CC-RRT Implementation Project

Pranay Katyal Ethan Zhong Manav Mepani Harsh Chhajed

***Abstract*—In this project, we used the paper "Chance Constrained RRT for Probabilistic Robustness to Environmental Uncertainty" as a foundational reference to set up the basic problem structure. The paper re-introduces a novel real-time planning algorithm, Chance Constrained Rapidly-Exploring Random Trees (CC-RRT) [1], which incorporates chance constraints to ensure probabilistic feasibility for linear systems affected by process noise and/or uncertain, potentially dynamic obstacles. By leveraging the RRT framework, the algorithm benefits from the computational advantages of sampling-based methods, such as efficient trajectory-wise constraint checking and the ability to incorporate heuristics, while formally accounting for uncertainty in the planning process. Assuming Gaussian noise, the method enables feasibility checks at each time step via simulation of the conditional mean state and evaluation of linear constraints. Additionally, it allows for the computation of less conservative probability bounds with minimal extra computation. We implemented this approach and replicated the results to validate its ability to efficiently identify and execute probabilistically safe trajectories in real time.**

***Index Terms*—Probabilistic Motion Planning, Dynamic Obstacles, RRT (Rapidly-exploring Random Trees), Conditional Mean Simulation, CC-RRT (Chance Constrained RRT), Motion planning under Uncertainty.**

I. Introduction and Literature Review

A major challenge in motion planning is identifying feasible paths for autonomous systems operating under various types of uncertainty. These uncertainties can generally be classified into four key categories: (1) uncertainty in system configuration; (2) uncertainty in the dynamic model of the system; (3) limited situational awareness of the environment; and (4) unpredictability in the future state of the environment. Real-world applications typically involve one or more of these uncertainties, which must be explicitly modeled for safe and robust planning.

However, in scenarios where uncertainties are either unbounded or the system lacks sufficient control authority, it may not always be possible to guarantee absolute feasibility of the planned path. To manage this, the trade-off between planner conservatism and the risk of infeasibility becomes critical.

This balance can be effectively captured using chance constraints, which define probabilistic bounds on constraint violations. These constraints allow the system to plan under uncertainty by ensuring that the likelihood of violating key constraints remains below a defined threshold. In particular, for systems affected by Gaussian noise, probabilistic feasibility can be approximated through deterministic tightening of state constraints, or by evaluating the conditional mean of the system state at each step.

The paper by Blackmore et al. (2006) [2] lays the foundational formulation of chance-constrained optimization for linear systems with Gaussian process noise. However, extending this framework to dynamic, high-dimensional, or logically constrained systems often leads to computationally expensive solutions, such as mixed-integer or nonlinear optimization programs, which are impractical for real-time planning.

To address these limitations, the Chance-Constrained Rapidly-Exploring Random Tree (CC-RRT) algorithm [1] introduces an incremental, sampling-based approach for planning in uncertain environments. CC-RRT builds upon standard RRT by incorporating chance constraints directly into the tree expansion process. It allows for efficient, probabilistically safe path generation by checking feasibility at each time step using either simulation of the conditional mean or a tighter probabilistic bound.

In our project, we used the CC-RRT framework as a reference for setting up the core planning problem and for designing our implementation. We aimed to replicate and extend the results presented in the paper to better understand the performance of sampling-based motion planning under uncertainty, particularly in the presence of process noise and dynamic obstacles.

II. Formulation Explanations

A. Chance Constraint

This section reviews the chance constraint formulation developed by Blackmore et al. [2], where all obstacles are assumed to be static and located at known positions.

Given a sequence of control inputs $u_0, \ldots, u_{N-1}$, the state x_t of a linear system under Gaussian process noise is modeled as a Gaussian random variable:

$$P(X_t \mid u_0, \ldots, u_{t-1}) = \mathcal{N}(\mu_t, P_t), \quad t \in [0, N] \tag{1}$$

where the mean and covariance can be computed explicitly as:

$$\mu_t = A^t x_0 + \sum_{k=0}^{t-1} A^{t-k-1} B u_k \tag{2}$$

$$P_t = A^t P_0 (A^T)^t + \sum_{k=0}^{t-1} A^{t-k-1} P_w (A^T)^{t-k-1} \tag{3}$$

or recursively as:

$$\mu_{t+1} = A\mu_t + Bu_t \tag{4}$$

$$P_{t+1} = AP_tA^T + P_w \tag{5}$$

To ensure that the probability of collision at each time step does not exceed $\Delta = 1 - p_{\text{safe}}$, it is sufficient to ensure that the probability of colliding with each of the B obstacles is less than or equal to β:

$$\sum_{i=1}^{n_j} P(a_{ij}^T X_t < b_{ij}) \leq \frac{\Delta}{\beta} \quad \text{for all } j \in [1, B] \tag{6}$$

