# PROBABILISTIC CONSTRAINTS FOR OPTIMIZATION-BASED MOTION PLANNING

by

Kevin S. Barnard

A thesis submitted to the Faculty and the Board of Trustees of the Colorado School of Mines in partial fulfillment of the requirements for the degree of Master of Science (Robotics).

Golden, Colorado

Date _____

Signed: _____

Kevin S. Barnard

Signed: _____

Dr. Neil T. Dantam
Thesis Advisor

Golden, Colorado

Date _____

Signed: _____

Dr. Kevin Moore
Professor and Robotics Program Director
Interdisciplinary

ABSTRACT

We present a novel method to formulate constraints over probability distributions within optimization-based motion planning, enabling robots to plan for uncertainty in dynamic environments. Typical approaches for motion planning in dynamic environments involve predictions of the environment (e.g., moving obstacles), so that the robot can plan around anticipated environmental changes. Uncertainty in predictions is a challenge that pervades these approaches. As such, properly modeling environmental uncertainty is crucial to obtain valid motion plans. We propose a general method of propagating uncertainty in environmental predictions to a probabilistic constraint in optimization-based motion planning. We demonstrate the efficacy of this approach for uncertainty in predicted human motions in the context of simultaneous manipulation in a shared workspace with a 7-DoF robot arm.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

Cartesian space (special Euclidean group) . . . . . $\mathcal{SE}(3)$

Jacobian . . . . . $\mathbf{J}$

Real set . . . . . $\mathbb{R}$

Integer set . . . . . $\mathbb{Z}$

Configuration space . . . . . $\mathcal{C}$

Free space . . . . . $\mathcal{C}_{\text{free}}$

Obstacle region . . . . . $\mathcal{C}_{\text{obs}}$

Configuration . . . . . $\mathbf{q}$

Initial configuration . . . . . $\mathbf{q}_I$

Goal configuration . . . . . $\mathbf{q}_G$

Motion plan . . . . . $\mathbf{Q}$

Probability measure . . . . . $P$

Probability mass function of random variable $X$ . . . . . $p_X$

Cumulative distribution function of random variable $X$ . . . . . $F_X$

Covariance matrix . . . . . $\mathbf{\Sigma}$

Workspace position . . . . . $\mathbf{x}$

Separation distance . . . . . $\ell$

Average probability of collision . . . . . $\mathcal{P}$

LIST OF ABBREVIATIONS

| | |
|---|---|
| Degree of freedom | DoF |
| Probabilistic roadmap | PRM |
| Rapidly-exploring random tree | RRT |
| Partial Motion Planner | PMP |
| Inevitable collision state | ICS |
| Nonlinear velocity-obstacle | NVLO |
| Gaussian process | GP |
| Hidden markov model | HMM |
| Stochastic ensemble simulation | SES |
| Dynamic Risk Tolerance | DRT |
| Probability mass function | PMF |
| Cumulative distribution function | CDF |
| Flexible Collision Library | FCL |
| Open Motion Planning Library | OMPL |
| Interior Point Optimizer | IPOPT |
| Graphics processing unit | GPU |

# ACKNOWLEDGMENTS

I would like to thank and acknowledge Dr. Neil T. Dantam for advising me and for all his guidance throughout this work. I would also like to thank my family for their perpetual support and encouragement. Furthermore, I would like to acknowledge the Dynamic Automata Lab and its members.

This work was supported in part by the Army Research Laboratory Distributed and Collaborative Intelligent Systems and Technology Collaborative Research Alliance [W911NF-17-2-0181] and the National Science Foundation [CNS-1823245].

# CHAPTER 1

# INTRODUCTION

Throughout the mid-to-late 20th century, robots rarely ventured outside the walls of factories or laboratories. In these highly-controlled settings, the "messiness" of the real world was kept out. Today, robots can be found almost everywhere: delivering your online orders, navigating deep in the ocean, and even roaming other planets. More than ever before, robots are being tasked to work in complicated, uncontrolled, real-world environments. In order to account for this complexity, robots need to be smarter when planning their motions through their environment.

As a basic level, robot motion is constrained by obstacles in the environment, but to avoid obstacles, a robot must know where they are. Some environments are static, where obstacles and constraints do not change (e.g., in position, orientation) over time, so if the robot knows the obstacles' poses (and corresponding free configuration space), it can plan for all future time. However, many real environments are dynamic, where obstacles and constraints change over time. *If a robot can only observe its environment at the current moment, how can it plan ahead for an environment that will be different?*

In a dynamic environment, obstacles occupy different spaces at different times, so robots must know both *where* objects are and *when* they will be there. Thus, the robot must predict the obstacles' future motion, typically using a model of object kinematics or dynamics parameterized by observations of the object over time. In practice, however, we can rarely, if ever, predict the future with full certainty; uncertainty is intrinsic to any real-world observations and predictions. As the uncertainty in a model's environmental prediction increases, robot motion planning may become difficult or impractical without addressing this uncertainty. Many approaches to predict object motion thus use probabilistic representations of predictions (see Chapter 2), which presents the key challenge we address: *How can motion planning incorporate uncertainty in a prediction of the environment?*

*We present a novel method to propagate uncertainty within optimization-based motion planning to robustly find plans in environments with uncertain, dynamic obstacles and constraints.*

The primary contribution of our work is a general approach to construct *probabilistic constraints* for optimization-based motion planning (see Chapter 3). Our secondary contribution is the formulation of probabilistic collision constraints in Chapter 4. We develop a method for optimization-based motion planning to avoid uncertain obstacles in the environment. The core of our method applies the idea of *uncertainty propagation*, a technique to approximate the uncertainty in a random quantity $\beta$ in terms of the estimated uncertainty in a measured quantity $\alpha$ using the Jacobian of a function that maps $\alpha$ to $\beta$ (see subsection 4.1).

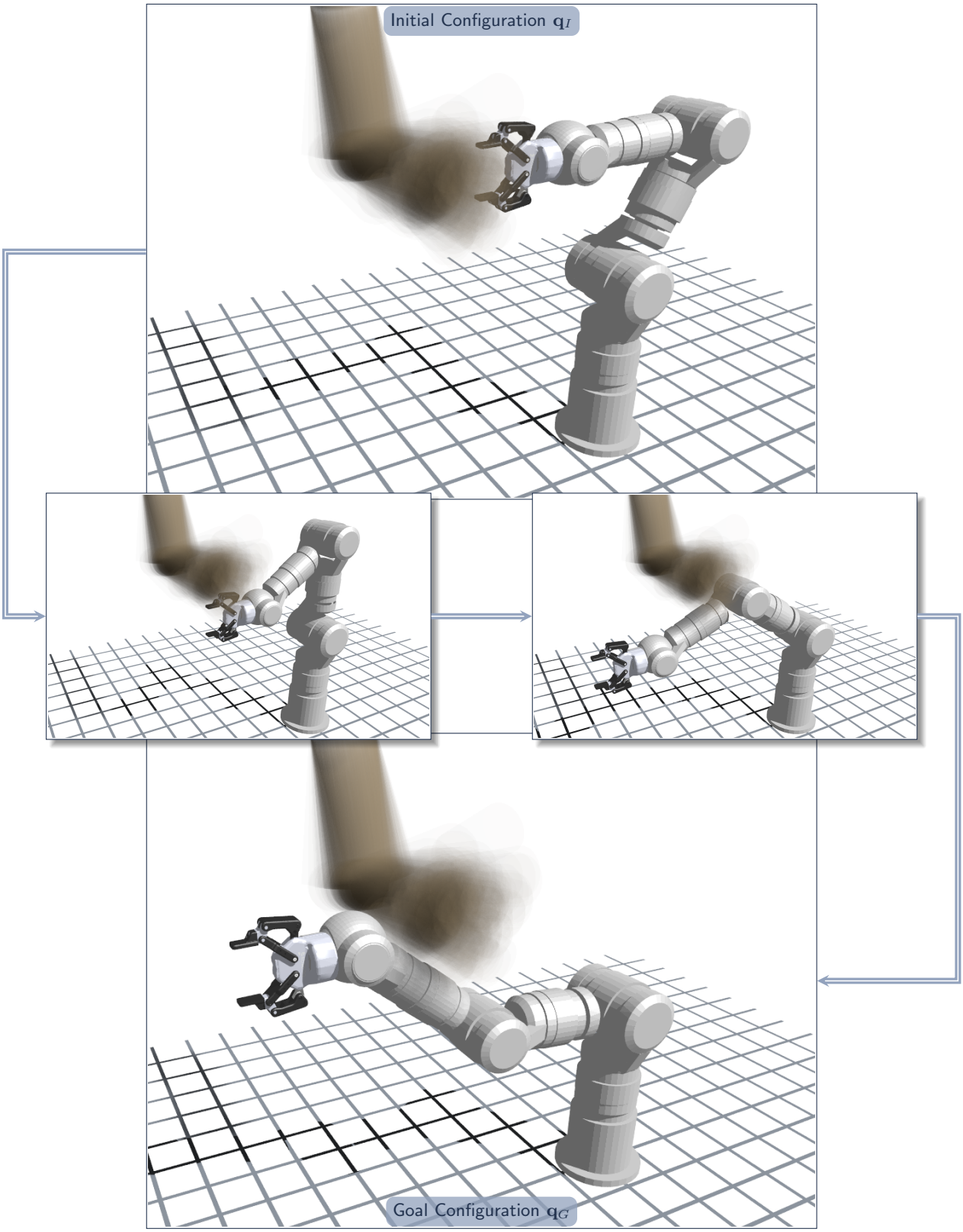



Figure 1.1 A simulated Schunk LWA4D arm planning around uncertainty in predicted human arm motions. The translucent regions around the human represent the uncertainty as samples of the distribution of the human's configurations. The robot limits the probability of collision by propagating uncertainty (covariance) through constraints for motion planning.

