Exploring Implicit Spaces for Constrained Sampling-Based Planning

The International Journal of Robotics Research
XX(X):1–33

DOI: 10.1177/ToBeAssigned
www.sagepub.com/
SAGE

Zachary Kingston[1], Mark Moll[1], and Lydia E. Kavraki[1]

Abstract

We present a review and reformulation of manifold constrained sampling-based motion planning within a unifying framework, IMACS (Implicit MAnifold Configuration Space). IMACS enables a broad class of motion planners to plan in the presence of manifold constraints, decoupling the choice of motion planning algorithm and method for constraint adherence into orthogonal choices. We show that implicit configuration spaces defined by constraints can be presented to sampling-based planners by addressing two key fundamental primitives: sampling and local planning, and that IMACS preserves theoretical properties of probabilistic completeness and asymptotic optimality through these primitives. Within IMACS, we implement projection- and continutation-based methods for constraint adherence, and demonstrate the framework on a range of planners with both methods in simulated and realistic scenarios. Our results show that the choice of method for constraint adherence depends on many factors and that novel combinations of planners and methods of constraint adherence can be more effective than previous approaches. Our implementation of IMACS is open source within the Open Motion Planning Library and is easily extended for novel planners and constraint spaces.

Keywords

Constrained Planning, Manipulation Planning, Sampling-Based Motion Planning, Manifold Constraints, End-Effector Constraints, Loop Closure Constraints, Motion Planning

1 Introduction

Motion planning is an essential tool for robotic systems. With motion planning, a robot can generate a feasible path given start configurations and goal specifications (Choset et al. 2005). Recently, there has been rapid development in creating robotic systems that are high-dimensional, such as humanoid robots, mobile manipulators, and mechanically redundant arms. Planning for these high-dimensional systems is hard due to the inherent difficulty of the motion planning problem (LaValle 2006). In addition, these complex systems face increasingly complex manipulation tasks, which require more information beyond start and goal specifications (some example tasks are shown in Figure 1 for NASA's Robonaut 2 (Diftler et al. 2011)). For example, consider a robot that must transfer a glass of water; the robot should be constrained to keep the glass level throughout the entire motion. *Task constraints* encode restrictions on robot motion, and are an important way to concisely specify complex robot motions. Common task constraints include many end-effector constraints and loop-closure constraints, as in parallel manipulators or bi-manual systems carrying an object. An important class of task constraints with special structure are *manifold constraints*, which encapsulate the examples above as well as many geometric constraints. Planning for complex robotic systems in the presence of manifold constraints is an important and relevant problem as these systems face greater manipulation challenges.

Sampling-based planners have proven effective at planning motions for high-dimensional systems (Choset et al. 2005). These planners randomly explore the robot's configuration space and build a discrete representation of valid motions. Many sampling-based planners have been developed with different methods to explore and exploit the valid motions of a robot. However, incorporating manifold constraints in planning is still difficult, as finding configurations that adhere to constraints is a challenging task. Recently,

[1]Department of Computer Science, Rice University, Houston, TX, USA

Corresponding author:
Zachary Kingston and Lydia E. Kavraki
Email: {zak, kavraki}@rice.edu

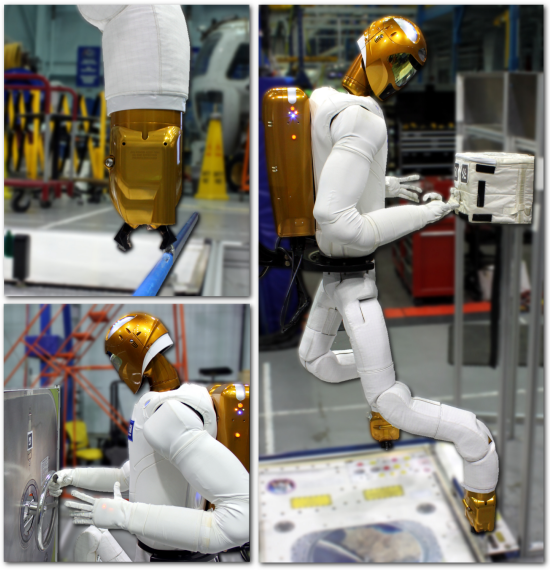


Figure 1. NASA's Robonaut 2 executing geometric constrained manipulation tasks. Clockwise from the top-left: R2's end-effector moves on the Z-axis to grasp a handrail, R2's legs form a closed-chain and a cargo bag is moved linearly from its hold, and R2 turns a valve about its axis.

researchers have developed algorithms for planning with manifold constraints that are effective on realistic problems (Berenson et al. 2011b; Jaillet and Porta 2013b; Kim et al. 2016). Despite their performance, these algorithms are somewhat limited in the sense that they adapt a single sampling-based algorithm to adhere to task constraints by using a specific method for constraint adherence.

1.1 Contributions

The contribution of this paper is a review and reformulation of methods for manifold constrained sampling-based planning within a unifying framework, IMACS (Implicit MAnifold Configuration Space). IMACS enables a broad class of motion planners to plan in the presence of manifold constraints, decoupling the choice of motion planning algorithm and method for constraint adherence. We provide an open source implementation of said framework in the Open Motion Planning Library (OMPL) (Şucan et al. 2012), a widely-used open source library for motion planning, and demonstrate its efficacy on complex simulated and real environments.

The key insight of this work is to view geometrically constrained motion planning as an unconstrained planning problem in an implicitly defined, lower-dimensional space. Sampling-based planners are modular and readily adapted to any space or robotic system. IMACS exploits the modularity of the interface between sampling-based planners and the configuration space to decouple the method of constraint adherence from a motion planner. That is, IMACS represents the implicitly defined space with primitives that are necessary for sampling-based planning. In particular, the primitives within the space representation are augmented samplers and local planners, using some method of constraint adherence. We present projection-based (similar to CBIRRT2 (Berenson et al. 2011b)) and continuation-based (similar to AtlasRRT (Jaillet and Porta 2013b) and TB-RRT (Kim et al. 2016)) methods of constraint adherence as space representations within IMACS.

With IMACS, a broad class of sampling-based planners can use many previously proposed constraint adherence methods and leverage planners and tools developed by the motion planning community. For example, asymptotically optimal planners (Karaman and Frazzoli 2011), heuristic path optimization (Geraerts and Overmars 2007), and domain specific planners for high-dimensional problems (Şucan and Kavraki 2008) all work without modification in IMACS. IMACS also shows that different methods of constraint adherence can all use the same underlying constraint representation. Additionally, we present theoretical results showing that IMACS preserves the probabilistic completeness and asymptotic optimality of sampling-based planners. These theoretical results generalize prior results within the literature. Finally, we demonstrate that different problems can be solved more successfully using novel combinations of planning algorithms and constraint adherence methods compared to existing combinations.

