$c_{tj} \sim \mathcal{N}(\hat{c}_{jt}, P_{c_{jt}}) \quad \forall j \in \mathbb{Z}_{1,B}, \quad \forall t. \quad (11)$$

The primary objective of the planning problem is to reach the goal region $\mathcal{X}_{\text{goal}} \subset \mathbb{R}^{n_x}$ in minimum time,

$$t_{\text{goal}} = \inf\{t \in \mathbb{Z}_{0,t_f} \mid x_t \in \mathcal{X}_{\text{goal}}\}, \quad (12)$$

while ensuring the constraints (7)-(8) are satisfied at each time step $t \in \{0, \dots, t_{\text{goal}}\}$ with probability of at least p_{safe} . In practice, since there is uncertainty in the state, we assume it is sufficient for the distribution mean to reach the goal region $\mathcal{X}_{\text{goal}}$. A secondary objective may be to avoid some undesirable behaviors, such as proximity to constraint boundaries, and can be represented through a penalty function $\psi(x_t, \mathcal{X}_t, \mathcal{U})$.

Problem 1. Given the initial state distribution (\hat{x}_0, P_{x_0}) and constraint sets \mathcal{X}_t and \mathcal{U} , compute the input control sequence u_t , $t \in \mathbb{Z}_{0,t_f}$, $t_f \in \mathbb{Z}_{0,\infty}$ that minimizes

$$J(\mathbf{u}) = t_{\text{goal}} + \sum_{t=0}^{t_{\text{goal}}} \psi(x_t, \mathcal{X}_t, \mathcal{U}) \quad (13)$$

while satisfying (1) for all time steps $t \in \{0, \dots, t_{\text{goal}}\}$, and satisfying (7)-(8) at each time step $t \in \{0, \dots, t_{\text{goal}}\}$ with probability of at least p_{safe} .

If an estimate for the probability of constraint satisfaction over the entire path is available (Section V), then this may be incorporated. Also note that ψ can be designed to incorporate soft constraints on probability of constraint satisfaction over each timestep or path, if desired.

III. The CC-RRT Algorithm

This section reviews the previously-developed CC-RRT algorithm,¹⁶ first by developing the chance constraint formulation,² then incorporating it within the RRT algorithm.¹¹

A. Chance Constraints

Note that here and in the remainder of this section, the assumptions of linear dynamics and Gaussian uncertainty remain in place; the implications of removing these assumptions are considered in subsequent sections.

Given a sequence of inputs u_0, \dots, u_{N-1} , under the assumptions of linear dynamics and Gaussian uncertainty, the distribution of the state x_t (represented as the random variable X_t) can be shown to be Gaussian:²

$$P(X_t | u_0, \dots, u_{N-1}) \sim \mathcal{N}(\hat{x}_t, P_{x_t}) \quad \forall t \in \mathbb{Z}_{0,N},$$

where N is some time step horizon. The mean \hat{x}_t and covariance P_{x_t} can be updated implicitly using the relations

$$\hat{x}_{t+1} = A\hat{x}_t + Bu_t \quad \forall t \in \mathbb{Z}_{0,N-1}, \quad (14)$$

$$P_{x_{t+1}} = AP_{x_t}A^T + P_w \quad \forall t \in \mathbb{Z}_{0,N-1}. \quad (15)$$

The key advantages of the chance constraint approach for linear systems and Gaussian noise are made clear at this point: (14) updates the distribution mean \hat{x}_t using the nominal, disturbance-free dynamics, while (15) is independent of the input sequence and thus can be computed *a priori*. This second advantage is not necessarily available if the dynamics are not linear (Section IV).

To ensure that the probability of collision with any obstacle on a given time step does not exceed $\Delta \equiv 1 - p_{\text{safe}}$, it is sufficient to show that the probability of collision with each of the B obstacles at

that time step does not exceed Δ/B .² The j th obstacle is represented through the conjunction of linear inequalities

$$\bigwedge_{i=1}^{n_j} a_{ij}^T x_t < a_{ij}^T c_{ijt} \quad \forall t \in \mathbb{Z}_{0,t_f}, \quad (16)$$

where n_j is the number of constraints defining the j th obstacle, and c_{ijt} is a point nominally (i.e. $c_{jt} = \hat{c}_{jt}$) on the i th constraint at time step t ; note that a_{ij} is not dependent on t , since the obstacle shape and orientation are fixed. To avoid all obstacles, the system must satisfy B disjunctions of constraints at each time step,

$$\bigvee_{i=1}^{n_j} a_{ij}^T x_t \geq a_{ij}^T c_{ijt} \quad \forall j \in \mathbb{Z}_{1,B}, \quad \forall t \in \mathbb{Z}_{0,N}. \quad (17)$$

For each obstacle - consider the j th one below - it is sufficient to not satisfy any one constraint in the conjunction (16). Thus, the probability of collision is *lower-bounded* by the probability of satisfying any single constraint:

$$P(\text{collision}) \leq P(a_{ij}^T X_t < a_{ij}^T c_{ijt}) \quad \forall i \in \mathbb{Z}_{1,n_j}. \quad (18)$$

Pulling everything together, to prove the probability of collision with the j th obstacle does not exceed Δ/B , it is sufficient to show that

$$\bigvee_{i=1}^{n_j} P(a_{ij}^T X_t < a_{ij}^T C_{ijt}) \leq \Delta/B, \quad (19)$$

where $C_{ijt} = c_{ijt} + (c_{jt} - \hat{c}_{jt})$ is a random variable due to (11).