We demonstrate our approach in the formulation of a probabilistic human-robot collision avoidance constraint, propagating uncertainty in predicted human joint angles to uncertainty in the separation distance between human-robot link pairs, as seen in Figure 1.1. We evaluate our approach through simulated robot experiments in Chapter 5.

# CHAPTER 2

# RELATED WORK

There is significant prior work regarding robot motion planning in dynamic environments. Most approaches focus on *motion safety*: the perception and active avoidance of dynamic obstacles. We classify and discuss related motion planning algorithms according to the following features [1]:

1. Methodology: the class of underlying motion planning algorithm
2. State vs. State-time: the space in which motion planning occurs
3. Planning Horizon: *global*, *partial*, or *reactive*
4. Obstacle Prediction: how future obstacle motion is modeled

## 2.1 Methodology

The two predominant motion planning methodologies for manipulators are *sampling-based* and *optimization-based* planning.

### 2.1.1 Sampling-based Planning

Sampling-based planning involves repeatedly sampling configurations from a sample space, expanding a search graph to cover that space, then finding a path from a start to a goal configuration. [2] evaluates a probabilistic roadmap (PRM) [3] in a dynamic environment, maintaining the roadmap through online obstacle observations by disallowing transitions to graph vertices (configurations) that would put the robot in collision. [2] further considers the tradeoffs of using a PRM versus a bidirectional rapidly-exploring random tree (RRT) [4, 5]. Though this approach can handle dynamic obstacles by continuously updating the roadmap, it is limited in that it does not have a predictive component. [6] presents a practical strategy for rapid upkeep of PRMs in dynamic environments for online motion planning.

RRTs are widely used for motion planning. Coupled with a probabilistic obstacle prediction model (see subsection 2.4, RRTs can efficiently plan under uncertainty, making them a useful structure for partial or reactive motion planning in dynamic environments [7–12]. In contrast to sampling-based methods, our approach poses constraints for optimization-based planning to optimize a nominal sampling-based plan.

### 2.1.2 Optimization-based Planning

Optimization-based planning involves iteratively minimizing a scalar-valued objective (cost) subject to constraints evaluated over an existing suboptimal motion plan. While sampling-based planning offers

robustness through probabilistic completeness, optimization-based planning are often more efficient in finding optimal and satisfactory motion plans. Sampling-based planning and optimization are often performed sequentially [13]; a sampling-based planner will quickly produce a (suboptimal) motion plan, and an optimization-based planner will modify the plan to achieve local optimality and constraint satisfaction.

With advances in optimization techniques and heterogeneous computing in the context of robotics (e.g., [14]), optimization-based motion planning has become more widely adopted by the robotics community as new formulations of constraints have been developed (e.g., [15–19]).

[20] presents the *Intention-aware motion Planner* (I-Planner), a highly related optimization-based approach. [20] models an upper bound on a probability of collision based on an estimated Gaussian distribution for predicted human motion.

In contrast to these optimization-based methods, however, our approach offers a general formulation of constraints over random variables. Our method is complementary to existing approaches for optimization-based planning, as both probabilistic and deterministic constraints may be applied under our proposed model.

### 2.2 State vs. State-time

The *state* or *state-time* space refers to the space in which motion planning occurs. Typically, the state space is the robot configuration space (e.g., manipulator motion planning) or workspace (e.g., path planning). For a $d$-DoF holonomic robot, the configuration space is typically a subset of $\mathbb{R}^d$ or in the discrete case $\mathbb{N}^d$, and the workspace is typically the six DoF Cartesian space or $\mathcal{SE}(3)$.

The state-time space, initially proposed by [21], is an extension of the state space by adding the time dimension. Fundamentally, state-time space representations require a prediction of future obstacle motion to be formulated. This representation is particularly helpful in dealing with dynamic environments, as constraint dynamics can be represented concretely in a single space. The planning algorithm in [22] is similar to that of [2] in its use of a PRM but plans in the state-time space, allowing for traditional sampling-based planning while maintaining constant-velocity obstacle avoidance. Velocity space planning [23] has also proven fruitful for motion planning in dynamic environments [24, 25]. Our approach is general to any planning space, allowing for any desired remappings to occur through the objective and constraints.

### 2.3 Planning Horizon

The planning horizon is either *global* (full plan from the start to goal), *partial* (partial plan from the start toward a goal), or *reactive* (anytime policy). In the case of dynamic environments, most current approaches are partial planners due to the difficulty in producing robust global plans [1]. For example, [12] presents the

*Partial Motion Planner* (PMP), an efficient RRT-based algorithm for partial planning that fundamentally avoids *inevitable collision states* (ICS) [26]. Some global methods for motion planning in dynamic environments have been attempted; as one may expect, these approaches rely on highly robust models for obstacle prediction. [24] presents the *non-linear velocity obstacle* (NLVO) concept, a mapping from obstacle positions and trajectories to a constrained robot configuration space to allow for rapid global sampling-based motion planning. Our approach is general to any desired planning horizon with the provision that distributions over motion constraints can be computed over that horizon.

### 2.4 Obstacle Prediction

Obstacle prediction methods vary significantly by the motion planning problem subclass. The most common forms of obstacle prediction are velocity extrapolation and dynamical modeling [1]. Learning-based approaches, such as *Gaussian Processes* (GPs) [9, 11], have also been employed for obstacle prediction. [27] employs an artificial neural network for motion planning in a nonstationary environment. Unlike the other approaches in this section, [27] plans without prior knowledge of the dynamic environment, and without any form of explicit search. [10] presents a modified RRT algorithm for partial motion planning problems. This algorithm formulates a probability of collision as a function of an occupancy probability, modeling object motion using a *hidden Markov model* (HMM).

Though model extrapolation and learning-based approaches have proven useful, our work prioritizes models that can also parameterize the uncertainty present in their predictions. Prior approaches to this end vary drastically. [7] makes use of *Stochastic Ensemble Simulation* (SES) in modeling stochastic motions of obstacles in the environment; SES alone, however, can lead to ICS. [8] improves upon [7] with the contribution of *Dynamic Risk Tolerance* (DRT). DRT aims to model the relationship of prediction uncertainty to time, providing a means of balancing short-term and long-term predictions made via stochastic motion modeling. Our approach is general to models of obstacle prediction where distributions over the obstacles can be computed over the planning horizon.

# CHAPTER 3
# PROBABILISTIC CONSTRAINT FORMULATION

We address motion planning problems with uncertainty in the obstacles and constraints. To this end, we aim to provide a general framework to formulate constraints defined over random variables in optimization.

A motion planning problem [28] consists of a configuration space $\mathcal{C}$, an initial configuration $\mathbf{q}_I$, and a goal configuration $\mathbf{q}_G$. The configuration space $\mathcal{C}$ is the union of the disjoint obstacle region $\mathcal{C}_{\text{obs}}$ and free space $\mathcal{C}_{\text{free}}$; both $\mathbf{q}_I, \mathbf{q}_G \in \mathcal{C}_{\text{free}}$.

DEFINITION 1 (Motion Plan). *A motion plan is a trajectory $\tau : [0,1] \mapsto \mathcal{C}_{\text{free}}$ such that for an initial and goal configuration $\mathbf{q}_I, \mathbf{q}_G \in \mathcal{C}_{\text{free}}$, $\tau(0) = \mathbf{q}_I$ and $\tau(1) = \mathbf{q}_G$.*

We provide a visualization of a motion plan in Figure 3.1.

