Motion Planning Under Uncertainty with Complex Agents and Environments via Hybrid Search (Extended Abstract) *

Daniel Strawser, **Brian Williams**
Massachusetts Institute of Technology
dstrawse@alum.mit.edu, williams@mit.edu

Abstract

As autonomous systems tackle more real-world situations, mission success oftentimes cannot be guaranteed and the planner must reason about the probability of failure. Unfortunately, computing a trajectory that satisfies mission goals while constraining the probability of failure is difficult because of the need to reason about complex, multidimensional probability distributions. Recent methods have seen success using chance-constrained, model-based planning. We argue there are two main drawbacks to these approaches. First, current methods suffer from an inability to deal with expressive environment models such as 3D non-convex obstacles. Second, most planners rely on considerable simplifications when computing trajectory risk including approximating the agent's dynamics, geometry, and uncertainty. We apply hybrid search to the risk-bound, goal-directed planning problem. The hybrid search consists of a region planner and a trajectory planner. The region planner makes discrete choices by reasoning about geometric regions that the agent should visit in order to accomplish its mission. In formulating the region planner, we propose landmark regions that help produce obstacle-free paths. The region planner passes paths through the environment to a trajectory planner; the task of the trajectory planner is to optimize trajectories that respect the agent's dynamics and the user's desired risk of mission failure. We discuss three approaches to modeling trajectory risk: a CDF-based approach, a sampling-based collocation method, and an algorithm named Shooting Method Monte Carlo. A variety of 2D and 3D test cases are presented in the full paper including a linear case, a Dubins car model, and an underwater autonomous vehicle. The method is shown to outperform other methods in terms of speed and utility of the solution. Additionally, the models of trajectory risk are shown to better approximate risk in simulation.

*Full paper in the Journal of Artificial Intelligence Research, Volume 75 [Strawser and Williams, 2022].

1 Introduction

As they become more commonplace, autonomous systems must deal with increasingly uncertain surroundings. Autonomous cars quickly make decisions while navigating dynamic urban settings. Similarly, autonomous underwater vehicles accomplish a range of scientific and exploratory tasks while dealing with stochastic ocean currents.

Such widespread applicability has inspired a vast amount of prior work into motion planning under uncertainty. A common method of approximating trajectory risk is to assume a state distribution for which there is a quick-to-evaluate cumulative distribution function (CDF), such as the Gaussian distribution. In one such approach the authors combine a disjunctive program with a CDF-based evaluation of trajectory risk [Blackmore *et al.*, 2011]. Chance-constrained RRT proposes an extension of the popular rapidly-exploring random tree (RRT) algorithm [Luders *et al.*, 2010]. Probabilistic constraints are added to the basic RRT formulation; the path must avoid collisions with a predetermined probability. Another approach does not directly assume Gaussian uncertainty but models the state distribution of an agent that collides with an obstacle as a truncated Gaussian [Patil *et al.*, 2012].

Hybrid search has seen promise solving problems with discrete and continuous decision variables. ScottyActivity combines activity and trajectory planning over long horizons [Fernandez-Gonzalez *et al.*, 2018]. The Scotty planner decomposes the workspace into convex subregions and performs graph search over the subregions. Another example of hybrid search is the pSulu planner [Ono *et al.*, 2013]. pSulu solves a Chance-Constrained Qualitative State Plan (CCQSP) using branch and bound and convex optimization. Learning has seen success in a hybrid approach to motion planning. In one instance, a probabilistic roadmap (PRM) planner is combined with reinforcement learning to generate long trajectories for nonlinear systems [Faust *et al.*, 2018]. Our work builds on these and relies on hybrid search for more sophisticated models of risk than previous approaches.