This paper extends the work presented in (Kingston et al. 2017). We present a more complete formulation of the algorithms used by the projection- and continuation-based space representations, proofs that the framework preserves probabilistic completeness and asymptotic optimality of sampling-based planners for both the projection- and continuation-based spaces, an open source implementation of the framework, and additional empirical results including implementation of the framework for NASA's Robonaut 2.

The organization of this paper is as follows. Section 2 defines constraints, the constrained planning problem, and other mathematical background. Section 3 contains a survey of related work for constrained motion planning. IMACS, our framework which decouples constraints from sampling-based algorithms, is presented in Section 4. Theoretical guarantees of IMACS are presented in Section 5. The implementation of IMACS and empirical results, both in simulated problems and on the real Robonaut 2 system, are shown and discussed

in Section 6. Section 7 contains concluding remarks and directions for future work.

2 Preliminaries

This section presents mathematical background for motion planning under manifold constraints, as well as differentiating our framework from related work described in the next section. We first present the motion planning problem and associated terminology. Next, we introduce manifold constraints, the type of constraints we consider, and how they are represented. Finally, we present the constrained motion planning problem as used in this work.

2.1 Motion Planning

This work considers the geometric motion planning problem, or the *generalized movers' problem*, which concerns finding a feasible path of a robot that avoids collision with obstacles. Canonically, the robot is viewed as a single point in an abstract space, the *configuration space* (Lozano-Pérez 1983). The configuration space, or $\mathcal{C}$*-space*, of a robot system is the space defined by all possible *configurations* of that system, denoted as $\mathcal{Q}$. A *configuration* $q \in \mathcal{Q}$ of a robot is a complete specification of the position of every point of that system, with respect to some reference frame. The *degrees of freedom* (DOF) of a robot system is equivalent to the dimension of its configuration, denoted by $n > 0$. Additionally, we assume that the $\mathcal{C}$-space has a *metric* defined between all points, thus the $\mathcal{C}$-space is a *metric space*. Here we assume that $\mathcal{C}$-space is also closed and bounded, and thus a complete metric space. We also assume that $\mathcal{Q}$ is a measurable space $(\mathcal{Q}, \mathcal{B}_\mathcal{Q})$ where $\mathcal{B}_\mathcal{Q}$ is the Borel σ-algebra on the $\mathcal{C}$-space, generated from the metric (Resnick 2014).

For the basic motion planning problem, the robot also needs to avoid some set of *obstacles*. Obstacles form a closed set $\mathcal{Q}_{\text{obs}}$, which in turn defines the free configuration space $\mathcal{Q}_{\text{free}} = \text{cl}(\mathcal{Q} \setminus \mathcal{Q}_{\text{obs}})$, where $\text{cl}(\cdot)$ denotes the closure of a set. We are interested in finding a *curve*, or *path*, from a point q_{start} to some region of interest $Q_{\text{goal}} \subset \mathcal{Q}$, which is a continuous injective map $\sigma : [0,1] \rightarrow \mathcal{Q}_{\text{free}}$ with $\sigma(0) = q_{\text{start}}$, $\sigma(1) \in \text{cl}(Q_{\text{goal}})$ and bounded total variation (Karaman and Frazzoli 2011). Let us denote the set of all paths in $\mathcal{Q}$ as $\Sigma_\mathcal{Q}$, and all collision-free paths as $\Sigma_{\mathcal{Q}_{\text{free}}}$. We also use the notion of δ-clearance from (Karaman and Frazzoli 2011). A configuration $q \in \mathcal{Q}_{\text{free}}$ is defined as a δ-interior configuration of $\mathcal{Q}_{\text{free}}$ if a closed ball of radius $\delta > 0$ centered at q is entirely inside of $\mathcal{Q}_{\text{free}}$. The δ-interior of $\mathcal{Q}_{\text{free}}$, $\text{int}_\delta(\mathcal{Q}_{\text{free}})$ is the set of all δ-interior states. A collision-free path $\sigma \in \Sigma_{\mathcal{Q}_{\text{free}}}$ has *strong* δ*-clearance*, or is robust, if $\sigma(t) \in \text{int}_\delta(\mathcal{Q}_{\text{free}})$ for all $t \in [0,1]$.

Definition 1. Feasible Motion Planning. *Find a robust collision-free path* $\sigma : [0,1] \rightarrow \mathcal{Q}_{free}$ *given a path planning problem* $(\mathcal{Q}_{free}, q_{start}, Q_{goal})$ *if one exists.*

For simplicity of presentation, we consider configurations spaces that are a closed subset of a Euclidean space, i.e., $\mathcal{Q} \subset \mathbb{R}^n$, using the Euclidean metric. We also denote the Lebesgue measure of a set in $\mathcal{Q}$ as $\mu(\cdot)$.

2.1.1 Asymptotically Optimal Motion Planning As discussed in the next section, there has been much recent work on asymptotically optimal motion planning. These algorithms are concerned not only with finding a collision-free path, but also finding a path that is optimal with respect to some cost function $c : \Sigma_\mathcal{Q} \rightarrow \mathbb{R}^+$ that is at least Lipschitz continuous (Janson et al. 2015). Generally, c is the arclength of a path so that algorithms optimize for the shortest path.

Definition 2. Optimal Motion Planning. *Find a robust collision-free path* $\sigma^* : [0,1] \rightarrow \mathcal{Q}_{free}$ *given a path planning problem* $(\mathcal{Q}_{free}, q_{start}, Q_{goal})$ *and cost function* c *such that* $c(\sigma^*) = \min_{\sigma \in \Sigma_{\mathcal{Q}_{free}}} c(\sigma)$ *if one exists.*

2.2 Manifold Constraints

In many cases, avoiding collisions or optimizing a cost function is the only concern for computing a valid path. However, for the constrained motion planning problem presented in this work, we would also like to adhere to some set of constraints on the robot's motion. In particular, we consider constraints that are functions of the robot's geometry—constraints that only use the robot's current configuration $q \in \mathcal{Q}$ to evaluate if they are satisfied—and lower the dimensionality of the problem. These constraints are referred to as *manifold constraints*, which we shorten to just "constraints". Manifold constraints encapsulate many possible path constraints, such as end-effector constraints (e.g., task space regions (Berenson et al. 2011b)), loop-closure constraints, and many more. A brief discussion of non-manifold constraints is given in Section 2.3

We define constraints as *constraint functions* $f_1, \ldots, f_k$, which are k functions $(1 \leq k < n)$ that are at least C^2-smooth, $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$. We consider a constraint function *adhered to* when $f_i(q) = 0$. We denote the composite constraint function $F : \mathbb{R}^n \rightarrow \mathbb{R}^k$:

$$F(q) = \begin{bmatrix} f_1(q) \\ \vdots \\ f_k(q) \end{bmatrix}$$

We say that F is adhered to when $F(q) = \mathbf{0}$, where $\mathbf{0} \in \mathbb{R}^k$ is a vector of all zeros. In practice, F is adhered to when $\|F(q)\|_2 < \epsilon$, where $\epsilon > 0$ is a small tolerance. Constraint functions can also be concatenated together, $F(q) = \begin{bmatrix} F_1(q) & F_2(q) & \dots \end{bmatrix}^T$.