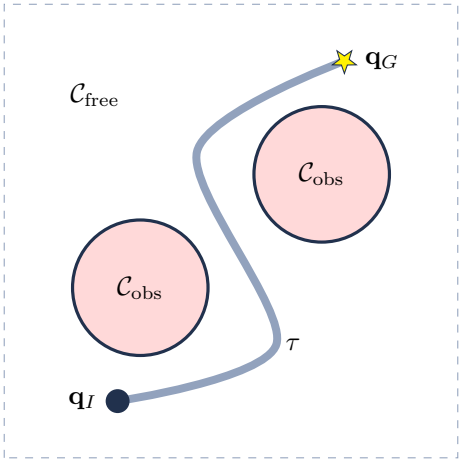



Figure 3.1 Visualization of a motion plan in $\mathbb{R}^2$. $\tau$ is a path from $\mathbf{q}_I$ to $\mathbf{q}_G$ through $\mathcal{C}_{\text{free}}$.

## 3.1 Background: Motion Planning as Optimization

As discussed in Chapter 2, optimization-based motion planning can efficiently handle complex motion constraints. Further, it allows abstraction of the objective function and constraints. For simplicity, we demonstrate our approach via *trajectory discretization*. Other optimization-based planning approaches use continuous representations (e.g. [19]), and our general approach of probabilistic modeling may be extended to these optimization-based planners. We discretize the trajectory $\tau$ into a sequence of "control points" (configurations) over which the objective and constraints are evaluated. To maintain partial continuity of constraint satisfaction, we constrain a maximum distance $\epsilon_{\mathbf{q}}$ between successive configurations. To this end, we will use the following definition for a *discrete motion plan*.

DEFINITION 2 (Discrete Motion Plan). *A discrete motion plan is a sequence of $n$ configurations* $\mathbf{Q} = \left(\mathbf{q}^{\langle 1 \rangle}, \mathbf{q}^{\langle 2 \rangle}, \ldots, \mathbf{q}^{\langle n \rangle}\right)$ *such that for an initial and goal configuration* $\mathbf{q}_I, \mathbf{q}_G \in \mathcal{C}_{\text{free}}$, $\mathbf{q}^{\langle 1 \rangle} = \mathbf{q}_I$ *and* $\mathbf{q}^{\langle n \rangle} = \mathbf{q}_G$*; each* $\mathbf{q}^{\langle k \rangle} \in \mathcal{C}_{\text{free}}$ *for* $k \in \{1 \ldots n\}$ *and* $\left\|\mathbf{q}^{\langle k+1 \rangle} - \mathbf{q}^{\langle k \rangle}\right\| < \epsilon_{\mathbf{q}}$ *for a small value* $\epsilon_{\mathbf{q}}$.

We provide a visualization of a discrete motion plan in Figure 3.2.

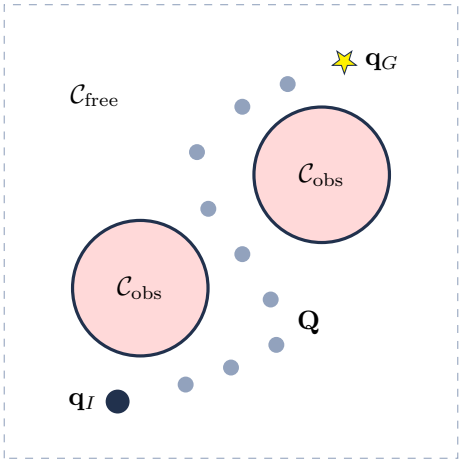



Figure 3.2 Visualization of a discrete motion plan in $\mathbb{R}^2$. $\mathbf{Q}$ is a sequence of configurations in $\mathcal{C}_{\text{free}}$.

Using Definition 2, we restate the motion planning as finding the optimal satisfying sequence of configurations $\mathbf{Q}^*$. We reframe $\mathbf{Q}$ as a stacked vector of decision parameters, where each row of $\mathbf{Q}$ represents each configuration in the motion plan:

$$\mathbf{Q} = \begin{bmatrix} \mathbf{q}^{\langle 1 \rangle} \\ \mathbf{q}^{\langle 2 \rangle} \\ \vdots \\ \mathbf{q}^{\langle n \rangle} \end{bmatrix}. \tag{3.1}$$

With $\mathbf{Q}$ as a single vector, the motion plan is a single parameter to objective function, $f : (\mathcal{C}_{\text{free}})^n \mapsto \mathbb{R}$, (3.2). We further have a set of $n_{\text{ineq}}$ inequality constraints (3.3) and a set of $n_{\text{eq}}$ nonlinear equality constraints (3.4):

$$\mathbf{Q}^* = \arg\min_{\mathbf{Q}} \quad f(\mathbf{Q}) \tag{3.2}$$

$$\text{s.t.} \quad g_i(\mathbf{Q}) \leq x_{g,i}, \quad i \in \{1 \ldots n_{\text{ineq}}\} \tag{3.3}$$

$$h_i(\mathbf{Q}) = x_{h,i}, \quad i \in \{1 \ldots n_{\text{eq}}\} \tag{3.4}$$

To apply the optimization-based approach, we must select an appropriate objective function $f$ and constraints $g_i$ and $h_i$ in order to optimize for a desired trajectory.

### 3.2 Probabilistic Constraints

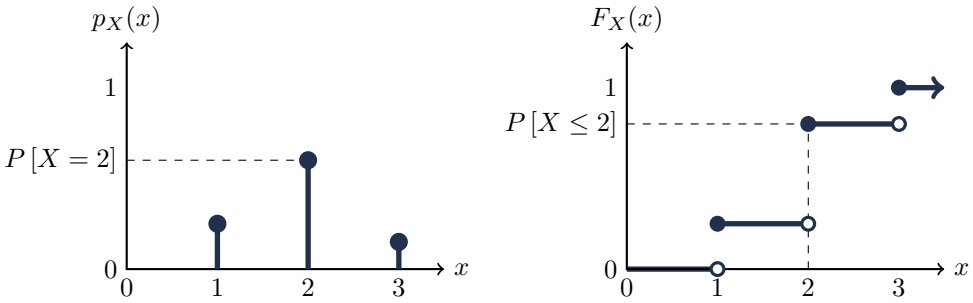
The key difference in our formulation of motion planning is the incorporation of probabilistic constraints within the optimization problem. Optimization-based motion planning typically includes constraints modeled as functions of explicit or derived quantities known at planning-time. For example, constraints for collision avoidance are often modeled using the signed distance function (`sdf`) evaluated on pairs of pertinent bodies in the workspace (i.e., robot links and obstacles) such that the `sdf` is always $\geq 0$.

However, such constraints assume no uncertainty in the positions and orientations of these bodies. In reality, uncertainty is unavoidable when we must sense or *predict* these quantities, so the output of the `sdf` function is uncertain, leaving a possibility for collision. We thus propose a formulation of *probabilistic constraints* in the following manner, enabling planning over uncertainty in random variables such as object poses.

We consider random variable $X$ defined over a set $\mathbb{X}$, where $X$ follows a distribution characterized by probability mass function (PMF) $p_X : \mathbb{X} \mapsto [0, 1]$ and cumulative distribution function (CDF) $F_X : \mathbb{X} \mapsto [0, 1]$,

$$p_X(x) = P\left[X = x\right] \tag{3.5}$$
$$F_X(x) = P\left[X \leq x\right] \ , \tag{3.6}$$

where $P$ is the probability measure. We provide graphs of the PMF and CDF for an example discrete random variable $X$ in Figure 3.3.




(a) PMF $p_X$ (b) CDF $F_X$

Figure 3.3 Graphs of the (a) probability mass function $p_X$ and (b) cumulative distribution function $F_X$ for an example discrete random variable $X$ defined over $\mathbb{X} = \{1, 2, 3\}$.

### 3.2.1 Probabilistic Inequality Constraints

Ordinary inequality constraints restrict a real-valued function $g : \mathbb{X} \mapsto \mathbb{R}$ to be less than or equal to some value $x_g \in \mathbb{X}$,

$$g(X) \leq x_g \ . \tag{3.7}$$

We use the CDF of $g$ to obtain a probability measure for constraint (3.7). Let the CDF of $g$ be $F_g : \mathbb{R} \mapsto [0, 1]$,

$$F_g(x_g) = P\left[g(X) \leq x_g\right] \ . \tag{3.8}$$

Thus, we reformulate the inequality constraint in terms of $F_g(x_g)$—i.e., the probability that the original constraint (3.7) is satisfied. We select a *minimum acceptable probability of satisfaction* $\lambda_g$ and constrain the value of $F_g(x_g)$ accordingly, yielding the general *probabilistic inequality constraint*:

$$F_g(x_g) \geq \lambda_g \tag{3.9}$$

### 3.2.2 Probabilistic Equality Constraints

We similarly formulate an equality constraint under uncertainty by using the PMF in place of the CDF. Ordinary equality constraints restrict a real-valued function $h : \mathbb{X} \mapsto \mathbb{R}$ to be equal to some value $x_h \in \mathbb{X}$,