1.1 Problem Statement

We consider the problem of an agent navigating from an initial state $\mathcal{I}$ to a sequence of goal states $\mathcal{G}$ while avoiding obstacles with probability $1 - \epsilon$. The purpose of the algorithm is to output an open loop control trajectory $\mathbf{u}$. The control

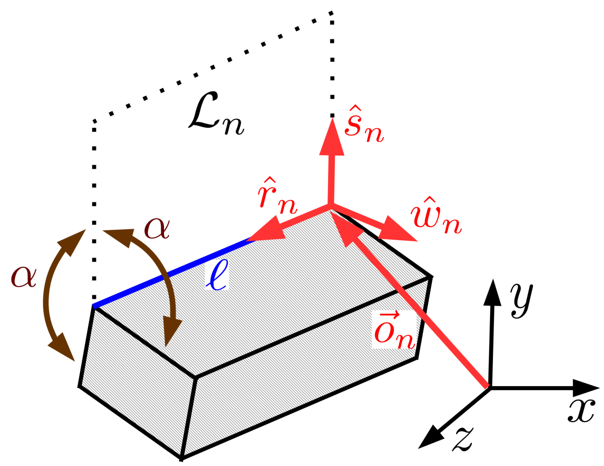

Figure 1: The plane and coordinate system that represent a landmark region $\mathcal{L}_n$ for a 3D workspace. The dotted lines do not represent a constraint; they are shown to illustrate the landmark region's plane.

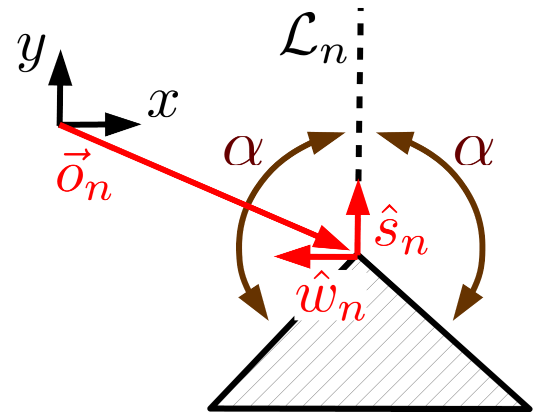

Figure 2: The ray that represents a landmark region $\mathcal{L}_n$ for a 2D workspace

outputs $\mathbf{u}$ represent inputs to the agent's actuators. The navigation problem can be framed as the optimal control problem:

$$\underset{\mathbf{x},\mathbf{u},\mathbf{h}}{\text{minimize}} \quad \mathbb{E}\left[J\left(\mathbf{x}_{0:K}, \mathbf{u}_{0:K-1}, \mathbf{h}_{0:K-1}\right)\right] \tag{1}$$

$$\text{subject to} \quad \dot{\mathbf{x}} = \mathbf{f}\left(\mathbf{x}, \mathbf{u}\right) + \mathbf{w} \tag{2}$$

$$P\left(\bigwedge_{k\in K} \mathbf{x}_k \in \mathcal{W}_{free}\right) \geq 1 - \epsilon \tag{3}$$

$$\mathbf{x}_0 \in \mathcal{I} \tag{4}$$

$$\mathbf{x}_{k_g} \in \mathcal{G}_g \;\; \forall \;\; 0 \leq g \leq G \tag{5}$$

Constraint (3) is important to this work; it models the probability that the trajectory is in the free region of the workspace, $\mathcal{W}_{free}$, and is termed the *chance constraint*.

We solve the risk-bound motion planning problem using hybrid search. The search consists of a region planner and a trajectory planner. The region planner passes candidate paths to the trajectory planner; the trajectory planner then optimizes a trajectory constrained to a specific path and determines if the chance constraint can be satisfied. The components are introduced in the following sections.

2 The Region Planner

The purpose of the region planner is to determine an appropriate set of active geometric constraints that allow the agent to accomplish its mission. To do this, the region planner reasons about geometric regions of the agent workspace, termed *regions*. The regions are used to model obstacle-avoidance constraints, goal constraints, and other geometric constraints on the agent's trajectory. The region planner connects sequences of regions into paths. Each path can be compiled into a set of constraints on the agent's continuous state trajectory.