As F is a concatenation of at least C^2-smooth functions, the first derivative can be taken at a point $q \in \mathbb{R}^n$, $J : \mathbb{R}^n \to \mathbb{R}^{k \times n}$ (the *Jacobian*):

$$\nabla_q F(q) = J(q) = \begin{bmatrix} \frac{\partial f_1}{\partial q_1} & \cdots & \frac{\partial f_1}{\partial q_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_k}{\partial q_1} & \cdots & \frac{\partial f_k}{\partial q_n} \end{bmatrix}$$

We assume that $J(q)$ is of full rank when $F(q) = \mathbf{0}$, i.e., the gradients of the constraint functions are not linearly dependent at q. Note that the pseudoinverse of $J(q)$ is a continuous function when $J(q)$ is of full rank, a fact used in Section 5.1.

Thus, the problem of constrained motion planning results in finding a path in an implicitly defined constrained configuration space:

$$\mathcal{M} = \{q \in \mathcal{Q} \mid F(q) = \mathbf{0}\}$$

Importantly, as constraint functions are smooth and have a Jacobian of full rank when $F(q) = \mathbf{0}$, $\mathcal{M}$ is an $(n-k)$-dimensional smooth submanifold of $\mathbb{R}^n$ via the implicit function theorem (Gomes et al. 2009). Additionally, $\mathcal{M}$ is a closed set as it is the preimage of a closed set by a continuous function, and as the configuration space is bounded, is thus compact. We refer to $\mathcal{M}$ as the *constraint manifold*, and denote $\mathcal{M}_{\text{free}} = \mathcal{M} \cap \mathcal{Q}_{\text{free}}$ as the free constraint manifold. The *codimension* of the manifold, i.e., the number of constraints, is denoted as k. Note the constraint manifold is of lower dimension than the ambient configuration space and is smooth, and as such is of measure zero with respect to the ambient configuration space, i.e., $\mu(\mathcal{M}) = 0$ (Lee 2003).

The fact that $\mathcal{M}$ is a manifold is leveraged by the different methods for constraint adherence within IMACS. $\mathcal{M}$ also is a metric space, using the metric induced from the configuration space, and a measurable space $(\mathcal{M}, \mathcal{B}_{\mathcal{M}})$ where $\mathcal{B}_{\mathcal{M}}$ is the Borel σ-algebra generated from the induced metric. Let us denote the set of all paths in $\mathcal{M}$ as $\Sigma_{\mathcal{M}}$, and all collision-free paths as $\Sigma_{\mathcal{M}_{\text{free}}}$. We define a configuration $q \in \mathcal{M}_{\text{free}}$ as a δ-interior configuration of $\mathcal{M}_{\text{free}}$ if a closed ball of radius $\delta > 0$ centered at q is entirely inside of $\mathcal{M}_{\text{free}}$. Note that, as we are using the induced metric from the configuration space, all δ-interior configurations of $\mathcal{M}_{\text{free}}$ are also δ-interior for $\mathcal{Q}_{\text{free}}$. Similar to the case for $\mathcal{Q}_{\text{free}}$, the δ-interior of $\mathcal{M}_{\text{free}}$, $\text{int}_\delta(\mathcal{M}_{\text{free}})$ is the set of all δ-interior states and a collision-free path $\sigma \in \Sigma_{\mathcal{M}_{\text{free}}}$ has strong δ-clearance if $\sigma(t) \in \text{int}_\delta(\mathcal{M}_{\text{free}})$ for all $t \in [0,1]$. Note that q_{start} and Q_{goal} must be path connected in $\mathcal{M}_{\text{free}}$ for a valid solution to be found. Thus, the constrained motion planning problem is finding a path $\sigma : [0,1] \to \mathcal{M}_{\text{free}}$ from a point $q_{\text{start}} \in \mathcal{M}_{\text{free}}$ to some region of interest $Q_{\text{goal}} \subset \mathcal{M}_{\text{free}}$ such that the path is collision-free and for all $t \in [0,1]$, $F(\sigma(t)) = \mathbf{0}$. We can now define two forms of the constrained motion planning problem:

Definition 3. Constrained Motion Planning. *Find a robust collision-free path $\sigma : [0,1] \to \mathcal{M}_{free}$ given a path planning problem $(\mathcal{Q}_{free}, q_{start}, Q_{goal})$ and constraint function F if one exists.*

Definition 4. Constrained Optimal Motion Planning. *Find a robust collision-free path $\sigma^* : [0,1] \to \mathcal{M}_{free}$ given a path planning problem $(\mathcal{Q}_{free}, q_{start}, Q_{goal})$, constraint function F, and cost function c such that $c(\sigma^*) = \min_{\sigma \in \Sigma_{\mathcal{M}_{free}}} c(\sigma)$ if one exists.*

This representation of constraints encompasses a broad set of constraints as presented in the literature. For example, consider a simple point robot in $\mathbb{R}^3$. The constraint function $f_1(q) = \|q\|_2 - 1$ implicitly defines the surface of the unit sphere, a two-dimensional manifold in $\mathbb{R}^3$. We can also consider a simple manipulator robot with n joints (a $\mathcal{C}$-space in $\mathbb{R}^n$) with forward kinematics to the end-effector given by f By defining a constraint function $f_1(q) = \|f(q)_x - c\|_2$ we constrain the end-effector to some YZ-plane.