Next, a change of variables is made to render the problem tractable for path planning algorithms. For the i th constraint of the j th obstacle at time step t apply the change of variable

$$V = a_{ij}^T X_t - a_{ij}^T C_{ijt}; \quad (20)$$

it can be shown¹⁶ that the mean and covariance for V are

$$\hat{v} = a_{ij}^T \hat{x}_t - a_{ij}^T c_{ijt}, \quad (21)$$

$$P_v = \sqrt{a_{ij}^T (P_{x_t} + P_{c_{jt}}) a_{ij}}. \quad (22)$$

With this change of variables, the probabilistic constraint can then be written as

$$P(V < 0) \leq \Delta/B. \quad (23)$$

This probabilistic constraint can be shown to be equivalent to a deterministic constraint,²

$$P(V < 0) \leq \Delta/B \Leftrightarrow \hat{v} \geq \sqrt{2}P_v \text{erf}^{-1} \left(1 - 2\frac{\Delta}{B} \right), \quad (24)$$

where $\text{erf}(\cdot)$ denotes the standard error function. Using this, the constraints (17) are probabilistically satisfied for the true state x_t if the conditional mean \hat{x}_t satisfies the modified constraints

$$\bigvee_{i=1}^{n_j} a_{ij}^T \hat{x}_t \geq b_{ij} + \bar{b}_{ijt} \quad \forall j \in \mathbb{Z}_{1,B}, \quad \forall t \in \mathbb{Z}_{0,N}, \quad (25)$$

$$\bar{b}_{ijt} = \sqrt{2}P_v \text{erf}^{-1} \left(1 - 2\frac{\Delta}{B} \right). \quad (26)$$

The term \bar{b}_{ijt} represents the amount of *deterministic* constraint tightening necessary to ensure *probabilistic* constraint satisfaction.

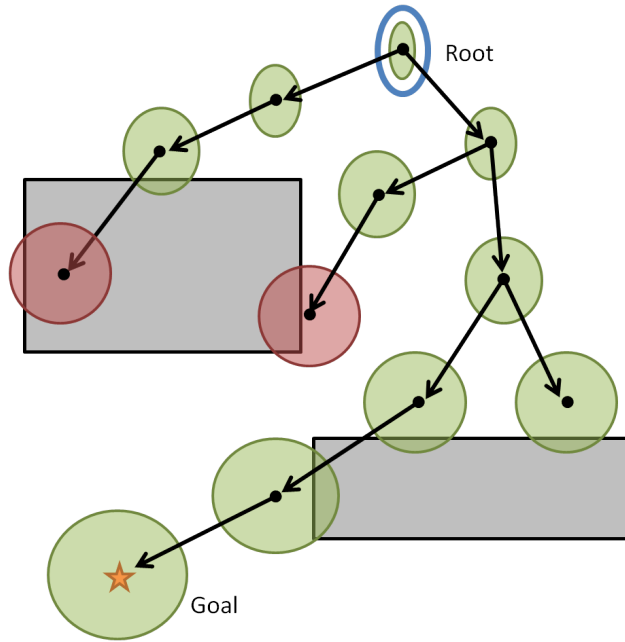


Figure 1. Diagram of the chance constrained RRT algorithm. Given an initial state distribution at the tree root (blue), the algorithm grows a tree of state distributions in order to find a probabilistically feasible path to the goal (yellow star). The uncertainty in the state at each node is represented as an uncertainty ellipse. Each state distribution is checked probabilistically against the constraints (gray). If the probability of collision is too high, the node is discarded (red); otherwise the node is kept (green) and may be used to grow future trajectories.

B. Algorithm Overview

This section reviews the chance constrained RRT (CC-RRT) algorithm, an extension of the traditional RRT algorithm which allows for probabilistic constraints. Whereas the traditional RRT algorithm grows a tree of states which are known to be feasible, the chance constrained RRT algorithm grows a tree of state distributions which are known to satisfy an upper bound on probability of collision (Figure 1).