The region planner receives as inputs an agent and a sequence of goals that the agent must accomplish. The output of the region planner is an ordered set of regions, a *path*, that represents a sequence of geometric constraints on the agent's position. This path is then passed to the trajectory planner.

2.1 Landmark Regions

Our work proposes landmark regions to model obstacle-avoidance constraints and facilitate modeling the collision probability. Geometrically, landmark regions are bounded hyperplanes. In three dimensions, this hyperplane is a bounded plane and in two dimensions, the constraint is a ray. The plane is bounded because it is bordered by a line ℓ_n incident to an obstacle sub-facet. For each sub-facet $\mathcal{S}$ in an environment, there is a corresponding implied landmark constraint $\mathcal{L}$. The geometry is illustrated in Figs. 1 and 2. The bounded hyperplane extends upwards and in all directions from the blue line in Fig. 1. The hyperplane's orientation divides the angle between the two neighboring obstacle facets. If the angle between the facets is 2α (measured on the side of the obstacle surface with outward pointing normal vectors), then the hyperplane is positioned α from each neighboring facet.

An advantage of using a hyperplane is that it is straightforward to generate convex constraints that force the trajectory to pass through each landmark region. Additionally, landmark regions are implicitly defined for any polytope surface and do not require additional calculation such as other convex decomposition methods [Deits and Tedrake, 2015].

2.2 First Feasible Hybrid Search

First Feasible Hybrid Search (FFHS) specifies how the region planner determines which paths to pass to the trajectory planner. It has the objective of generating good, feasible paths quickly in complex environments. Because trajectory optimization calls are expensive, the search seeks to minimize them. To do this, FFHS only passes *mission complete* paths to the trajectory planner where *mission complete* paths are paths that include all goals in $\boldsymbol{\mathcal{G}}$. This idea is illustrated in Fig. 3. The green path includes $\mathcal{I}$ and $\mathcal{G}$; its implied constraints are passed to the trajectory planner for trajectory optimization. The cyan and red path violated the chance constraint and is separated into prefix and suffix paths.

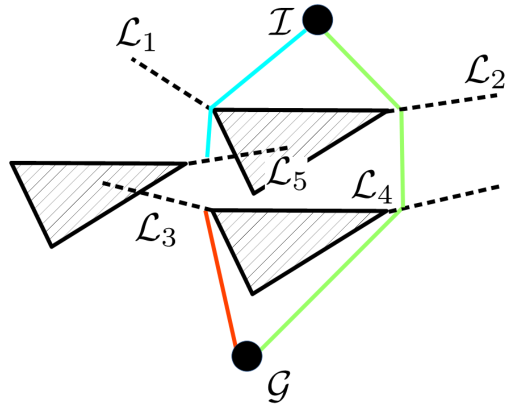

Figure 3: Example of First Feasible Hybrid Search. Mission complete path $\mathcal{P} = [\mathcal{I}, \mathcal{L}_2, \mathcal{L}_4, \mathcal{G}]$ is passed to the trajectory planner, which computes a trajectory constrained to lie in these regions.

3 The Trajectory Planner

The task of the trajectory planner is to optimize a trajectory that respects the path's implied geometric constraints as well as other constraints such as the agent's dynamics. Importantly, the trajectory planner models trajectory risk, Constraint (3). It uses techniques from trajectory optimization and off-the-shelf optimization libraries.

The trajectory planner receives as inputs a sequence of region constraints from the region planner. It requires a model of the agent's dynamics and a predetermined bound on mission failure, i.e. the chance constraint. The full article details three models of trajectory risk, which we briefly summarize here.