$$h(X) = x_h \ . \tag{3.10}$$

We obtain a probability measure for constraint (3.10) using the PMF $p_h : \mathbb{R} \mapsto [0, 1]$,

$$p_h(x_h) = P\left[h(X) = x_h\right] \ . \tag{3.11}$$

In this case, $p_h(x_h)$ represents the probability the original constraint (3.10) is satisfied. We similarly select an acceptable probability $\lambda_h$ and constrain $p_h(x_h)$, yielding the general *probabilistic equality constraint*:

$$p_h(x_h) \geq \lambda_h \tag{3.12}$$

## 3.3 Probabilistically-constrained Motion Planning

We now extend the description of optimization-based motion planning (subsection 3.1) and probabilistic constraints (subsection 3.2) to formally define motion planning with probabilistic constraints. Fundamentally, we restate the discrete motion plan optimization problem (3.2)-(3.4) to use *probabilities* over inequality and equality constraint values:

$$\mathbf{Q}^* = \arg\min_{\mathbf{Q}} \quad f(\mathbf{Q}) \tag{3.13}$$

$$\text{s.t.} \quad P\left[g_i(\mathbf{Q}) \leq x_{g,i}\right] \geq \lambda_{g,i}, \quad i \in \{1 \dots n_{\text{ineq}}\} \tag{3.14}$$

$$P\left[h_i(\mathbf{Q}) = x_{h,i}\right] \geq \lambda_{h,i}, \quad i \in \{1 \dots n_{\text{eq}}\} \tag{3.15}$$

Using the CDFs and PMFs for each $g_i$ and $h_i$, we rewrite the problem, we rewrite (3.13)-(3.15) to define probabilistically-constrained motion planning.

DEFINITION 3. *A probabilistically-constrained motion planning problem is given by 3-tuple $(f, G, H)$, where*

- $f : (\mathcal{C}_{\text{free}})^n \mapsto \mathbb{R}$ *is the objective function,*
- $G$ *is the set of cumulative distribution functions $F_{g,i}$ for inequality constraints $g_i : (\mathcal{C}_{\text{free}})^n \mapsto \mathbb{R}$ for $i \in \{1 \dots n_{\text{ineq}}\}$, and*
- $H$ *is the set of probability mass functions $p_{h,i}$ for equality constraints $h_i : (\mathcal{C}_{\text{free}})^n \mapsto \mathbb{R}$ for $i \in \{1 \dots n_{\text{eq}}\}$.*

*The resulting $n$-configuration plan $\mathbf{Q}^*$ must satisfy the following:*

$$\begin{aligned} \mathbf{Q}^* = \arg\min_{\mathbf{Q}} \quad & f(\mathbf{Q}) \\ \text{s.t.} \quad & F_{g,i}(x_{g,i}) \geq \lambda_{g,i}, \quad i \in \{1 \dots n_{\text{ineq}}\} \\ & p_{h,i}(x_{h,i}) \geq \lambda_{h,i}, \quad i \in \{1 \dots n_{\text{eq}}\} \, . \end{aligned}$$

# CHAPTER 4

# MOTION PLANNING OVER OBSTACLE UNCERTAINTY

Next, we construct a probabilistically-constrained motion planning problem (Definition 3) for a set of uncertain obstacles. First, we summarize the general technique of uncertainty propagation. Then, we propagate uncertainty about obstacles to the objective and constraints of the motion planning problem.

We center our example problem around avoiding human-robot collision in a shared workspace. Articulated human motion is difficult to model [29, 30], as it depends on many factors (e.g. intention, environmental stimuli, changing objectives) [20, 31, 32]. As such, predictions of human motion have a high degree of uncertainty, and thus provide a motivating example for probabilistic collision avoidance modeling.

## 4.1 Background: Uncertainty Propagation

Propagation of uncertainty is the core of our approach to construct probabilistic collision constraints. To approximate the uncertainty in some Gaussian quantity $\beta$ that is a function of some other Gaussian quantity $\alpha$, we can *propagate* the uncertainty (covariance) from $\alpha$ to $\beta$. To propagate this uncertainty, we need the Jacobian $\mathbf{J}_\beta$, such that

$$\mathbf{J}_\beta = \frac{\partial \beta}{\partial \alpha} \tag{4.1}$$

Then, we approximate the $\beta$-covariance $\mathbf{\Sigma}_\beta$ by pre-multiplying the $\alpha$-covariance $\mathbf{\Sigma}_\alpha$ by $\mathbf{J}_\beta$ and post-multiplying by $\mathbf{J}_\beta^\intercal$ [33]:

$$\mathbf{\Sigma}_\beta = \mathbf{J}_\beta \mathbf{\Sigma}_\alpha \mathbf{J}_\beta^\intercal \tag{4.2}$$

In our experiments, (4.2) enables us to propagate the uncertainty in the predicted human joint configurations $\mathbf{\Sigma}_h^{\langle k \rangle}$ through to the signed separation distance between human-robot link pairs. As such, we compute a distribution over the distance between any human-robot link pair as a function of the uncertainty in the predicted human motion.

## 4.2 Objective and Constraints

We consider the simple objective to minimize path length. For our vector of configurations $\mathbf{Q}$, we define $f$ as the sum of the $\ell_2$ norms of the successive differences,

$$f(\mathbf{Q}) = \sum_{k=1}^{n-1} \left\| \mathbf{q}^{\langle k+1 \rangle} - \mathbf{q}^{\langle k \rangle} \right\|_2 = \sum_{k=1}^{n-1} \sqrt{\sum_{j=1}^{d} \left( q_j^{\langle k+1 \rangle} - q_j^{\langle k \rangle} \right)^2} \tag{4.3}$$

where $\mathbf{q}^{\langle k \rangle}$ is the $k^{\text{th}}$ configuration in the plan and $d$ is the dimension of the configuration space.

Next, the inequality and equality constraints ensure collision avoidance.

To introduce collision avoidance, we must first formally define collision. Between two *links* (rigid bodies) $\Omega_1, \Omega_2 \in \mathcal{SE}(3)$, we represent collision in terms of the signed distance function $\texttt{sdf} : \mathcal{SE}(3) \times \mathcal{SE}(3) \mapsto \mathbb{R}$, which provides the distance between the two closest points $x_{\Omega_1}, x_{\Omega_2} \in \mathbb{R}^3$, on each link. The signed distance is positive when the two bodies are not in collision, and negative when in collision. Thus, we compute whether two given links are in collision based on the value of the `sdf` function for the pair.

We provide a visualization of this concept in Figure 4.1.

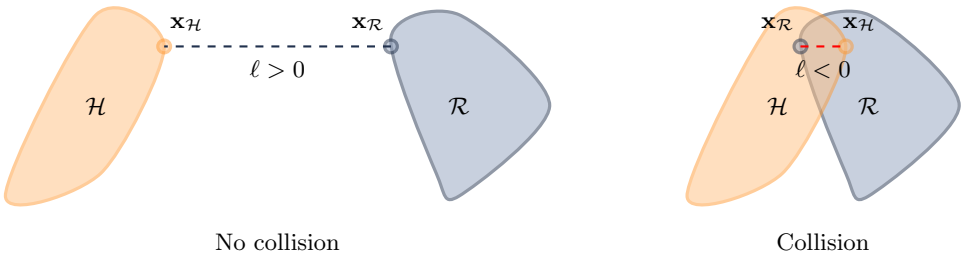



Figure 4.1 Visualization of collision checking. $\texttt{sdf}(\mathcal{R}, \mathcal{H})$ gives (1) $\mathbf{x}_\mathcal{R}$, the point on the body $\mathcal{R}$ closest to $\mathcal{H}$, (2) $\mathbf{x}_\mathcal{H}$, the point on $\mathcal{H}$ closest to $\mathcal{R}$, and (3) $\ell$, the signed distance between these points.

We consider two kinds of collision to avoid: *self-collision* and *human-robot collision*. Self-collision occurs when the manipulator links intersect with each other, whereas human-robot collision occurs when any pair of links from the human and robot intersect. Given a set of robot and human links $\mathcal{R}, \mathcal{H} \subset \mathcal{SE}(3)$, we could enforce non-collision via constraints on the signed distance between relevant links:

$$\texttt{sdf}\left(\mathcal{R}_i, \mathcal{R}_j\right) \geq 0 \quad \forall i, j \in \{1, 2, \ldots, |\mathcal{R}|\} \quad \text{(self-collision)} \tag{4.4}$$

$$\texttt{sdf}\left(\mathcal{R}_i, \mathcal{H}_j\right) \geq 0 \quad \forall i \in \{1, 2, \ldots, |\mathcal{R}|\} \quad \forall j \in \{1, 2, \ldots, |\mathcal{H}|\} \quad \text{(human-robot collision)} \tag{4.5}$$

At a given timestep, this provides us $|\mathcal{R}|^2 + |\mathcal{R}|\,|\mathcal{H}|$ constraints. To ensure no collision over the entire trajectory, we must unroll these constraints over each of the $n$ timesteps, giving a total of $n\,|\mathcal{R}|\,(|\mathcal{R}| + |\mathcal{H}|)$ inequality constraints.