The fundamental operation in the standard RRT algorithm is the incremental growth of a tree of dynamically feasible trajectories, rooted at the system’s current state x_t .¹¹ Crucially, the RRT algorithm employs trajectory-wise constraint checking, allowing for the incorporation of possibly complex constraints. The CC-RRT algorithm can leverage this by explicitly computing a bound on the probability of collision at each node, rather than simply satisfying tightened constraints for a fixed bound. To grow a tree of dynamically feasible trajectories, it is necessary for the RRT to have an accurate model of the vehicle dynamics (1) for simulation. Since the CC-RRT algorithm grows a tree of state distributions, in this case the model is assumed to be the propagation of the state conditional mean (14) and covariance (15), rewritten here as

$$\hat{x}_{t+k+1|t} = A\hat{x}_{t+k|t} + Bu_{t+k|t}, \quad (27)$$

$$P_{t+k+1|t} = AP_{t+k|t}A^T + P_w, \quad (28)$$

where t is the current system time step and $(\cdot)_{t+k|t}$ denotes the predicted value of the variable at time step $t+k$.

The CC-RRT tree expansion step, used to incrementally grow the tree, is given in Algorithm 1. Algorithm 2 shows how the algorithm executes some portion of the tree while continuing to grow it. Please consult [16] for more details on the algorithms and their implementation. Note that p_{safe} denotes the lower bound on the likelihood of constraint satisfaction at each timestep.

There are two ways to utilize the chance constraint formulation within the CC-RRT framework: **Offline CC-RRT** and **Online CC-RRT**. In the Offline CC-RRT approach, similar to the formulation presented by Blackmore *et al.*,² a fixed probability bound Δ/B is applied evenly across all obstacles, resulting in tightened deterministic constraints (25) which can be computed off-line for each time step. During tree growth, one can compute whether the deterministic chance constraints (25) are satisfied for all obstacles for

Algorithm 1 CC-RRT, Tree Expansion

```
1: Inputs: tree  $\mathcal{T}$ , current time step  $t$ 
2: Take a sample  $x_{\text{samp}}$  from the environment
3: Identify the  $M$  nearest nodes using heuristics
4: for  $m \leq M$  nearest nodes, in the sorted order do
5:    $N_{\text{near}} \leftarrow$  current nodes
6:    $(\hat{x}_{t+k|t}, P_{t+k|t}) \leftarrow$  final state distribution of  $N_{\text{near}}$ 
7:   while  $\Delta_{t+k}(\hat{x}_{t+k|t}, P_{t+k|t}) \leq 1 - p_{\text{safe}}$  and  $\hat{x}_{t+k|t}$  has not reached  $x_{\text{samp}}$  do
8:     Select input  $u_{t+k|t} \in \mathcal{U}$ 
9:     Simulate  $(\hat{x}_{t+k+1|t}, P_{t+k+1|t})$  using (27)-(28)
10:    Create intermediate nodes as appropriate
11:     $k \leftarrow k + 1$ 
12:  end while
13:  for each probabilistically feasible node  $N$  do
14:    Update cost estimates for  $N$ 
15:    Add  $N$  to  $\mathcal{T}$ 
16:    Try connecting  $N$  to  $\mathcal{X}_{\text{goal}}$  (lines 5-13)
17:    if connection to  $\mathcal{X}_{\text{goal}}$  probabilistically feasible then
18:      Update upper-bound cost-to-go of  $N$  and ancestors
19:    end if
20:  end for
21: end for
```

a given conditional mean \hat{x} and covariance P_x . Growth of the simulated trajectory continues only if these deterministic chance constraints are satisfied.

It is also possible to identify a more precise bound on the probability of collision specific to each time step, as used for Online CC-RRT. This approach leverages the relationship in (24) to compute the exact probability of satisfying each individual constraint for a given distribution $\mathcal{N}(\hat{x}, P_x)$ - an operation which is possible due to iterative constraint checking in the RRT algorithm. Through repeated use of this operation, a bound can be computed for each time step on the probability of collision. The tradeoff is the computational complexity introduced by computing this bound at each time step, though the increase is sufficiently small to maintain the approach's suitability for on-line implementation.

The key to the Online CC-RRT approach is the relationship

$$P(V < 0) = \frac{1}{2} \left(1 - \operatorname{erf} \left[\frac{\hat{v}}{\sqrt{2}P_v} \right] \right), \quad (29)$$

which has been shown¹⁶ to be derived from (24). Again consider the i th constraint of the j th obstacle at time step t , using the change of variables (20). Let $\Delta_{ijt}(\hat{x}, P_x)$ denote the probability that this constraint is satisfied for a Gaussian distribution with mean \hat{x} and covariance P_x ; using (29), we have that

$$\Delta_{ijt}(\hat{x}, P_x) = \frac{1}{2} \left(1 - \operatorname{erf} \left[\frac{a_{ij}^T \hat{x}_t - a_{ij}^T c_{ijt}}{\sqrt{2a_{ij}^T (P_{x_t} + P_{c_j}) a_{ij}}} \right] \right). \quad (30)$$

Now define

$$\Delta_t(\hat{x}_t, P_{x_t}) \equiv \sum_{j=1}^B \min_{i=1, \dots, n_j} \Delta_{ijt}(\hat{x}_t, P_{x_t}), \quad (31)$$

used in line 8 of Algorithm 1; it can be shown that this term provides an upper bound on the probability of a collision with any obstacle at time step t .