3.1 CDF-Based Chance Constraints

The chance constraint models the probability that the trajectory fails in execution. The constraint is difficult to compute because it involves the integration of a probability distribution over many dimensions as the state evolves in time. Our first and simplest model of risk involves CDF-based chance constraints. This simplification is appropriate to agents with linear dynamics whose geometry is unimportant when computing collision risk, i.e. point robots. First, an independence assumption is used to simplify the joint chance constraint, Constraint (3), into a product of probabilities:

$$P(S_N) = \prod_{n=1}^{N} P(S_n | S_{n-1}) \geq 1 - \epsilon \tag{6}$$

Each term on the right-hand side of Eq. (6) models the probability of success at a specific point in time. Mathematically, it is given by the integral:

$$P(S_n) = \int_{\ell} f(\mathbf{x}_n)\, d\mathbf{x}_n \tag{7}$$

where $f(\mathbf{x}_n)$ is a distribution of the agent's state at region n and ℓ represents the limits of integration. In general, Eq. (7) must be computed using numerical methods; however, the central assumption of this section is that we have access to a subroutine that can quickly evaluate the integral. What must be specified are the limits of integration ℓ.

An advantage of Section 2.1's landmark regions is that they form not only constraints on the trajectory but also enable a geometric simplification when computing CDF-based chance constraints. Instead of calculating the integral with respect to the non-convex space, it is simplified by projecting the distribution onto the landmark region. For example, in Fig. 1, state uncertainty is projected onto $\mathcal{L}_n$ and the resulting bivariate distribution integrated from coordinate $\hat{s}_n = 0$ to ∞. For a complete description of the calculation, the reader is referred to the full article [Strawser and Williams, 2022].

3.2 Approximating Trajectory Risk via Intersecting Region

The downside of CDF-based chance constraints is they require modeling uncertainty with a distribution for which there is an easily-evaluated CDF. Our second method approximates risk via samples drawn from the nominal trajectory. If $\mathcal{W}_{obs}$ represents the workspace occupied by the obstacles, let $\mathcal{W}_{int,s}$ be the intersecting region between the agent's body, $\mathcal{W}_{k,s}$, and obstacles summed over the entire trajectory scenario s. Specifically:

$$\mathcal{W}_{int,s} = \sum_k \mathcal{W}_{k,s} \cap \mathcal{W}_{obs} \tag{8}$$

To model the collision probability, an indicator variable is set to one if the intersecting region is greater than zero, and zero otherwise:

$$P(\text{Collision}) \approx \frac{1}{S} \sum_s \mathbb{1}_{|\mathcal{W}_{int,s}|>0} \tag{9}$$

where $|\mathcal{W}_{int,s}|$ is the size of the intersecting region of scenario s and S is the total number of scenarios. The Jacobian of the indicator function is a Dirac delta function; Eq. (9) has a non-zero gradient only at the point of collision between the agent and the obstacle. This provides the trajectory optimizer with little information about how to satisfy the constraint and would likely prevent it from succeeding. Instead, the paper approximates the indicator variable through a sigmoid function. The overall chance constraint is then written:

$$\epsilon - \frac{1}{S} \sum_s \text{Sig}_\alpha(|\mathcal{W}_{int,s}|) \geq 0 \tag{10}$$

where α controls the sigmoid's steepness, i.e. how well it approximates the indicator function. An advantage of the sigmoid function is that it is straightforward to differentiate when computing the Jacobian matrix. Specifically,

$$\frac{d\text{Sig}_\alpha(s)}{ds} = \text{Sig}_\alpha(s)\,(1 - \text{Sig}_\alpha(s)) \tag{11}$$

The chain rule is used to derive the gradients with respect to the decision variables. The Jacobian of the chance constraint C_{cc} with respect to a decision variable state x_n is:

$$\frac{\partial C_{cc}}{\partial x_n} = \frac{d\text{Sig}_\alpha(\mathcal{W}_{int})}{d\mathcal{W}_{int}} \frac{\partial \mathcal{W}_{int}}{\partial x_n} \tag{12}$$

	Lawnmower Pattern (Trajectory Cost)	Lawnmower Pattern (Time)
FFHS	**924**	**556**
CC-RRT	50,750	992
CC-RRT-Connect	5,306	754
Interleaved	Timeout	Timeout

Figure 4: Comparison of the trajectory cost and time to reach an initial feasible solution for a linearized undersea glider test case. A timeout was set at 5 minutes. CC-RRT and CC-RRT-Connect are variants of chance-constrained RRT. "Interleaved" refers to a hybrid search where a trajectory optimization is performed for every new region added to a path.