In the case where we know with full certainty the values of $\mathcal{R}$ and $\mathcal{H}$ over all timesteps, these constraints would be sufficient to enforce a collision-free robot trajectory. However, this is rarely the case in practice. Though we may know robot configurations with high accuracy, we often do not know the human's planned trajectory (i.e., $\mathcal{H}$ over all timesteps) at the motion planning stage. Even if we can predict the human's movements, any deviation from the expected human motion could cause human-robot collision.

As such, we aim to provide a method for handling uncertainty inherent to the prediction of human motion. We introduce a *probabilistic* collision constraint, which enables the robot motion plan to anticipate deviation from a nominal prediction of human motion. In effect, this allows us to specify a maximum allowable probability of human-robot collision when planning the robot's motion.

### 4.3 Humans as Uncertain Obstacles

We address robots moving in close proximity to humans though the problem formulation in Definition 3. Human motion presents inherent uncertainty since we cannot directly observe the human's intent (i.e., internal mental state) but only a history of past actions. There is significant prior work on human motion prediction, including work that produces probability distributions of future human motion [31, 34, 35]. Thus, our problem takes the output of such system as a prediction of probability distributions over human motion.

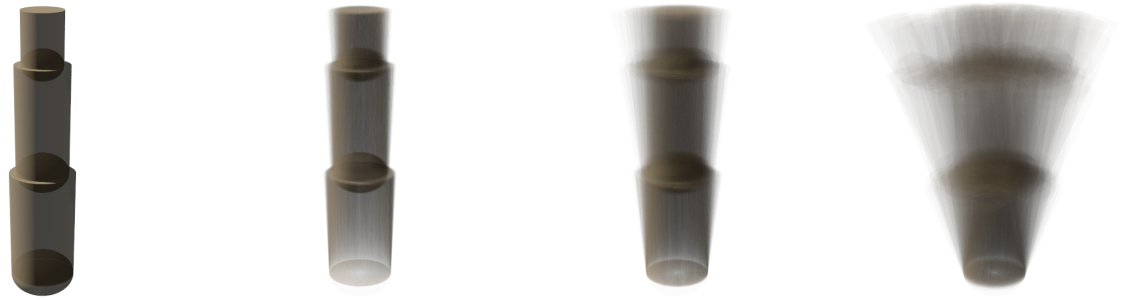
We represent the predicted human trajectory as a discrete motion plan $\mathbf{Q}_h$, of human joint configurations, where each corresponding pair of robot and human configurations $\mathbf{q}_r^{\langle k \rangle}, \mathbf{q}_h^{\langle k \rangle}$ occur at the same time corresponding to timestep $k$:

$$\mathbf{Q}_h = \left(\mathbf{q}_h^{\langle 1 \rangle}, \mathbf{q}_h^{\langle 2 \rangle}, \ldots, \mathbf{q}_h^{\langle n \rangle}\right) \tag{4.6}$$

We model uncertainty in the prediction by considering each human configuration as a random vector $\tilde{\mathbf{q}}_h^{\langle k \rangle}$ that follows a multivariate Gaussian distribution with mean vector $\boldsymbol{\mu} = \mathbf{q}_h^{\langle k \rangle}$ and covariance matrix $\boldsymbol{\Sigma}_h^{\langle k \rangle}$,

$$\tilde{\mathbf{q}}_h^{\langle k \rangle} \sim \mathcal{N}(\mathbf{q}_h^{\langle k \rangle}, \boldsymbol{\Sigma}_h^{\langle k \rangle})\ . \tag{4.7}$$

Our experimental scenarios assume we have predictions of the distributions of human motion in terms of mean $\mathbf{q}_h^{\langle k \rangle}$ and covariance matrix $\boldsymbol{\Sigma}_h^{\langle k \rangle}$ for $k \in \{1 \ldots n\}$. We provide a visualization of human configuration uncertainty in Figure 4.2.



(a) $\sigma_h^2 = 0$ (b) $\sigma_h^2 = 10^{-4}$ (c) $\sigma_h^2 = 10^{-3}$ (d) $\sigma_h^2 = 10^{-2}$

Figure 4.2 Visualization of 100 samples of a human arm primitive collision model with configuration uncertainty, for diagonal joint covariances $\mathbf{\Sigma}_h = \sigma_h^2 I$.

### 4.4 Probabilistic Collision Constraint

Our probabilistic collision constraint replaces the human-robot collision constraint in (4.5). We do so by propagating the uncertainty (4.2) in the predicted human motion through to the signed distance between each human-robot link pair. There are two major computational steps involved in the propagation: *differential kinematics* and *signed distance*.

#### 4.4.1 Differential Kinematics

The human arm is, in essence, a serial manipulator, so we compute the kinematic Jacobian for the human arm given a particular configuration.

$$\dot{\mathbf{x}} = \mathbf{J}_{\mathbf{x}} \dot{\mathbf{q}} \tag{4.8}$$

For each predicted configuration $\mathbf{q}_h^{\langle k \rangle}$, we first compute the mean signed distance between each human-robot link pair. This provides us three values: the nearest point on the human link $\mathbf{x}_{\mathcal{H}}^{\langle k \rangle} \in \mathbb{R}^3$, the nearest point on the robot link $\mathbf{x}_{\mathcal{R}}^{\langle i \rangle} \in \mathbb{R}^3$, and the scalar signed distance between these points $\ell^{\langle k \rangle} \in \mathbb{R}$.

We then compute the kinematic Jacobian $\mathbf{J}_{\mathbf{x}_{\mathcal{H}}}^{\langle k \rangle}$, treating the human-link nearest point $\mathbf{x}_{\mathcal{H}}^{\langle k \rangle}$ as a virtual end-effector. Now, we propagate the the human configuration uncertainty $\mathbf{\Sigma}_h^{\langle k \rangle}$ to obtain the uncertainty in the human-link nearest point:

$$\mathbf{\Sigma}_{\mathbf{x}_{\mathcal{H}}}^{\langle k \rangle} = \left( \mathbf{J}_{\mathbf{x}_{\mathcal{H}}}^{\langle k \rangle} \right) \mathbf{\Sigma}_h^{\langle k \rangle} \left( \mathbf{J}_{\mathbf{x}_{\mathcal{H}}}^{\langle k \rangle} \right)^{\mathsf{T}} \tag{4.9}$$

At this point, we have a random vector $\tilde{\mathbf{x}}_{\mathcal{H}}$:

$$\tilde{\mathbf{x}}_{\mathcal{H}}^{\langle k \rangle} \sim \mathcal{N}_3(\mathbf{x}_{\mathcal{H}}^{\langle k \rangle}, \mathbf{\Sigma}_{\mathbf{x}_{\mathcal{H}}}^{\langle k \rangle}) \tag{4.10}$$

### 4.4.2 Separation Distance

We further propagate the uncertainty in the position of the random human-link nearest point $\tilde{\mathbf{x}}_{\mathcal{H}}^{\langle k \rangle}$ to the (signed) separation distance. We do this in a similar fashion to the differential kinematics.

First, we compute the gradient of the separation distance $\nabla_{\ell}^{\langle k \rangle}$ with respect to the human-link nearest point $\mathbf{x}_{\mathcal{H}}$:

$$\nabla_{\ell}^{\langle k \rangle} = \frac{\partial}{\partial \mathbf{x}_{\mathcal{H}}^{\langle k \rangle}} \left( \mathbf{x}_{\mathcal{H}}^{\langle k \rangle} - \mathbf{x}_{\mathcal{R}}^{\langle k \rangle} \right) \tag{4.11}$$

$$= \frac{\mathbf{x}_{\mathcal{H}}^{\langle k \rangle} - \mathbf{x}_{\mathcal{R}}^{\langle k \rangle}}{\left\| \mathbf{x}_{\mathcal{H}}^{\langle k \rangle} - \mathbf{x}_{\mathcal{R}}^{\langle k \rangle} \right\|_2} \tag{4.12}$$

$$= \frac{\mathbf{x}_{\mathcal{H}}^{\langle k \rangle} - \mathbf{x}_{\mathcal{R}}^{\langle k \rangle}}{\ell^{\langle k \rangle}} \tag{4.13}$$

*Note:* We assume full certainty of the robot configuration and consequently the value of $\mathbf{x}_{\mathcal{R}}$ to be deterministic.