IV. Nonlinear Dynamics

In this section, the implications of removing the assumption of linear dynamics on the CC-RRT algorithm are considered. The key step in our approach is to linearize the dynamics at each timestep, based on the

Algorithm 2 CC-RRT, Execution Loop

```
1: Initialize tree  $\mathcal{T}$  with node at  $(\hat{x}_0, P_{x_0}), t = 0$ 
2: while  $\hat{x}_t \notin \mathcal{X}_{\text{goal}}$  do
3:   Use measurements, if any, to repropagate state distributions
4:   while time remaining for this time step do
5:     Expand the tree by adding nodes (Algorithm 1)
6:   end while
7:   Use cost estimates to identify best path  $\{N_{\text{root}}, \dots, N_{\text{target}}\}$ 
8:   if no paths exist then
9:     Apply safety action and goto line 17
10:  end if
11:  Repropagate the path state distributions using (27)-(28)
12:  if repropagated best path is probabilistically feasible then
13:    Apply best path
14:  else
15:    Remove infeasible portion of best path and goto line 7
16:  end if
17:   $t \leftarrow t + \Delta t$ 
18: end while
```

current system state and input, in order to maintain a predicted state distribution which is Gaussian. In particular, though the theoretical guarantees on probabilistic constraint satisfaction are lost through this Gaussian approximation, the resulting bounds can provide a useful estimate on path safety, especially as the timestep duration and the degree of nonlinearity decrease.

A. Open-Loop Dynamics

Suppose that the true system dynamics take the form (1),(5)-(6). The primary impact of the loss of linear dynamics is that the future state distributions can no longer be assumed to be Gaussian, which invalidates (14)-(15). The key question is then how to modify lines 7 and 8 of Algorithm 1, in which inputs are selected and used to simulate the state distribution dynamics, in order to reasonably admit nonlinear dynamics. Those steps use equations (32)-(33), repeated here:

$$\hat{x}_{t+k+1|t} = A\hat{x}_{t+k|t} + Bu_{t+k|t}, \quad (32)$$

$$P_{t+k+1|t} = AP_{t+k|t}A^T + P_w. \quad (33)$$

The key missing piece is the lack of an A matrix in the case of nonlinear dynamics; however, this can be obtained by linearizing the nonlinear dynamics. First, assume the disturbance free (i.e., zero-mean) dynamics $w_t \equiv 0$. For a given state x_t and input u_t , the nonlinear dynamics (1) can be linearized as follows:

$$x_{t+1} \approx \tilde{A}x_t + \tilde{B}u_t, \quad (34)$$

$$\tilde{A} = \frac{\partial f}{\partial x}(x_t, u_t), \quad (35)$$

$$\tilde{B} = \frac{\partial f}{\partial u}(x_t, u_t), \quad (36)$$

where $\frac{\partial f}{\partial x}(x_t, u_t)$ denotes the partial derivative of f with respect to x at the point $(x = x_t, u = u_t, w = 0)$.

In the case of open-loop dynamics, only \tilde{A} is necessary to perform covariance updates; the full nonlinear model can and should still be used to update the mean of the state distribution, \hat{x}_{t+1} . Thus, in the case of nonlinear open-loop dynamics, the system model (27)-(33) takes the form

$$\hat{x}_{t+k+1|t} = f(\hat{x}_{t+k|t}, u_{t+k|t}, 0), \quad (37)$$

$$P_{t+k+1|t} = \tilde{A}P_{t+k|t}\tilde{A}^T + P_w, \quad (38)$$

where \tilde{A} is defined in (35).

Note that due to this linearization, the theoretical guarantees previously presented for the case of linear dynamics under Gaussian noise¹⁶ are no longer valid; however, it is straightforward to show that the quality of the approximation will improve as the timestep duration and the degree of nonlinearity are decreased. Each of these results in the linearization presenting a more accurate approximation of the nonlinear dynamics.

B. Closed-Loop Dynamics

Consider the case of closed-loop dynamics, where the feasible input in line 7 of Algorithm 1 is computed using some closed-loop control law. In closed-loop, some reference state $r \in \mathbb{R}^{n_r}$ is propagated based on the sample x_{samp} ; the input is then selected through the controller

$$u = \kappa(x, r), \quad (39)$$

where u is the input to be applied to the node with state x and κ is the (nonlinear) control law. Through the reference r , the trajectories can be designed to meet certain qualitative considerations, such as a desired speed.¹⁵

For the assumption of linear dynamics, this control law would most typically take the form

$$u_t = K(x_t - r_t), \quad (40)$$

where K is some gain matrix. It is straightforward to show that upon this application of the closed-loop control law, the equations (14)-(15) become

$$\hat{x}_{t+k+1|t} = (A + BK)\hat{x}_{t+k|t} - Br_{t+k|t}, \quad (41)$$

$$P_{t+k+1|t} = (A + BK)P_{t+k|t}(A + BK)^T + P_w. \quad (42)$$

Note that in both equations, A has become $A + BK$, indicating the closed-loop nature of the uncertainty environment.¹⁵