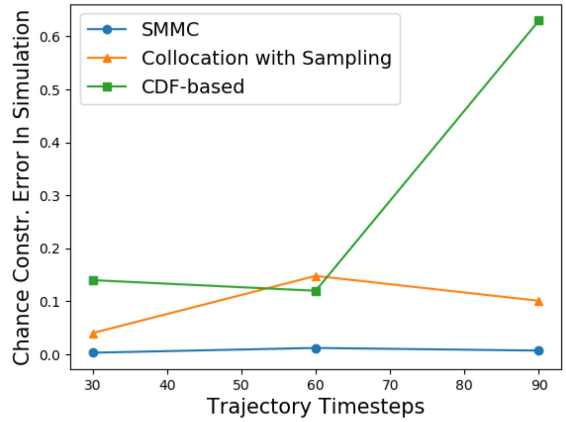

Figure 5: Comparison of the three chance constraint models' ability to approximate the chance constraint in simulation

Computing gradient $\frac{\partial \mathcal{W}_{int}}{\partial x_n}$ is possible through the use of vector calculus and considering the flux of incremental area (or volume) entering or leaving $\mathcal{W}_{int}$ due to an incremental change in one of the states. The journal paper details this calculation more fully.

3.3 Shooting Method Monte Carlo

The previous section's sampling-based chance constraint has the drawback that it requires a nominal trajectory from which to draw samples. Stochastic systems do not have a deterministic nominal trajectory. In this section, we propose a method that computes the joint chance constraint by forward simulating realizations of entire trajectories. We then determine which scenarios fail and use this information to compute the overall trajectory risk. To do this, we incorporate a shooting method trajectory optimization. In the deterministic case, the shooting method works by forward simulating a trajectory and updating the control inputs until the final simulated state matches the goal state. The insight underlying this section is that the shooting method of trajectory optimization is similar to a single Monte Carlo simulation of a stochastic process. We combine the two strategies into the Shooting Method Monte Carlo (SMMC) algorithm.

The vanilla deterministic shooting method first simulates the agent's dynamics. Given the results of the simulations, a defect d_i is calculated for each constraint C_i. The defect is measured as:

$$C_i - \overline{C}_i\left(\mathbf{u}, \mathbf{h}\right) = d_i \tag{13}$$

where C_i is the desired value of constraint i, and $\overline{C}_i\left(\mathbf{u}, \mathbf{h}\right)$ is the value of the constraint resulting from simulating the dynamics using decision variables $(\mathbf{u}, \mathbf{h})$. The task of the trajectory optimization routine is to reduce defects to some small tolerance.

Shooting Method Monte Carlo differs from the traditional shooting method by first sampling from the underlying stochastic process's distribution. The trajectory is then simulated S times and there is no single $\overline{C}_i\left(\mathbf{u}, \mathbf{h}\right)$. Instead, SMMC approximates the expected value of each constraint C_i that results from simulating the trajectory with respect to decision variables $(\mathbf{u}, \mathbf{h})$ and samples $\mathbf{Z}_s$. SMMC's defect is expressed as:

$$C_i - \mathbb{E}_{\hat{f}}\left[\overline{C}_i\left(\mathbf{u}, \mathbf{h}\right)\right] = d_i \tag{14}$$

where $\hat{f}$ is the trajectory's simulated empirical distribution. The expected value in Eq. (14) is:

$$\mathbb{E}_{\hat{f}}\left[\overline{C}_i\left(\mathbf{u}, \mathbf{h}\right)\right] = \frac{1}{S}\sum_{s}\overline{C}_{i,s}\left(\mathbf{u}, \mathbf{h}\right) \tag{15}$$

Approximating the objective function follows a similar scheme. Its value is computed for each scenario, J_s, and the expected cost is approximated as:

$$\hat{J} = \mathbb{E}_{\hat{f}}\left[J\right] = \frac{1}{S}\sum_{s} J_s \tag{16}$$

Given the sampled trajectory, the chance constraint is computed via (10). The chief difference between Shooting Method Monte Carlo and the second method described in Section 3.2 is that SMMC relies on sampled trajectories rather than a nominal trajectory. This allows it to better approximate the true underlying distribution.