Then, we use (4.2) again to propagate $\mathbf{\Sigma}_{\mathbf{x}_{\mathcal{H}}}^{\langle k \rangle}$ to the variance in the signed distance $\left( \sigma_{\ell}^{\langle k \rangle} \right)^2$:

$$\left( \sigma_{\ell}^{\langle k \rangle} \right)^2 = \nabla_{\ell}^{\langle k \rangle} \mathbf{\Sigma}_{\mathbf{x}_{\mathcal{H}}}^{\langle k \rangle} \left( \nabla_{\ell}^{\langle k \rangle} \right)^{\intercal} \tag{4.14}$$

Finally, we arrive at the signed distance as random variable $\tilde{\ell}^{\langle k \rangle}$ following a univariate Gaussian distribution:

$$\tilde{\ell}^{\langle k \rangle} \sim \mathcal{N} \left( \ell^{\langle k \rangle}, \left( \sigma_{\ell}^{\langle k \rangle} \right)^2 \right) \tag{4.15}$$

It is important to reiterate that the obtained random variable $\tilde{\ell}$ represents an *approximation* of separation distance obtained via first-order Gaussian uncertainty propagation. For large variances $\sigma_{\ell}^2$, this approximation may not hold.

As we have obtained a distribution over the signed distance, we use the Gaussian CDF $\Phi : \mathbb{R} \mapsto \mathbb{R}$ to constrain the probability that the value of the signed distance is less than some value $x$. The standard Gaussian CDF is as follows:

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{t^2}{2}} \, dt \tag{4.16}$$

To compute the Gaussian CDF for a given mean $\mu$ and variance $\sigma^2$, we evaluate $\Phi(\frac{x-\mu}{\sigma})$. In our case, we select $x = 0$ to include values of the `sdf` that represent collision. Our constraint will specify a maximum probability of collision $\lambda \in [0, 1]$.

Putting it all together, our probabilistic collision constraint is as follows:

$$\Phi\left(\frac{-\ell^{\langle k\rangle}_{\mathcal{R}_i,\mathcal{H}_j}}{\left(\sigma^{\langle k\rangle}_{\ell}\right)_{\mathcal{R}_i,\mathcal{H}_j}}\right) \leq \lambda \quad \forall i \in \{1\dots|\mathcal{R}|\} \quad \forall j \in \{1\dots|\mathcal{H}|\} \tag{4.17}$$

This probabilistic collision constraint allows us to construct a probabilistically-constrained motion planning problem that ensures we avoid collisions up to a desired probability.

At this point, it is important to note that this constraint is applied *at each timestep*, rather than over the entire trajectory. Other formulations of probabilistic collision constraints are possible; this formulation was simply designed to demonstrate the direct application of our probabilistic constraint to a traditional, baseline deterministic collision constraint.

# CHAPTER 5

# EXPERIMENTS

We evaluate our probabilistic collision constraint (4.17) over three scenarios and consider metrics of time efficiency and collision avoidance robustness. We benchmark the optimization performance against a baseline deterministic collision constraint (4.5): traditional trajectory optimization with collision avoidance. We run the experiments for three simulated robot manipulators: the Universal Robots UR10 (6-DoF), Schunk LWA4D (7-DoF), and Rethink Robotics Baxter (14-DoF).

Our implementation is written in Python using *FCL* [36] for collision checking, *IPOPT* [37] for sparse nonlinear optimization, *OMPL* [38] sampling-based motion planning of nominal motion plans (RRT-connect), and *Amino* [39] for robot modeling and kinematics. To predict human motion and joint configuration uncertainties, we implement a learning-based approach following [40], using *OpenPose* for 3D skeletal pose estimation [41–43]. We test our software on an Intel® i7-8700K CPU under Linux `5.8.0`.

## 5.1 Evaluation Criteria

We evaluate the time efficiency of our approach in terms of *slowdown* $S$ over a baseline of deterministic constraints—i.e., the ratio of time to plan with probabilistic constraints, $t_P$, over time with deterministic constraints, $t_D$:

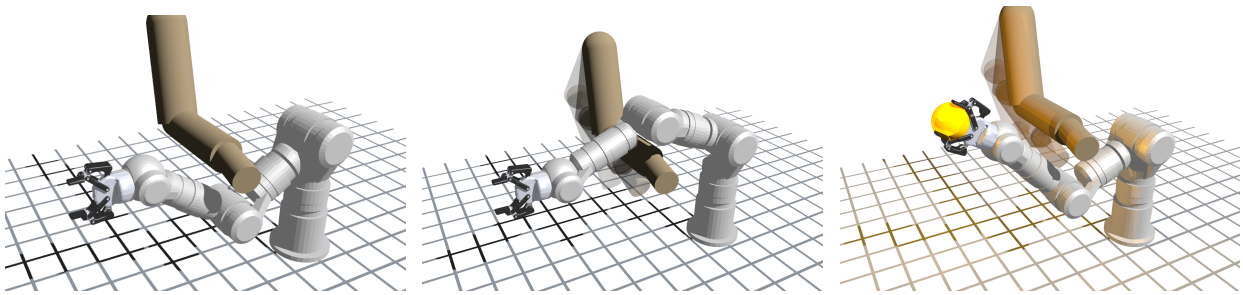
$$S = \frac{t_P}{t_D} \tag{5.1}$$

Slowdown measures the overhead imposed by probabilistic constraints compared to deterministic constraints of prior optimization-based methods.

We evaluate collision avoidance robustness in terms of the average probability of collision $\mathcal{P}$—i.e., the cumulative probability of collision for a discrete motion plan normalized by the number of steps in the plan $n$. For each human link $\mathcal{H}$ and robot link $\mathcal{R}$, we compute the approximate probability of collision at each step and normalize over all steps:

$$\mathcal{P} = \frac{1}{n} \sum_{k=1}^{n} \left( \sum_{i=1}^{|\mathcal{R}|} \sum_{j=1}^{|\mathcal{H}|} \Phi \left( \frac{-\ell^{\langle k \rangle}_{\mathcal{R}_i, \mathcal{H}_j}}{\left( \sigma^{\langle k \rangle}_{\ell} \right)_{\mathcal{R}_i, \mathcal{H}_j}} \right) \right) \tag{5.2}$$

While this metric does not represent the *overall* trajectory collision probability, it was used to give a measure of the approximate collision likelihood at any given time during the trajectory.



(a) Static (b) Dynamic (c) Task-specific

Figure 5.1 Experimental Scenarios. (a) *Static*: the robot cuts close to collision with the human given low uncertainty. (b) *Dynamic*: the robot gives a wider berth due to greater uncertainty. (c) *Task-specific constraint*: the robot places greater effort on avoiding collisions between the human arm and the "hot object" it holds.

### 5.2 Experimental Scenarios

Three experimental scenarios were chosen to highlight the differences that arise from our probabilistic collision constraint as compared to traditional, deterministic collision constraints.

We evaluate our approach for the three scenarios in Figure 5.1.

In the *static* scenario the human does not move. The initial motion plan would collide with the human's arm. Since the human is stationary, our model predicts high certainty (low variance) in the human's predicted motion (see Figure 4.2). Consequently, the robot replans to avoid the human without drastic changes from the nominal motion plan.

In the *dynamic* scenario, the human begins far from the robot. The initial robot motion plan would not collide with the human. As the robot motion is moving, the human swings their arm into the path of the robot. From the detected motion, the prediction reflects uncertainty in human's future positions. As such, the robot must replan to move around the human, this time providing a wider berth than the static scenario due to the increased uncertainty.

The *task-specific constraint* is identical to the *dynamic* scenario, with the modification of unequal collision probability constraints dependent on each link. In this scenario, the robot is carrying a "hot object" that should not come into contact with the human. The probabilistic collision constraints involving this object will have a lower maximum cumulative probability of collision $\lambda$ relative to the robot links themselves. This scenario shows how different robot links may have different probabilistic constraints to achieve a task-specific behavior. In this scenario, the robot replans to keep the object far away from the human, while relaxing the requirement for non-collision between human-robot link pairs.

We find plans with 20 steps $n = 20$, a minimal path length objective $f$ (4.3), and minimum acceptable probability of satisfaction of 95% $\lambda_{g_i} = 0.95$.

In the task-specific scenario with the "hot object", we vary $\lambda_g$ to place an increased requirement on avoiding collision between the human arm and the robot end-effector, while relaxing the constraints on human-robot collision otherwise:

- Relaxed $\lambda_{g_i} = 0.9$ for all $g_i$ between human and robot non-end-effector links
- Increased $\lambda_{g_h} = 0.99$ for all $g_h$ between human links and the robot end-effector link

### 5.3 Sensing and Data

To generate the human configuration predictions $\mathbf{Q}_h$ in parallel with the robot motion, we created a simple experimental setup to collect human motion data during the time period over which the robot motion would occur. To note, our experimental setup was designed simply to collect the necessary data to demonstrate the efficacy of our method. We do not present this experimental setup as part of our approach in applying probabilistic constraints to motion planning.