If the assumption of linear dynamics is lifted, (34)-(36) can still be used to approximate the state distributions as Gaussian; however, in this case, it is also necessary to linearize the control law (39). Thus, in the case of nonlinear closed-loop dynamics, the system model (27)-(33) takes the form

$$\hat{x}_{t+k+1|t} = f(\hat{x}_{t+k|t}, \kappa(x_{t+k|t}, r_{t+k|t}), 0), \quad (43)$$

$$P_{t+k+1|t} = (\tilde{A} + \tilde{B}\tilde{K})P_{t+k|t}(\tilde{A} + \tilde{B}\tilde{K})^T + P_w, \quad (44)$$

$$\tilde{K} = \frac{\partial \kappa}{\partial x}(x_t, r_t). \quad (45)$$

where (45) is defined in the same manner as (35)-(36).

V. Non-Gaussian Uncertainty

In the case of nonlinear dynamics subject to Gaussian noise, it was reasonable to approximate a future state distribution as Gaussian by linearizing the dynamics. However, when even the noise itself is non-Gaussian, it is likely that a Gaussian model is no longer appropriate to capture the predicted uncertainty. In this case, a particle-based framework can be used to statistically approximate this uncertainty, at a resolution which can be dictated by the user. This section introduces the particle-based chance constraint framework used to approximate both timestep and path feasibility when subject to non-Gaussian uncertainty.

Algorithm 3 presents the tree expansion step and execution loop, respectively, for the particle-based extension of CC-RRT, referred to here as PCC-RRT; the modifications to Algorithm 2 are straightforward and omitted for brevity. A set of P_{max} particles are maintained at each node, each with a position x and a weight w (weights are discussed below). There are two parameters the user can specify to indicate the degree of probabilistic constraint violation allowed - the average likelihood of feasibility at each node cannot exceed $p_{\text{safe}}^{\text{node}}$, while the average likelihood of feasibility over an entire path cannot exceed $p_{\text{safe}}^{\text{node}}$. The latter bound is a key advantage of this particle-based approach, as it is quite difficult to approximate theoretically. The former likelihood is computed by averaging the weights of all feasible nodes, $\sum_p w_{t+k|t}^{(p)}$ (line 7), while the latter is computed iteratively over a path by multiplying the prior node's path probability by the weight of existing nodes that are still feasible (line 15).

The tree expansion step is carried out as follows. After identifying the nearest nodes (line 3) to a new sample (line 2), the number of particles is first resampled up to P_{max} (line 8), such that the number of simulations per step is constant at P_{max} . After identifying the appropriate input across all particles (line 9), each particle is simulated using the vehicle uncertain dynamics (line 11); disturbances are sampled using the process noise distribution. A weight is then assigned to each particle (see below; line 12). By removing any

Algorithm 3 PCC-RRT, Tree Expansion

```
1: Inputs: tree  $\mathcal{T}$ , current time step  $t$ 
2: Take a sample  $x_{\text{samp}}$  from the environment
3: Identify the  $M$  nearest nodes using heuristics
4: for  $m \leq M$  nearest nodes, in the sorted order do
5:    $N_{\text{near}} \leftarrow$  current node
6:    $\{x_{t+k|t}^{(p)}, w_{t+k|t}^{(p)}\} \leftarrow$  set of feasible particles at  $N_{\text{near}}$ , with weights
7:   while  $\sum_p w_{t+k|t}^{(p)} \geq p_{\text{safe}}^{\text{node}}$  and  $P_k^{\text{path}} \geq p_{\text{safe}}^{\text{path}}$  and  $\hat{x}_{t+k|t}$  has not reached  $x_{\text{samp}}$  do
8:     Resample particles up to count of  $P_{\text{max}}$ , using weights  $w_{t+k|t}^{(p)}$ 
9:     Select input  $u_{t+k|t} \in \mathcal{U}$ 
10:    for each particle  $p$  do
11:      Simulate  $x_{t+k+1|t}^{(p)}$  using (1) and sampled disturbance  $w_{t+k}$ 
12:      Assign weight  $w_{t+k+1|t}^{(p)}$  to particle
13:    end for
14:    Remove infeasible particles
15:     $P_{k+1}^{\text{path}} \leftarrow P_k^{\text{path}} \cdot \sum_p w_{t+k|t}^{(p)}$ 
16:     $k \leftarrow k + 1$ 
17:  end while
18:  for each probabilistically feasible node  $N$  do
19:    Update cost estimates for  $N$ 
20:    Add  $N$  to  $\mathcal{T}$ 
21:    Try connecting  $N$  to  $\mathcal{X}_{\text{goal}}$  (lines 5-13)
22:    if connection to  $\mathcal{X}_{\text{goal}}$  probabilistically feasible then
23:      Update upper-bound cost-to-go of  $N$  and ancestors
24:    end if
25:  end for
26: end for
```

particles that are infeasible, the sum of weights of remaining particles indicates the likelihood of constraint satisfaction at that node, which can be used to iteratively compute the likelihood of constraint satisfaction of the path up to that point (line 15). Any paths which reach a new sample while satisfying these bounds (line 7) is added to the tree (lines 18-25).