Manifold constraints are also known as *holonomic constraints* (Laumond 1986; Latombe 1991), as they are integrable functions of the robot's configurations. Note that holonomic constraints do not fundamentally change the motion planning problem.Sometimes, the structure of constraints allows for direct reparameterization. One type of constraints that allows for reparameterization are explicit constraints—constraints that explicitly determine the value of some configuration parameters, e.g., a manipulator grasping an object determines the objects configuration from the manipulator's forward kinematics. Under explicit constraints, the dependent configuration parameters are not necessary. Another example of reparameterization is any articulated robot. Typically, articulated robots are parameterized by their joint angles. It is also possible to model an articulated robot as a constrained system—each of the links of the manipulator is a free-flying rigid body (with $SE(3)$ as its $\mathcal{C}$-space), and the constraints imposed by the mechanical linkage of the joints introduce a lower-dimensional configuration space. The lower-dimensional space induced by the

constraints is equivalent to the space of joint angles. However, it is not always possible to reparameterize the configuration space according to a constraint function. The framework presented in this paper provides a way to plan in the implicitly defined configuration space without reparameterization. We show an example of constrained motion planning for both implicitly defined kinematic chains as well as a realistic robot (shown in Section 6.3 and 6.5).

Differential geometry is core to the representation used in this work, which can broadly be described as the study of smooth manifolds with some geometric structure. A comprehensive overview can be found in (Spivak 1999; Lee 2003). Additional definitions are provided in Sections 4 and 5 as needed.

2.3 Volume-Reducing Constraints

In this work we do not discuss constraints other than manifold constraints in detail. A brief discussion of volume-reducing constraints and their relation to the methods presented in this paper is given here.

Closely related to manifold constraints are volume-reducing constraints, or inequality constraints. These are constraints of the form $G(q) \leq \mathbf{0}$. There are many useful constraints that naturally are formulated as inequalities, e.g., valid workspace volumes for an end-effector or balance constraints for a humanoid robot. But, inequality constraints do not introduce a lower-dimensional manifold as manifold constraints do. Assuming similar properties of G as in the constraint functions above, inequality constraints instead define an n-dimensional region within the configuration space:

$$V = \{q \in \mathcal{Q} \mid G(q) \leq \mathbf{0}\}$$

This volume either is of non-zero volume or is empty due to being unsatisfiable. For some regions of non-zero volume, rejection sampling and standard unconstrained planning techniques can be used. This is similar to what is done in some constrained sampling-based techniques that "inflate" the manifold into a volume by loosening the tolerance ϵ (e.g. (Bonilla et al. 2017)). If the inequality constraint is violated (i.e., $G(q) > \mathbf{0}$), the inequality constraint be folded into the set of manifold constraints in an active set method, using $G(q) = \mathbf{0}$ (Nocedal and Wright 2006). Some projection-based methods (Section 4.2) have demonstrated this method is effective in practice (Hauser et al. 2008; Berenson et al. 2011b). Note that the theoretical properties of the projection operator still hold when using inequality constraints as a manifold constraints. A discussion of continuation-based methods (Section 4.3) with inequality constraints is left as future work. However, it should be possible to apply continuation-based approaches to volume-reducing constraints—Henderson (2002) demonstrated that a continuation-based approach can cover $\mathbb{R}^n$, but this has not been shown for general constraints.

3 Related Work

In this work we study sampling-based planning in the presence of manifold task constraints. There is a wide breadth of literature concerning techniques to plan motions, represent constraints, and adhere to constraints. Using task constraints to specify the motion of a robotic system has its roots within industrial control (Mason 1981; Khatib 1987). Task constraints on robot motion can be used to specify many useful manipulation tasks (Siméon et al. 2004), generate valid motion for parallel manipulators and closed chains (Tsai 1999; LaValle 2006) and even model proteins in structural biology (Zhang and Hauser 2013).

The first applications of geometric constraints to planning in low-dimensional spaces were reduced to problems of finding geodesics on polyhedral structures (Mitchell et al. 1987), similar to finding shortest paths of visibility graphs (Asano et al. 1985; Alexopoulos and Griffin 1992). Most early work with task constraints did not focus on geometric constraints and was directed at non-holonomic constraints, such as differential drive cars (Barraquand and Latombe 1993). However, as planners were applied to more complex high-dimensional systems with more interesting manipulation tasks, geometric constraints were revisited as a difficult addition to the motion planning problem; constraints within higher dimensions (particularly those for articulated mechanisms) require specialized constraint adherence techniques.

3.1 Other Methods

While not the focus of this paper, a short survey of other methods for planning in the presence of manifold constraints is given for completeness.

For end-effector constraints, one approach is to plan a path for the end-effector, so constraints can be directly evaluated and adhering poses can be sampled. After planning a path within the workspace, a corresponding path in the robot's configuration space is generated using inverse kinematics (IK) (Sentis and Khatib 2005; James et al. 2015; Rakita et al. 2018). However, these methods may not be efficient as re-planning is required if a computed path cannot be mapped into the configuration space of the robot. Completeness is also not guaranteed unless all feasible IK solutions can be generated from any given constraint. Additionally,

if multiple end-effector constraints are placed upon the system, null-space projection methods (Sentis and Khatib 2005) will not yield a probabilistically complete algorithm.

Another approach that operates within the robot's workspace is IK-based reactive control which uses convex optimization to find constraint adhering motions, e.g., those used at the DARPA Robotics Challenge (Fallon et al. 2015; Johnson et al. 2015; Atkeson et al. 2015). While effective with operator supervision, these controllers are usually incomplete and are prone to reaching local minima created by the interaction of the objective function and constraints. As local controllers are optimization-based methods, hard geometric constraints are relaxed into soft cost-based constraints, and invalid motions can be generated.

Trajectory optimization approaches (e.g., (Zucker et al. 2013; Schulman et al. 2014)) optimize within trajectory space and are effective for everyday manipulation tasks, but suffer from many of the same shortfalls as reactive control. However, trajectory optimization methods can generate motions that adhere to task constraints by either adding the constraint as a penalty to optimization objective or as an equality constraint. Recently, Bonalli et al. (2019) presented a method for trajectory optimization on an implicitly defined manifold. No comprehensive comparison of constrained non-sampling-based methods to sampling-based methods exists, and a thorough analysis is outside the scope of this paper.

3.2 Sampling-Based Planning

The key idea of sampling-based planning is to avoid computing the free configuration space $\mathcal{Q}_{\text{free}}$ by instead sampling the $\mathcal{C}$-space. Sampling-based planners fall broadly into two categories: multi-query planners which are generally graph-based methods such as PRM (Kavraki et al. 1996) and single-query planners that are generally tree-based methods such as EST (Hsu et al. 1999) or RRT (LaValle and Kuffner 2001).

Multi-query methods construct a "roadmap" within the configuration space that can be queried multiple times. Single-query methods build a tree of motions rooted from the start or goal. Many techniques perform a bidirectional search for efficiency (e.g., (Kuffner and LaValle 2000)) or use coverage estimates to bias search towards unexplored space (e.g., (Şucan and Kavraki 2008)). For a more in-depth review of sampling-based planning see (Choset et al. 2005; LaValle 2006; Elbanhawi and Simic 2014; Kavraki and LaValle 2016).