We used an Intel® RealSense D435 to capture RGB-depth data at 30 Hz of a human performing the motions corresponding to the scenarios described in 5.2. The images and depth data were input to *OpenPose* for 3D skeletal pose estimation [41–43], giving the ground truth human configurations.

We implemented a learning-based motion prediction model following [40]. This model was provided the ground truth human configurations for times up to the beginning of the robot motion plan, and the model output predicted human configurations $\mathbf{q}_h^{\langle k \rangle}$ and covariances $\mathbf{\Sigma}_h^{\langle k \rangle}$ for all $k \in \{1 \dots n\}$ (i.e., $\mathbf{Q}_h$).

For ease of experimentation, we pre-recorded human motion and the resulting predicted configurations and uncertainties. $\mathbf{q}_h$ and $\mathbf{\Sigma}_h$ were unchanged across experimental iterations.

### 5.4 Results and Discussion

To demonstrate the scalability of our approach, we evaluated three simulated manipulators with varying DoF and number of moving links $|\mathcal{R}|$. We present the profiled slowdown ratios for each manipulator in Table Table 5.1. Renderings of the various manipulators are presented in Figure 5.2.

The slowdowns in Table 5.1 come primarily from two computations: differential (Jacobian/gradient) evaluation (4.8), (4.13) and uncertainty propagation (4.9), (4.14). The uncertainty propagation computation of constraint probability distributions involves sevaral matrix multiplications, which are the key source of overhead.

We performed 10 experiments for each probabilistically-constrained motion planning problem per simulated robot manipulator.

Table 5.1 Slowdown ratios of constraint evaluations for manipulators with different DoF and numbers of moving links

| **Manipulator** | DoF | $\lvert\mathcal{R}\rvert$ | $S$ |
|---|---|---|---|
| *Universal Robot UR10* | 6 | 3 | 1.336 |
| *Schunk LWA4D* | 7 | 3 | 1.353 |
| *Rethink Robotics Baxter* | 14 | 6 | 1.465 |

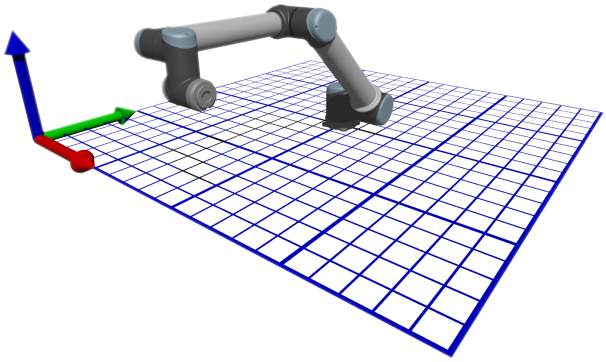
Table 5.2 Average probability of collision for the Static, Dynamic, and Task-specific constraint scenarios evaluated on the Schunk LWA4D

| **Constraint** | **Scenario** | $\mathcal{P}$ |
|---|---|---|
| *Deterministic* (4.5) | Static | 5.24% |
| | Dynamic | 30.62% |
| | Task-specific | 30.51% |
| *Probabilistic* (4.17) | Static | 3.64% |
| | Dynamic | 4.93% |
| | Task-specific | 7.80% |

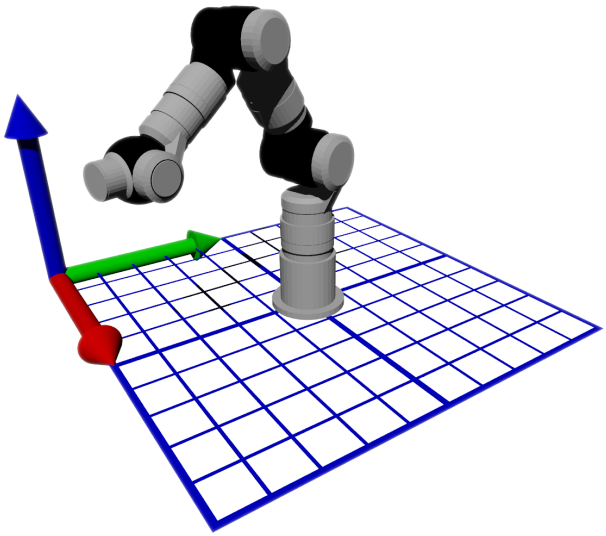
We present averaged measurements of the average probability of collision (5.2) in Table 5.2. Due to the near-deterministic nature of the optimizer and initial conditions, the variance of $\mathcal{P}$ over all experiemntal iterations was negligible.

Table 5.2 shows the strength of our approach; while there may be a relatively high probability of collision, deterministic approaches ignore this factor and thus pose a greater risk of collision. Our probabilistic collision constraint allows for a general specification of an upper bound on collision probability with low computation overhead.

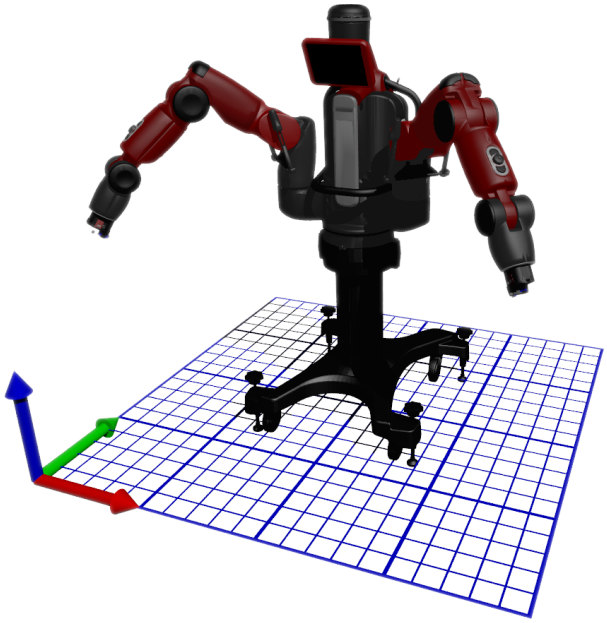
While these results are not sufficient to ensure a collision-free trajectory, they demonstrate the feasibility of satisfying the formulated probabilistic collision constraints with an off-the-shelf optimization solver. To properly gauge the overall continuous collision safety of generated trajectories, further experiments would be necessary.



(a) *Universal Robot UR10*



(b) *Schunk LWA4D*



(c) *Rethink Robotics Baxter*

Figure 5.2 Profiled manipulators.

# CHAPTER 6
# CONCLUSION

We proposed a novel formulation of *probabilistic constraints* for optimization-based motion planning. Probabilistic constraints offer a general method to handle uncertainty in constraints defined over random variables. We presented a *probabilistic collision constraint* that uses the probabilistic constraint formulation, showing how to propagate uncertainty to estimate a Gaussian distribution over constraints via the Jacobian of the deterministic constraint function. Our experimental results compare the probabilistic formulation to a baseline, deterministic collision constraint and show that probabilistic constraints reduce the probability of collision. In our implementation, planning with probabilistic constraints requires approximately 35%-45% more computation time than deterministic constraints.

## 6.1 Impacts

Due to the generality of their formulation, probabilistic constraints may be applied to a wide class of optimization problems. In essence, any constraint which can take the form of a random variable whose distribution is known may now be modeled for use in optimization. If the distribution is unknown but is defined by a function of random variables, we have shown that the distribution can be linearly approximated via Gaussian uncertainty propagation.

The proposed probabilistic constraints may be applied to other domains; constrained optimization problems under uncertainty are ubiquitous in the real world. For example, in economics, the problem of *utility maximization* involves choosing how much of a commodity to consume while constrained by a budget. This problem is often performed under uncertainty in utility, commodity price, and budget, lending toward probabilistic representations of these quantities.

In the context of optimization-based motion planning, probabilistic constraints provide a way to specify a lower bound on the satisfaction of motion constraints that are uncertain in value. In this work, we constructed a probabilistic constraint for collision avoidance, one of the most universal constraints used in robot motion planning. There are, however, many other types of motion constraints that could be modeled under our approach, such as communication uncertainty in teams of networked robots or path planning of an aerial robot subject to uncertain external wind forces.

When robots need to operate in a changing environment, having a predictive model of the environment becomes crucial for proactive planning; however, as robots are introduced to increasingly dynamic environments, exact models of environmental processes may become difficult to produce. As a result, approximate models are often used, introducing uncertainty into predictions of the environment. Our method

aims to bridge the gap between the predicted uncertainty in an environment and how a robot can plan around it.

### 6.2 Future Work

Future directions for this research include the formulation and evaluation of other motion constraints under uncertainty. As a direct application, robot-human handoff tasks could be evaluated using our approach.

Additionally, uncertainty in configurations of the robot itself may affect the workspace precision of an end-effector. In high-precision tasks, such as robot welding, probabilistic collision and end-effector workspace pose constraints that involve the robot uncertainty may bolster performance.