A. Uniform Resampling

In the uniform resampling scheme, all particles are assigned an identical weight at line 12, say (without loss of generality) $w_{t+k+1|t}^{(p)} = 1$. When this is the case, every particle has an equal likelihood of being resampled at line 8.

B. Probabilistic Resampling

In the probabilistic resampling scheme, each particle is assigned a weight based on the likelihood of that particle actually existing; this is a function of the likelihood of each portion of the uncertainty that is sampled. This can be computed iteratively by node, as follows:

$$w_{t+k+1|t}^{(p)} = w_{t+k|t}^{(p)} \cdot \mathbb{P}(W_{t+k} = w_{t+k}). \quad (46)$$

This requires additional computation, especially as the complexity of the uncertainty environment increases; however, it can provide a better overall approximation of the state distribution at each timestep, for the same number of particles.

VI. Results

The following results demonstrate the validity of both approaches in extending the CC-RRT framework to provide probabilistic constraint satisfaction bounds for nonlinear dynamics and/or non-Gaussian uncertainty. Future revisions will significantly expand these results to include multiple types of system model, closed-loop dynamics for Section IV, probabilistic resampling, and statistical analysis of reaching the goal safely.¹⁶

Consider the operation of a skid-steered vehicle in a 2D environment, with nonlinear dynamics

$$\begin{aligned}
 x_{t+1} &= x_t + dt(1/2)(v_t^L + v_t^R) \cos \theta_t, \\
 y_{t+1} &= y_t + dt(1/2)(v_t^L + v_t^R) \sin \theta_t, \\
 \theta_{t+1} &= \theta_t + dt(v_t^R - v_t^L), \\
 v_t^L &= \text{sat}(\bar{v}_t + 0.5\Delta v_t + w_t^L, 0.5), \\
 v_t^R &= \text{sat}(\bar{v}_t - 0.5\Delta v_t + w_t^R, 0.5),
 \end{aligned}$$

where $dt = 0.02$ s, (x, y) is the vehicle position, θ is the heading, v^L and v^R are the left and right wheel speeds, respectively, sat is the saturation function and (w^L, w^R) is a bounded disturbance. The system inputs are specified in terms of a mean velocity \bar{v} and differential velocity Δv . A variation of the pure pursuit controller¹⁰ is applied, assuming forward direction only. Simulations were performed using a Java implementation, run on an Intel 2.53 GHz quad-core laptop with 3.48GB of RAM.

Figure 2 demonstrates the use of Algorithm 1 using the open-loop linearization (37)-(38); paths are expanded by randomly sampling many inputs, then selecting those inputs that guide the vehicle closer to the sample (this leads to much of the observed “zig-zag” behavior). Figure 3 demonstrates the use of Algorithm 3 using the uniform resampling scheme; note that the per-node safety bound is fixed in both figures, whereas the per-path safety bound is increased in the rightmost figure.

VII. Conclusions

This paper has presented several extensions to the previously-developed CC-RRT path planning framework, which can be used to generate paths in real-time which satisfy probabilistic bounds on constraint satisfaction at each timestep. These extensions seek to remove the two assumptions which limit the applicability of the existing approach, linear dynamics and non-Gaussian uncertainty. The former assumption is removed by allowing the state distribution covariance at future timesteps to be approximated via a linearization of the nonlinear dynamics; though the probabilistic bounds can no longer be guaranteed, the approximation can still be quite useful for sufficient small timesteps and systems that are not highly nonlinear. The assumption of non-Gaussian uncertainty is removed by addressing the uncertainty statistically, through a particle-based framework. By using the same number of particles at each timestep, and maintaining bounds on constraint satisfaction both at every node and along every path, the user maintains a great deal of control on the level of allowable risk in the formulation.

Future work will consider the use of additional heuristics to take advantage of the computed risk evaluations, such as probabilistic selection heuristics.^{16,25} We will also consider the use of techniques from the unscented Kalman filter to maintain state distributions for highly nonlinear system dynamics.

Acknowledgements

This work was sponsored (in part) by the AFOSR and USAF under grant FA9550-08-1-0086. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Office of Scientific Research or the U.S. Government.

References

- ¹L. Blackmore. A probabilistic particle control approach to optimal, robust predictive control. In *Proceedings of the AIAA Guidance, Navigation and Control Conference*, 2006.
- ²L. Blackmore, H. Li, and B. Williams. A probabilistic approach to optimal robust path planning with obstacles. In *Proceedings of the IEEE American Control Conference*, pages 2831–2837, 2006.
- ³B. Burns and O. Brock. Sampling-based motion planning with sensing uncertainty. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3313–3318, Roma, Italy, April 2007.
- ⁴W. J. A. Dahm. Technology horizons: A vision for air force science & technology during 2010-2030. Technical report, United States Air Force, 2010.
- ⁵A. Foka and P. Trahanias. Real-time hierarchical POMDPs for autonomous robot navigation. *Robotics and Autonomous Systems*, 55:561–571, 2007.
- ⁶M. Greytak and F. Hover. Motion planning with an analytic risk cost for holonomic vehicles. In *Proceedings of the IEEE Conference on Decision and Control*, 2009.