3.2.1 Asymptotically Optimal Sampling-Based Planning

While sampling-based planners have been shown to be very efficient in finding feasible paths, paths often have poor quality with respect to a given cost function. One approach to improve path quality is to post-process and locally optimize paths with heuristic methods (Geraerts and Overmars 2007) or a trajectory optimization-based approach (Dai et al. 2018). Sampling-based algorithms can also provide asymptotic optimality guarantees (Karaman and Frazzoli 2011; Janson et al. 2015; Gammell et al. 2015). These methods guarantee that the solution path converges to a globally optimal path for a given cost function. If the connection radius is greater than the threshold, then asymptotic optimality is guaranteed. Note that in practice smoothing and post-processing techniques typically generate solutions comparable to paths from asymptotically-optimal planners (Luna et al. 2013; Luo and Hauser 2014).

3.3 Constrained Sampling-Based Planning

Constraints introduce a new element of difficulty to the problem: configurations must adhere to the constraint function. The approaches to handling constraints within a sampling-based framework can be organized into a spectrum of the complexity used to compute adhering configurations. As described in Section 2, the constraint function implicitly defines a manifold within the configuration space. The *method of constraint adherence* is the way a particular method copes with planning in this lower dimensional space. For this paper, of particular importance are projection- and continuation-based method:

- *Projection*: Finding configurations that adhere to the constraint function $F(q) = \mathbf{0}$ requires finding solutions to the constraint's system of equations. A *projection operator* takes a configuration and maps it onto the implicit manifold.
- *Continuation*: From a known adhering configuration, a tangent space of the implicit manifold can be generated. Adhering configurations and valid local motions can be generated by applying projection to configurations sampled within the tangent space. Tangent spaces can be composed together to create a piecewise-linear approximation of the manifold, or a *continuation*. This approximation can then be used for sampling adhering configurations or for local planning.

A more detailed survey of constrained motion planning techniques along with other classes of methods can be found in (Kingston et al. 2018).

3.3.1 Projection-Based Methods

In a constrained motion planning problem, a path only contains configurations that adhere to the constraint function. Projection-based methods use a *projection operator* to

find adhering configurations starting from potentially invalid configurations, defined formally in Section 4.2.

The first projection methods capable of solving constrained problems dealt with specialized cases of constraints—specifically, loop closures in parallel manipulators. Planning for loop closure problems in robotics was first solved with PRM variants using active/passive chain methods (Han and Amato 2000; Yakey et al. 2001; Cortés et al. 2002), and were also relevant in structural biology (Wedemeyer and Scheraga 1999). Active/passive chain methods use IK as a projection operator to join the passive chain to the active chain, closing the loop and creating an adhering configuration. Typically, IK algorithms are numerical methods that require computation of manipulator Jacobian (pseudo-)inverses, but methods such as Cyclic-Coordinate Descent (Canutescu and Dunbrack 2003) and FABRIK (Aristidou and Lasenby 2011) are Jacobian free.

The idea of projection to adhere to constraints was applied to general end-effector constraints in (Yao and Gupta 2005). Task Constrained RRT (Stilman 2010) further generalized the idea of constraints and utilized Jacobian gradient descent (Buss 2004) for projection. More recently, CBIRRT2 (Berenson et al. 2011b) and the motion planner implemented for the Humanoid Path Planner (HPP) System (Mirabel et al. 2016) utilize projection with general constraints and can solve complex combinations of constraints. HPP combines explicit constraints and implicit manifold constraints into a more effective and efficient projection routine (Mirabel and Lamiraux 2018).

Projection methods have been extended to find low-cost paths GradienT-RRT (Berenson et al. 2011a) (note that this is not an asymptotically optimal motion planner), and to asymptotically optimal planners (albeit without formal guarantees) in (Jaillet and Porta 2013a). We show how IMACS emulates projection-based methods (such as CBIRRT2 and other previous approaches) in Section 4.2.

3.3.2 Continuation-Based Methods Projection, while effective at adhering to constraints, utilizes very little information from the constraint. It is possible to locally approximate the manifold defined by the constraint using a *tangent space* of an adhering configuration. In this case, the tangent space is a chart of the manifold, locally parameterizing the neighborhood around a configuration. The tangent space can be used to generate new configurations that are close to the manifold, and close to the original adhering configuration. As the complexity of the constraint manifold approximation increases, sampling in the tangent space becomes more accurate at the price of increased computational cost per sample.

Projection from tangent spaces was utilized within the work of (Yakey et al. 2001) to generate nearby samples, which are then projected to adhere to the constraint. Tangent spaces have been used by (Weghe et al. 2007; Stilman 2010) for manipulators under general end-effector constraints. The technique has also seen many applications in curve tracking constraints for redundant manipulators (Oriolo and Vendittelli 2009; Vendittelli and Oriolo 2009; Cefalo et al. 2013) and structural biology to generate valid motions of proteins with loop closures (Zhang and Hauser 2013; Pachov and van den Bedem 2015; Fonseca et al. 2018).

Rather than discarding a tangent space after computation, some methods assemble the collection of tangent spaces into an *atlas* of the manifold. The atlas is composed of many charts (in this case tangent spaces), a concept borrowed from the definition of differentiable manifolds (Spivak 1999). To be precise, the atlas is defined as a piece-wise linear approximation of the constraint manifold using tangent spaces, which fully cover the manifold (Henderson 2002). These methods are derived from numerical continuation techniques, which are designed to compute solutions to nonlinear systems of equations (e.g., the constraint manifold). The key difference between continuation methods (Henderson 2002) and continuation-based planners is the incremental construction of the atlas interleaved with space exploration, allowing the planner to explore online or reuse results from previous runs.

AtlasRRT (Jaillet and Porta 2013b) and TB-RRT (Kim et al. 2016) both construct an atlas by incrementally building the set of tangent spaces that approximate the manifold. TB-RRT evaluates the manifold lazily and does not separate tangent spaces, leading to overlap and potential problems with invalid points. AtlasRRT computes halfspaces to separate tangent spaces into tangent polytopes to guarantee uniform coverage in the limit at additional computational cost. AtlasRRT has been extended to an asymptotically optimal algorithm AtlasRRT* (Jaillet and Porta 2013a) and to kinodynamic planning (Bordalba et al. 2018). Like projection-based methods, both TB-RRT and AtlasRRT are emulated within IMACS, shown in Section 4.3.