Each obstacle is represented as a conjunction of n_j linear constraints:

$$a_{ij}^T x_t < b_{ij}, \quad i = 1, \ldots, n_j \tag{7}$$

To make this constraint tractable for real-time planning, a change of variable is applied:

$$V = a_{ij}^T X_t - b_{ij} \tag{8}$$

$$v = a_{ij}^T \mu_t - b_{ij} \tag{9}$$

$$P_v = a_{ij}^T P_t a_{ij} \tag{10}$$

Thus, the constraint:

$$P(V < 0) \leq \frac{\Delta}{\beta} \tag{11}$$

can be equivalently written using the error function erf:

$$v \geq \sqrt{2P_v}\,\text{erf}^{-1}(1 - 2\frac{\Delta}{\beta}) \tag{12}$$

This results in a tightened deterministic constraint:

$$a_{ij}^T \mu_t \leq b_{ij} - b_{ijt} \tag{13}$$

where

$$b_{ijt} = \sqrt{2P_v}\,\text{erf}^{-1}(1 - 2\frac{\Delta}{\beta}) \tag{14}$$

Since P_t can be computed offline, so can b_{ijt}, ensuring that the complexity of the problem remains unaffected even with chance constraints.

B. CC-with Dynamic Obstacles

This section extends the formulation in previous section to account for uncertainty in dynamic obstacles. This extension allows the chance constraint formulation to address additional forms of uncertainty that arise in common motion planning scenarios, including environmental unpredictability.

Let the jth obstacle at time step t be represented by a set of n_j linear inequalities:

$$a_{ij}^T x_t < a_{ij}^T c_{ijt}, \quad i = 1, \ldots, n_j, \quad t \in [0, t_f] \tag{15}$$

where c_{ijt} is a nominal point on the ith constraint of the jth obstacle at time t. Note that the shape and orientation of the obstacle are fixed, and a_{ij} is independent of t.

To avoid all B obstacles, the system must satisfy the disjunctive constraints at each time step:

$$\bigvee_{i=1}^{n_j} a_{ij}^T x_t \geq a_{ij}^T c_{ijt}, \quad j \in [1, B], \quad t \in [0, t_f] \tag{16}$$

To ensure that the probability of collision does not exceed β, it suffices to show:

$$\sum_{i=1}^{n_j} P(a_{ij}^T X_t < a_{ij}^T C_{ijt}) \leq \frac{\Delta}{\beta} \tag{17}$$

where $C_{ijt} = c_{ijt} + c_j$ is a random variable modeling the uncertainty of the obstacle's location.

We define a new random variable:

$$V = a_{ij}^T X_t - a_{ij}^T C_{ijt} \tag{18}$$

Then, V is normally distributed:

$$v = \mathbb{E}[V] = a_{ij}^T \mu_t - a_{ij}^T c_{ijt} \tag{19}$$

$$P_v = \text{Var}(V) = a_{ij}^T (P_t + P_{c_j}) a_{ij} \tag{20}$$

The chance constraint becomes:

$$P(V < 0) \leq \frac{\Delta}{\beta} \tag{21}$$

which is equivalent to the deterministic constraint:

$$a_{ij}^T \mu_t \geq a_{ij}^T c_{ijt} + b_{ijt} \tag{22}$$

where:

$$b_{ijt} = \sqrt{2P_v}\,\text{erf}^{-1}(1 - 2\frac{\Delta}{\beta}) \tag{23}$$

Summary of Modifications

To incorporate uncertain and/or dynamic obstacles into the original chance-constrained formulation, it is sufficient to:

- Replace the right-hand side of the inequality with $a_{ij}^T c_{ijt}$, where c_{ijt} tracks the obstacle's nominal trajectory.
- Replace the original variance $P_v = a_{ij}^T P_t a_{ij}$ with $P_v = a_{ij}^T (P_t + P_{c_j}) a_{ij}$ to account for obstacle uncertainty.

This modification shifts the constraint to follow the moving obstacle and adds the obstacle's uncertainty to the total variance, enabling more robust planning in dynamic environments.

C. CC-RRT

This section introduces the Chance-Constrained RRT (CC-RRT) algorithm, an extension of the traditional Rapidly-Exploring Random Tree (RRT) algorithm, which allows for the inclusion of probabilistic constraints. Unlike the standard RRT, which constructs a tree of feasible states, the CC-RRT algorithm constructs a tree of state *distributions* that satisfy a probabilistic upper bound on collision risk.