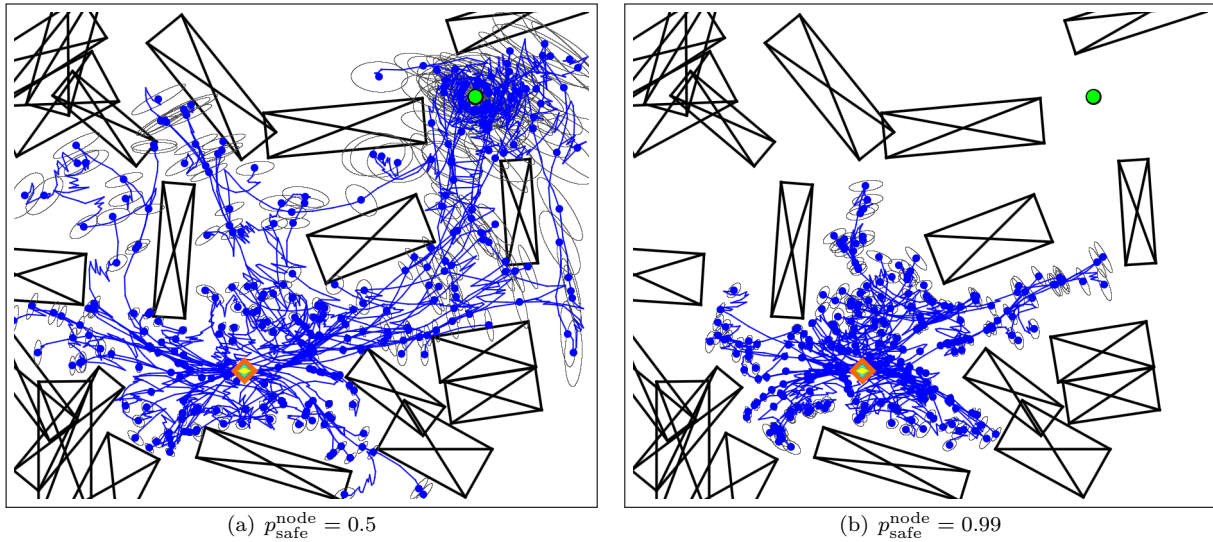


Figure 2. Representative trees (blue) generated by the nonlinear, online CC-RRT algorithm after 20 seconds of growth for a skid-steered vehicle (orange) attempting to reach the goal (green). The vehicle is controlled in open-loop, and is subject to small position process noise and large heading process noise. Here $p_{\text{safe}}^{\text{node}}$ denotes the required probability of feasibility at each timestep. The state distribution at each node is denoted by a $1\text{-}\sigma$ uncertainty ellipse; note that the ellipses are generally aligned perpendicular to the direction of motion.