3.3.3 Other Methods There are many other approaches to constrained sampling-based planning. One class of methods are reparameterization-based approaches which are similar in spirit to IMACS—they create a new representation of the space that adheres to constraints. However, in the case of "Deformation space" (Han et al. 2008) and "Reachable volume space" (McMahon 2016), this representation is an

explicit transformation for a particular class of robot and constraint. IMACS, on the other hand, does not explicitly reparameterize the manifold, and instead provides a means of representing the implicitly defined space.

Additionally, there are offline sampling-based methods which construct an approximation of the constraint manifold offline. That is, these methods precompute a set of constraint-adhering configurations, and then use this set for sampling and local planning. A general approach for offline computation for online sampling was described in (Şucan and Chitta 2012). This kind of approach was employed by (Burget et al. 2013) to adhere to balance constraints on a humanoid robot.

3.4 Beyond Single Constraints

Motion planning with manifold constraints is closely tied to manipulation planning. Manipulation planning, as opposed to motion planning, requires planning a sequence of actions, each of which has different path constraints on the motion of the robot (Siméon et al. 2004). For example, consider a robot that must transfer a cup from one table to another. The robot must approach the cup, pick the cup up, and approach the other table and place the cup down. Each of these motions has different constraints to consider, and are each a constrained planning problem with a different submanifold.

Problems with this structure are *multi-modal*: continuous motion must switch between different discrete *modes* of interaction, each mode imposing different constraints on the robot. Multi-modal planning considers planning not just a single submanifold, but on a union of multiple submanifolds. Single mode planning (i.e., constrained motion planning) is an essential component of a multi-modal planning algorithm. A motion planner within IMACS can be used for single mode planning.

Hauser and Latombe (2010) address the multi-modal planning problem for a finite number of modes, which was extended for infinite modes in (Hauser and Ng-Thow-Hing 2011). An asymptotically optimal multi-modal planning algorithm was proposed in (Vega-Brown and Roy 2016). Additionally, (Şucan and Kavraki 2011) implements multi-modal planning via acyclic task-motion multigraphs, and HPP implements multi-modal planning via constraint graphs (Mirabel et al. 2016). Multi-modal planning is also closely related to *task and motion planning*, which takes a more hierarchical approach to planning multi-modal paths (Srivastava et al. 2014; Dantam et al. 2018; Garrett et al. 2018).

Another essential component for multi-modal planning is the ability to sample transitions from one mode to the next. This is equivalent to sampling from the manifold defined by the concatenation of the two constraint functions, i.e., the intersection of their manifolds. In Section 5, we show that projection-based sampling covers a manifold defined by a constraint function, and thus can be used as a component in a probabilistically-complete manifold sampler. Thus, projection-based sampling can be used as a probabilistically-complete transition sampler in multi-modal planning. Note that IMACS only provides components of a multi-modal planner—while single-mode planning and transition sampling are important, they do not form a complete multi-modal planner.

3.5 Software Frameworks

There are some software packages available that can perform constrained motion planning, such as the CUIK Suite (Porta et al. 2014) and HPP (Mirabel et al. 2016). However, these are integrated frameworks for robotics that provide specific, specialized constrained motion planning algorithms, rather than the conceptual abstractions in IMACS which allow for mixing and matching of constraint solving techniques and planning algorithms. We provide an implementation of IMACS in the Open Motion Planning Library (OMPL 1.4) (Şucan et al. 2012), discussed further in Section 6.

4 Representing Implicit Manifolds

Despite their differences, most sampling-based planners have similar requirements from the robot's configuration space (LaValle 2006). The capabilities we are concerned with are the following:

- *Sampling*: Critical to sampling-based planning and exploration is the capability to sample over the entire space or near known configurations.
- *Local Planning*: Typically, sampling-based planners employ a local planner: an efficient, deterministic, and not necessarily complete method that is used to validate whether two states can be connected.

Sampling and local planning—rather than being defined as being planner specific operations—can be defined as operations on the space itself, and need not be specific to any planner. In general, prior works in geometrically constrained planning have augmented an existing sampling-based planner with some capability to "handle" constraints. More accurately, the augmentations required to craft a constrained motion planner are augmentations of the

capabilities as outlined above. The core insight of this work is that, as sampling-based planners plan within an abstract space, augmentation of the space is all that is needed to plan with constraints.

The contribution of this paper is a conceptual framework, IMACS (Implicit MAnifold Configuration Space), which is outlined within Section 4.1. IMACS enables a broad class of motion planners to plan with constraints, decoupling the choice of motion planner and method of constraint adherence. This section is organized as follows. First we discuss IMACS at an abstract level in Section 4.1 and describe how each of the space primitives are used by a sampling-based planner. The specific way a planner handles these critical components is referred to as the method of constraint adherence, as discussed previously in Section 3.3. Then, we show the two methods for constraint adherence that are implemented within IMACS. The first method we show is a projection-based method in Section 4.2, similar to CBIRRT2 (Berenson et al. 2011b). Next, we show a continuation-based method in Section 4.3, similar to Tangent Bundle RRT (Kim et al. 2016) and AtlasRRT (Jaillet and Porta 2013b).

4.1 Implicit Manifold Configuration Space

A sampling-based planning algorithm plans within a configuration space, and generates a collision-free path by using a configuration validity checker along with properties of the configuration space, as shown in Figure 2a. Prior works augmented the planning algorithm with a means of finding constraint adhering motions, as shown in Figure 2c. In contrast, IMACS is a layer of abstraction that lies between the representation of the robot's configuration space and the sampling-based planner, as shown in Figure 2d. IMACS can be thought of as a means to present the implicit manifold $\mathcal{M}$ defined by the constraint function F to a sampling-based planner, thus enabling the planner to plan with constraints.

As discussed above, there are only a few critical components that a sampling-based planner uses from its underlying space. We first discuss metrics and how they are affected within IMACS. Next, we discuss sampling and what properties a sampler should have within IMACS. Finally, we present local planning and interpolation on the constraint manifold within IMACS.

4.1.1 Metrics Normally, the distance metric utilized by a sampling-based planner is defined by the configuration space. The metric is primarily used for nearest-neighbor computations, by which states near a given state can be found. For example, a point robot in $\mathbb{R}^3$ and a manipulator arm with $\mathcal{Q} \subseteq \mathbb{R}^n$ commonly use the Euclidean norm. However, the notion of distance in the ambient space does not necessarily correspond to the intrinsic distance on the constraint manifold, as it may twist and curve relative to the ambient space (Tenenbaum et al. 2000). To remedy this problem, a natural metric to use would be the intrinsic Riemannian metric of the constraint manifold, or the arclength of the *geodesic* between points. A geodesic is the shortest curve between two points on the manifold, a generalization of a straight line in an Euclidean space. However, computing geodesics is infeasible at the scale needed for nearest-neighbor computations, and thus infeasible for planning.