At an implementation level, additional work may be done to improve the computation time of the probabilistic constraints. Hardware acceleration (e.g., on a GPU) has potential to address the slowdown imposed by the probabilistic constraints and reduce computational overhead, as many of the required operations are parallelizable or exhibit otherwise patterned serial execusion.

REFERENCES


[1] Hao-Tien Lewis Chiang, Baisravan HomChaudhuri, Lee Smith, and Lydia Tapia. Safety, challenges, and performance of motion planners in dynamic environments. In *International Journal of Robotics Research*, pages 793–808. Springer, 2020.

[2] Marcelo Kallman and Maja Mataric. Motion planning using dynamic roadmaps. In *International Conference on Robotics and Automation (ICRA)*, volume 5, pages 4399–4404. IEEE, 2004.

[3] Lydia E Kavraki, Petr Svestka, J-C Latombe, and Mark H Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *Transactions on Robotics and Automation*, 12(4): 566–580, 1996.

[4] Steven M. LaValle. Rapidly-exploring random trees: A new tool for path planning. Technical Report TR 98-11, Computer Science Dept., Iowa State University, October 1998.

[5] James J Kuffner and Steven M LaValle. RRT-connect: An efficient approach to single-query path planning. In *International Conference on Robotics and Automation (ICRA)*, volume 2, pages 995–1001. IEEE, 2000.

[6] Tobias Kunz, Ulrich Reiser, Mike Stilman, and Alexander Verl. Real-time path planning for a robot arm in changing environments. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 5906–5911. IEEE/RSJ, 2010.

[7] Hao-Tien Chiang, Nathanael Rackley, and Lydia Tapia. Stochastic ensemble simulation motion planning in stochastic dynamic environments. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 3836–3843. IEEE/RSJ, 2015.

[8] Hao-Tien Lewis Chiang, Baisravan HomChaudhuri, Abraham P Vinod, Meeko Oishi, and Lydia Tapia. Dynamic risk tolerance: Motion planning by balancing short-term and long-term stochastic dynamic predictions. In *International Conference on Robotics and Automation (ICRA)*, pages 3762–3769. IEEE, 2017.

[9] Chiara Fulgenzi, Christopher Tay, Anne Spalanzani, and Christian Laugier. Probabilistic navigation in dynamic environment using rapidly-exploring random trees and gaussian processes. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 1056–1062. IEEE/RSJ, 2008.

[10] Chiara Fulgenzi, Anne Spalanzani, and Christian Laugier. Probabilistic motion planning among moving obstacles following typical motion patterns. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 4027–4033. IEEE/RSJ, 2009.

[11] Chiara Fulgenzi, Anne Spalanzani, Christian Laugier, and Christopher Tay. Risk based motion planning and navigation in uncertain dynamic environment. Research report, INRIA, October 2010. URL https://hal.inria.fr/inria-00526601.

[12] Stéphane Petti and Thierry Fraichard. Safe motion planning in dynamic environments. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 2210–2215. IEEE/RSJ, 2005.

[13] Ryan Luna, Ioan A Şucan, Mark Moll, and Lydia E Kavraki. Anytime solution optimization for sampling-based motion planning. In *International Conference on Robotics and Automation (ICRA)*, pages 5068–5074. IEEE, 2013.

[14] Benjamin Chrétien, Adrien Escande, and Abderrahmane Kheddar. GPU robot motion planning using semi-infinite nonlinear programming. *Transactions on Parallel and Distributed Systems*, 27(10): 2926–2939, 2016.

[15] Nathan Ratliff, Matt Zucker, J Andrew Bagnell, and Siddhartha Srinivasa. CHOMP: Gradient optimization techniques for efficient motion planning. In *International Conference on Robotics and Automation (ICRA)*, pages 489–494. IEEE, 2009.

[16] Mrinal Kalakrishnan, Sachin Chitta, Evangelos Theodorou, Peter Pastor, and Stefan Schaal. STOMP: Stochastic trajectory optimization for motion planning. In *International Conference on Robotics and Automation (ICRA)*, pages 4569–4574. IEEE, 2011.

[17] John Schulman, Yan Duan, Jonathan Ho, Alex Lee, Ibrahim Awwal, Henry Bradlow, Jia Pan, Sachin Patil, Ken Goldberg, and Pieter Abbeel. Motion planning with sequential convex optimization and convex collision checking. *International Journal of Robotics Research*, 33(9):1251–1270, 2014.

[18] Kris Hauser. Semi-infinite programming for trajectory optimization with nonconvex obstacles. In *Workshop on the Algorithmic Foundations of Robotics (WAFR)*, pages 565–580. Springer, 2018.

[19] Mustafa Mukadam, Jing Dong, Xinyan Yan, Frank Dellaert, and Byron Boots. Continuous-time gaussian process motion planning via probabilistic inference. *International Journal of Robotics Research*, 37(11):1319–1340, 2018.

[20] Jae Sung Park, Chonhyon Park, and Dinesh Manocha. I-planner: Intention-aware motion planning using learning-based human motion prediction. *International Journal of Robotics Research*, 38(1):23–39, 2019.

[21] Thierry Fraichard. Trajectory planning in a dynamic workspace: A'state-time space'approach. *Advanced Robotics*, 13(1):75–94, 1998.

[22] Jur P Van Den Berg and Mark H Overmars. Roadmap-based motion planning in dynamic environments. *Transactions on Robotics*, 21(5):885–897, 2005.

[23] Paolo Fiorini and Zvi Shiller. Motion planning in dynamic environments using velocity obstacles. *International Journal of Robotics Research*, 17(7):760–772, 1998.

[24] Frederic Large, Sepanta Sekhavat, Zvi Shiller, and Christian Laugier. Towards real-time global motion planning in a dynamic environment using the NLVO concept. In *International Conference on Intelligent Robots and Systems (IROS)*, volume 1, pages 607–612. IEEE/RSJ, 2002.

[25] Eduardo Owen and Luis Montano. Motion planning in dynamic environments using the velocity space. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 2833–2838. IEEE/RSJ, 2005.

[26] Thierry Fraichard and Hajime Asama. Inevitable collision states—a step towards safer robots? *Advanced Robotics*, 18(10):1001–1024, 2004.

[27] Simon X Yang and Max Meng. An efficient neural network approach to dynamic robot motion planning. *Neural Networks*, 13(2):143–148, 2000.

[28] Steven M LaValle. *Planning algorithms*. Cambridge University Press, 2006.

[29] Jim Mainprice and Dmitry Berenson. Human-robot collaborative manipulation planning using early prediction of human motion. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 299–306. IEEE/RSJ, 2013.

[30] Judith Butepage, Michael J Black, Danica Kragic, and Hedvig Kjellstrom. Deep representation learning for human motion prediction and classification. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6158–6166, 2017.

[31] Judith Bütepage, Hedvig Kjellström, and Danica Kragic. Anticipating many futures: Online human motion prediction and synthesis for human-robot collaboration. *arXiv preprint arXiv:1702.08212*, 2017.

[32] G. Ferrer and A. Sanfeliu. Proactive kinodynamic planning using the extended social force model and human motion prediction in urban environments. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 1730–1735. IEEE/RSJ, 2014.

[33] Kai O Arras. An introduction to error propagation: derivation, meaning and examples of equation CY= FX CX FXT. Technical report, ETH Zurich, 1998.

[34] Sadegh Aliakbarian, Fatemeh Sadat Saleh, Mathieu Salzmann, Lars Petersson, and Stephen Gould. A stochastic conditioning scheme for diverse human motion prediction. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[35] J. M. Wang, D. J. Fleet, and A. Hertzmann. Gaussian process dynamical models for human motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):283–298, 2008. doi: 10.1109/TPAMI.2007.1167.

[36] Jia Pan, Sachin Chitta, and Dinesh Manocha. FCL: A general purpose library for collision and proximity queries. In *International Conference on Robotics and Automation (ICRA)*, pages 3859–3866. IEEE, 2012.

[37] Andreas Wächter and Lorenz T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, March 2006. ISSN 1436-4646. doi: 10.1007/s10107-004-0559-y. URL https://doi.org/10.1007/s10107-004-0559-y.

[38] Ioan A. Şucan, Mark Moll, and Lydia E. Kavraki. The open motion planning library. *Robotics and Automation Magazine (RAM)*, 19(4):72–82, 2012.

[39] Neil T. Dantam. Robust and efficient forward, differential, and inverse kinematics using dual quaternions. *International Journal of Robotics Research*, 2020.

[40] Julieta Martinez, Michael J Black, and Javier Romero. On human motion prediction using recurrent neural networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2891–2900, 2017.

[41] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. OpenPose: Realtime multi-person 2D pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[42] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2D pose estimation using part affinity fields. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[43] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.