The core of the traditional RRT algorithm is the incremental growth of a tree rooted at the system's current state x_t, where each node represents a dynamically feasible trajectory [3]. The probability of selecting a node to expand is proportional to the volume of its Voronoi region under a uniform sampling distribution, inherently biasing the algorithm toward exploring unexplored regions of the state space. RRT supports various heuristic extensions and permits trajectory-wise constraint checking, making it suitable for problems with complex constraints.

CC-RRT leverages these properties by explicitly calculating a bound on the probability of constraint violation at each node, rather than enforcing tightened deterministic constraints with fixed bounds. In doing so, the planner retains computational efficiency while accommodating stochasticity in system dynamics and/or environment.

In order to propagate the uncertainty, CC-RRT uses the conditional mean and covariance of the system state, modeled by:

$$x_{t+k+1|t} = Ax_{t+k|t} + Bu_{t+k|t} \tag{24}$$
$$P_{t+k+1|t} = AP_{t+k|t}A^T + P_w \tag{25}$$

where $x_{t+k|t}$ and $P_{t+k|t}$ represent the predicted mean and covariance at future time $t + k$ given the current state x_t and control sequence $\{u_t, \ldots, u_{t+k}\}$.

III. Approach

Our project involves implementing and evaluating the CC-RRT (Chance Constrained Rapidly-Exploring Random Tree) algorithm across three different setups:

1) **Python-based CC-RRT from Scratch**
2) **C++ Geometric RRT-based CC-RRT using OMPL**
3) **C++ Control-based RRT-based CC-RRT using OMPL**

Each of these implementations follows a unique development strategy, detailed below.

A. Python Implementation from Scratch

The Python version serves as a foundational prototype where we manually implement the core components of a basic RRT algorithm. The steps in this pipeline are:

- Implement the RRT planner structure with modular design.
- Design a basic 2D environment for navigation and path planning.
- Incorporate a collision checking function as a separate sampling module.
- Gradually extend the base RRT to include chance constraints by introducing Gaussian noise sampling and probabilistic feasibility checking.

This version provides flexibility and allows for rapid experimentation with CC-RRT extensions such as custom probability thresholds, Gaussian uncertainty modeling, and simplified visualization.

B. C++ Geometric RRT with OMPL

For this version, we use the existing RRT planner from the OMPL (Open Motion Planning Library) as a base and modify it to implement CC-RRT behavior. The development pipeline includes:

- **Extract the base Geometric RRT planner** from OMPL's planner repository.
- **Modify the sampling strategy** to include Gaussian sampling centered around existing nodes to model uncertainty in the system.
- **Introduce chance-constrained checks** before adding a node to the tree, reducing unnecessary collision checks.
- **Ensure compatibility with OMPL interfaces** like planner data logging, state validity checking, and nearest neighbor querying.

This implementation aims to provide a performance-efficient solution while maintaining compatibility with OMPL's benchmarking and visualization tools.

C. C++ Control-based RRT with OMPL

This version extends OMPL's RRT control-based planner by applying CC-RRT principles in a dynamic control context:

- Retrieve OMPL's Control-based RRT planner and initialize its components.
- Modify the forward propagation function to account for state distribution (mean and covariance) as part of each motion primitive.
- Apply probabilistic constraints using precomputed bounds derived from system dynamics and noise models.
- Evaluate probabilistic feasibility using the conditional mean and variance propagation formulas defined in the original CC-RRT paper.

This version is most suitable for dynamic systems like mobile robots, drones, or manipulators where control inputs are directly modeled and chance constraints can be defined in terms of dynamic feasibility.

Summary of Common Modifications

All three implementations share the following overarching CC-RRT concepts:

- Sampling from a Gaussian distribution to reflect state uncertainty.
- Replacing strict collision checks with probabilistic feasibility conditions.
- Introducing a chance constraint threshold (Δ) to guide tree expansion.
- Ensuring that constraint violations remain within allowable bounds across time steps.

This multi-pronged approach allows us to explore the CC-RRT algorithm's flexibility and performance across different platforms and abstraction levels.

References

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

[2] L. Blackmore, H. Li, and B. Williams, "A probabilistic approach to optimal robust path planning with obstacles," in *2006 American Control Conference*. IEEE, 2006, pp. 7–pp.

[3] S. LaValle, "Rapidly-exploring random trees: A new tool for path planning," *Research Report 9811*, 1998.