Thus, prior works generally use the ambient metric from the configuration space, although there has been some work on approximating geodesic distance (Zha et al. 2018). The ambient metric is still "good enough" for most motion planning algorithms in practice, but it is possible that some theoretical guarantees no longer hold. Note that the metric from the configuration space is still a metric within the constraint manifold (when restricted to $\mathcal{M}$), as $\mathcal{M}$ is a subset of the configuration space. Additionally, the configuration space's metric is always an under-approximation of the intrinsic distance on $\mathcal{M}$, but is a reasonable approximation locally (as $\mathcal{M}$ is a manifold). Within IMACS, the metric from the configuration space is used. Importantly, use of the ambient metric allows asymptotic optimality guarantees to still hold for an asymptotically optimal planner within IMACS (shown in Section 5.4).

4.1.2 Sampling The ability to sample new configurations in the configuration space is critical to sampling-based planners. This is normally as simple as drawing uniformly random values from $\mathcal{Q}$. However, with an implicit manifold, the structure of the manifold is not known *a priori*, and is thus hard to sample uniformly without careful consideration or pre-processing. How this sampling is done is contingent on the specific constrained space, but we do not guarantee that it will produce uniform samples. Instead, we simply guarantee that any method of constraint adherence within IMACS will almost-surely sample any volume of non-zero measure within the manifold. This is similar to sampling with some bias.

Many sampling-based planners do not use samplers that sample from the entire configuration space (e.g., EST (Hsu et al. 1999), KPIECE (Şucan and Kavraki 2008)). Instead, they sample from a neighborhood around a known valid sample, leveraging the intuition that these configurations are likely to be valid as well. We refer to these as *neighborhood samplers*. These samplers within IMACS must be able to sample (potentially with bias) from a ball of radius r around a known configuration.

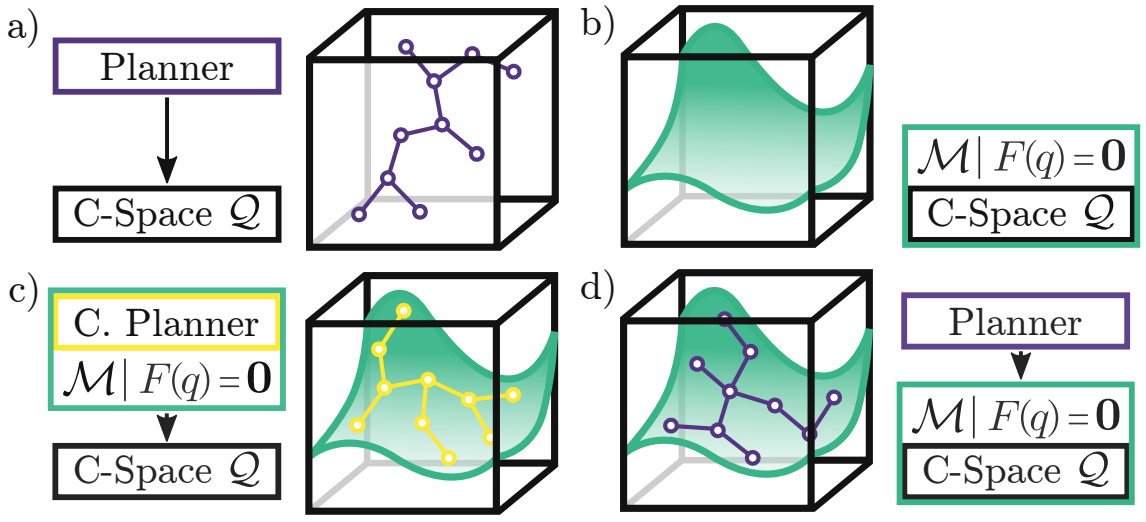


Figure 2. A depiction of the framework and its relation to sampling-based planners. **a)** A box configuration space $\mathcal{Q}$ is shown in black. A sampling-based planner (purple) plans in $\mathcal{Q}$ using primitives afforded by the space. **b)** A constraint function $F(q) = \mathbf{0}$ defines a implicit manifold $\mathcal{M}$ (green). **c)** An augmented constrained sampling-based planner (yellow) (e.g., CBIRRT2, etc.) plans on $\mathcal{M}$, using its constraint methodology. **d)** IMACS enables *any* sampling-based planner (purple) to plan on $\mathcal{M}$ by incorporating $\mathcal{Q}$ and the constraint function $F(q) = \mathbf{0}$.

Note that many planners (e.g., RRT) do not require samples to be valid, which in the context of constrained motion planning means the configuration is both collision-free and constraint adhering. Thus, the sampler for the ambient configuration space can be used to guide search if samples are not required to be valid.

Several sampling-based planners utilize a "projection" for estimating configuration space coverage in relation to a task (Sánchez and Latombe 2003; Ladd and Kavraki 2005; Şucan and Kavraki 2008), so that the planner can measure sample density and direct sampling towards less densely sampled regions (note "projection" in this case is not a projection operator as described before). "Projection" for coverage estimates are problem-specific heuristics to bias sampling and are left unaffected by the IMACS.

4.1.3 Local Planning Local planners are the means for sampling-based planners to quickly and efficiently attempt to find a path between two states. A local planner is not necessarily a complete planner, but typically is fast and deterministic. In geometric planning, the local planner generally takes the form of interpolation along a *geodesic* between two states. For example, within $\mathbb{R}^n$, interpolating along the geodesic from a configuration q_a to q_b has the analytic form of linear interpolation.

For implicit manifolds, traversing geodesics is more complicated. The generated geodesic must be a valid curve in $\mathcal{M}$. Generating a geodesic in the ambient configuration space and subsequently attempting to "fix" the geodesic to lie on the manifold ignores the manifold's curvature and can generate invalid motions. Thus, geodesic generation forms the basis for the local motion planner within IMACS, and is implemented by the method of constraint adherence. Figure 3 illustrates the way the methods within IMACS compute discrete geodesics. Note that, in general, these methods do not generate a "true" geodesic (the shortest curve between two points); these methods simply attempt to efficiently connect two points on the manifold, approximating the geodesic. Although this curve is not the true geodesic, it is referred to as the geodesic in this paper. Note that, as the manifold is locally Euclidean, as the distance between configurations lessens the curve generated by these methods approaches the true geodesic.

Within prior constrained sampling-based planning works, generating a geodesic takes the form of taking small enough steps to accurately traverse the manifold's curvature, creating a *discrete geodesic*.