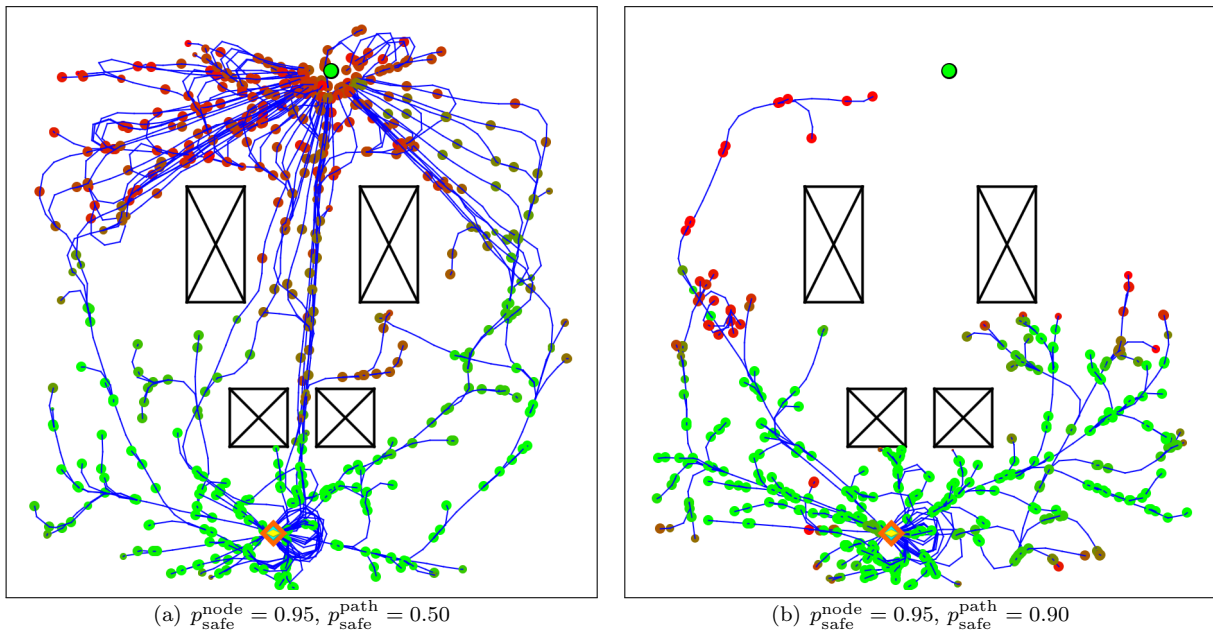


Figure 3. Representative trees (blue) generated by the particle CC-RRT algorithm after 40 seconds of growth for a skid-steered vehicle (orange) attempting to reach the goal (green); 100 particles are generated for each node, using uniform resampling. The vehicle is controlled in closed-loop; at each timestep there is a 50% chance that either its left or right skid will slip (but not both), causing a uniformly-distributed disturbance to that skid's speed. Here $p_{\text{safe}}^{\text{node}}$ denotes the required probability of feasibility at each timestep; a node's size decreases as it comes closer to violating this constraint. On the other hand, $p_{\text{safe}}^{\text{path}}$ denotes the required probability of feasibility along an entire path; a node's color changes from green to red as it comes closer to violating this constraint.

- ⁷L. J. Guibas, D. Hsu, H. Kurniawati, and E. Rehman. Bounded uncertainty roadmaps for path planning. In *Proceedings of the International Workshop on the Algorithmic Foundations of Robotics*, 2008.
- ⁸G. Kewlani, G. Ishigami, and K. Iagnemma. Stochastic mobility-based path planning in uncertain environments. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, pages 1183–1189, St. Louis, MO, USA, October 2009.
- ⁹Y. Kuwata, J. Teo, G. Fiore, S. Karaman, E. Frazzoli, and J. P. How. Real-time motion planning with applications to autonomous urban driving. *IEEE Transactions on Control Systems Technology*, 17(5):1105–1118, September 2009.
- ¹⁰Y. Kuwata, J. Teo, S. Karaman, G. Fiore, E. Frazzoli, and J. P. How. Motion planning in complex environments using closed-loop prediction. Submitted to the Proceedings of the IEEE Conference on Guidance, Navigation, and Control, 2008.
- ¹¹S. M. LaValle. Rapidly-exploring random trees: A new tool for path planning. Technical Report 98-11, Iowa State University, October 1998.
- ¹²S. M. LaValle. *Planning Algorithms*. Cambridge University Press, Cambridge, U.K., 2006.
- ¹³S. M. LaValle and R. Sharma. On motion planning in changing, partially-predictable environments. *International Journal of Robotics Research*, 16(6):775–824, 1995.
- ¹⁴P. Li, M. Wendt, and G. Wozny. A probabilistically constrained model predictive controller. *Automatica*, 38:1171–1176, 2002.
- ¹⁵B. D. Luders, S. Karaman, E. Frazzoli, and J. P. How. Bounds on tracking error using closed-loop rapidly-exploring random trees. In *Proceedings of the IEEE American Control Conference*, June-July 2010.
- ¹⁶B. D. Luders, M. Kothari, and J. P. How. Chance constrained RRT for probabilistic robustness to environmental uncertainty. In *Proceedings of the AIAA Guidance, Navigation and Control Conference*, 2010.
- ¹⁷N. A. Melchior and R. Simmons. Particle RRT for path planning with uncertainty. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2007.
- ¹⁸A. S. Miralles and M. A. S. Bobi. Global path planning in gaussian probabilistic maps. *Journal of Intelligent and Robotic Systems*, 40:89–102, 2004.
- ¹⁹P. E. Missiuro and N. Roy. Adapting probabilistic roadmaps to handle uncertain maps. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1261–1267, Orlando, FL, May 2006.
- ²⁰R. Pepy, M. Kieffer, and E. Walter. Reliable robust path planning. *International Journal of Applied Math and Computer Science*, 1:1–11, 2009.
- ²¹R. Pepy and A. Lambert. Safe path planning in an uncertain-configuration space using RRT. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, pages 5376–5381, October 2006.
- ²²S. Prentice and N. Roy. The belief roadmap: Efficient planning in lienar pomdps by factoring the covariance. In *Proceedings of the International Symposium of Robotics Research*, Hiroshima, Japan, November 2007.
- ²³N. Roy, G. Gordon, and S. Thrun. Planning under uncertainty for reliable health care robotics. *Field and Service Robotics*, STAR 24:417–426, 2006.
- ²⁴S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, 2005.
- ²⁵C. Urmson and R. Simmons. Approaches for heuristically biasing RRT growth. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, pages 1178–1183, Las Vegas, NV, October 2003.