4 Results

The paper includes a number of test cases. The first set of tests benchmark FFHS against other approaches on agents with linear dynamics and Gaussian uncertainty. One task requires an undersea glider to perform seafloor exploration via a lawnmower pattern. These results are depicted in Table 4; FFHS is able to outperform the other approaches by generating trajectories with lower cost in less time. We also benchmark the various chance constraint models against one another. CDF-based chance constraints are easiest to compute; however, they exhibit error if the dynamics are nonlinear and state distribution non-Gaussian. This is shown in Fig. 5 where SMMC more accurately approximates trajectory risk in simulation for a Dubins car. The reader is referred to the full paper for complete results and discussion.

Acknowledgments

This work was partially supported by the Exxon Mobil Corporation and the Toyota Motor Corporation. We would like to thank Richard Camilli for his invaluable advice regarding the undersea use case. We would also like to thank the members of the MIT MERS lab and in particular the Scouts Team for their support developing this work. Finally, the journal reviewers' feedback was essential to help focus the article

References

[Blackmore *et al.*, 2011] Lars Blackmore, Masahiro Ono, and Brian C. Williams. Chance-constrained optimal path planning with obstacles. *IEEE Trans. Robotics*, 27(6):1080–1094, 2011.

[Deits and Tedrake, 2015] Robin Deits and Russ Tedrake. *Computing Large Convex Regions of Obstacle-Free Space Through Semidefinite Programming*, pages 109–124. Springer International Publishing, Cham, 2015.

[Faust *et al.*, 2018] Aleksandra Faust, Oscar Ramirez, Marek Fiser, Ken Oslund, Anthony Francis, James Davidson, and Lydia Tapia. PRM-RL: Long-range robotic navigation tasks by combining reinforcement learning and sampling-based planning. In *2018 IEEE International Conference on Robotics and Automation, ICRA 2018, Brisbane*, pages 5113–5120, Brisbane, Australia, 2018.

[Fernandez-Gonzalez *et al.*, 2018] Enrique Fernandez-Gonzalez, Brian C. Williams, and Erez Karpas. Scotty-Activity: Mixed Discrete-Continuous Planning with Convex Optimization. *Journal of Artificial Intelligence Resesearch (JAIR)*, 62:579–664, 2018.

[Luders *et al.*, 2010] Brandon Luders, Mangal Kothari, and Jonathan P. How. Chance-constrained RRT for probabilistic robustness to environmental uncertainty. In *AIAA Guidance, Navigation, and Control Conference (GNC)*, Toronto, Canada, August 2010.

[Ono *et al.*, 2013] Masahiro Ono, Brian C. Williams, and Lars Blackmore. Probabilistic Planning for Continuous Dynamic Systems under Bounded Risk. *Journal of Artificial Intelligence Research (JAIR)*, 46:511–577, 2013.

[Patil *et al.*, 2012] Sachin Patil, Jur van den Berg, and Ron Alterovitz. Estimating probability of collision for safe motion planning under Gaussian motion and sensing uncertainty. In *IEEE International Conference on Robotics and Automation, ICRA 2012, St. Paul, Minnesota, USA*, 2012.

[Strawser and Williams, 2022] Daniel Strawser and Brian C. Williams. Motion Planning Under Uncertainty with Complex Agents and Environments via Hybrid Search. *Journal of Artificial Intelligence Research (JAIR)*, 75:1–81, 2022.