Definition 5. Discrete Geodesic. *A discrete geodesic is a representation of a continuous geodesic σ within a metric space $\mathcal{M}$ by a sequence $\{x_j\}_{j=1}^n$ of $x_j \in \mathcal{M}$. An approximation of the arc length $L(\sigma) \geq L(\{x_j\}_{j=1}^n)$ is given by summing distances along the sequence: $L(\{x_j\}_{j=1}^n) = \sum_{j=2}^n \|x_{j-1} - x_j\|_2$. Continuity is enforced on the discrete geodesic by a fixed allowable distortion λ and step size s (dependent on*

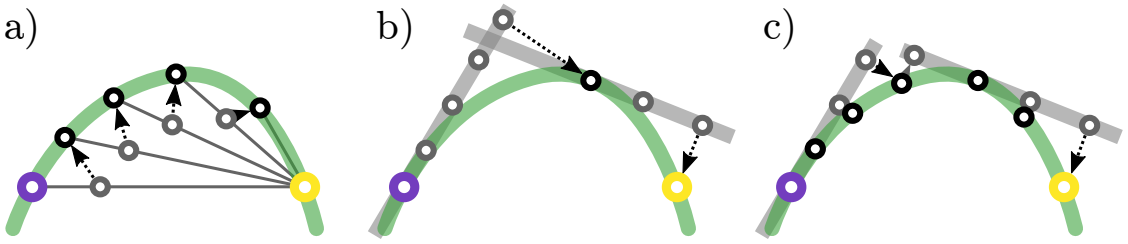


Figure 3. Projection-, tangent bundle-, and atlas-based geodesic interpolation. Between a start (purple) and goal (yellow) points on the implicit manifold (green), the discretized geodesic is computed (black). **a)** Projection-based (CBIRRT2). Small extensions are taken (grey) and projected using a projection operator (arrow). **b)** Tangent bundle-based (TB-RRT). The manifold is lazily evaluated with tangent spaces (grey), projecting when necessary. **c)** Atlas-based (AtlasRRT). Tangent spaces are traversed, projecting at every step.

the regularity of the constraint function), by which the following must hold: $\|x_{j-1} - x_j\|_2 \leq \lambda s, \forall j \in 2 \dots n$

Computing the exact geodesic between two points is hard and computationally costly (Hauser 2013; Mirabel and Lamiraux 2016). Typically, generating the discrete geodesic takes the form of integrating an ODE on the surface of the constraint manifold. Given two points $q_a, q_b \in M$, we can define the ODE:

$$\dot{q} = q_b - q(t) \qquad q(0) = q_a \tag{1}$$

If there is divergence or a lack of progress, then the method terminates unsuccessfully, akin to colliding with an obstacle. However, we show in Section 5.2 that the scheme of ODE integration on manifolds satisfies the criteria of a local planner, and maintains the probabilistic completeness of algorithms running on IMACS.

Many sampling-based planners need to compute states interpolated between two states. For example, RRT "steers" toward a sampled state, which in the geometric case means interpolating towards a sampled state up to some fixed distance. However, without the geodesic between two states, it is not known *a priori* what states lie between two states and where; the geodesic must be computed to interpolate between states. Once a discrete geodesic is computed, an interpolated state can be computed by doing piecewise interpolation, as shown in Alg. 1. Alg. 1 first sums distances along the generated geodesic (from an INTEGRATE method implemented by a method for constraint adherence, line 2) to approximate geodesic distance from the start q_a to the goal q_b. Using these distances, an approximation of the desired state at a time $t \in [0, 1]$ (where $t = 0$ gives q_a, $t = 1$ gives q_b) along the geodesic can be found by interpolating between intermediate states. The FIX routine within Alg. 1 (line 12) takes the interpolated point finds a close adhering

Algorithm 1 Geodesic Interpolate

Input $q_a \in \mathcal{M}$ the starting configuration, $q_b \in \mathcal{Q}$ the goal configuration, and $t \in [0, 1]$ the interpolation parameter.

Output A $q \in \mathcal{M}$ that is at t on the geodesic path from q_a to q_b. Returns q_a upon failure.

1: **procedure** INTERPOLATE(q_a, q_b, t)
// Call a method-specific integrator
2: $\{q_k\}_{k=1}^{j} \leftarrow$ INTEGRATE(q_a, q_b)
// Check if integrator reached q_b
3: **if** $\|q_j - q_b\|_2 > \varepsilon$ **then**
4: return q_a
// Compute arclength
5: $d_0 = 0$
6: **for** $i = 1, \dots j$ **do**
7: $d_i = d_{i-1} + \|q_i - q_{i-1}\|_2$
// Find intepolated state
8: $q \leftarrow q_a$
9: **for** $i = 0, \dots j$ **do**
10: **if** $d_i/d_j \leq t$ and $d_{i+1}/d_j > t$ **then**
11: $t' = (td_j - d_i)/(d_{i+1} + d_i)$
// Use method to adhere to F
12: $q \leftarrow$ FIX($q_i + t'(q_{i+1} - q_i)$)
13: **if** $q \neq$ false **then**
14: **break**
15: **return** q

configuration via the method for constraint adherence (e.g., the projection operator for the projection-based method, or ψ_C for the continuation-based method). For efficiency, collision checking can be done within INTEGRATE, terminating early if a collision is reached.

4.1.4 Methods for Constraint Adherence In summary, the key idea of IMACS is to represent the implicit constraint manifold with primitives that are necessary for sampling-based planning. IMACS consists of a

space representation that has augmented sampling and geodesic computation for a given underlying configuration space and constraint. This allows a broad class of sampling-based planners to plan with constraints without any special consideration.

The next two sections describe the approaches to sampling and geodesic computation for the projection-based and continuation-based methods. The projection-based method is similar in concept to CBIRRT2 (Berenson et al. 2011b), but using our constraint function. The continuation-based method is similar to AtlasRRT (Jaillet and Porta 2013b) and Tangent Bundle RRT (Kim et al. 2016), and emulates both using shared infrastructure.

4.2 Projection-Based Method

In a constrained motion planning problem, a an adhering path only contains configurations that adhere to the constraint function, $F(q) = \mathbf{0}$. One method to find adhering configurations is with a *projection operator*. A projection operator takes a configuration and maps it onto the set of adhering configurations, projecting it onto the constraint manifold. Both sampling and local planning for the projection-based method rely heavily on the projection operator.

4.2.1 Definitions Formally, we define the projection operator as follows:

Definition 6. Projection Operator. *A* projection operator *is a surjective function* $P : \mathbb{R}^n \rightarrow \mathcal{M}$ *that takes a given point* $q_0 \in \mathbb{R}^n$ *and* projects q_0 *onto* $\mathcal{M}$*, where* $P(q_0) = q_0$ *if* $q_0 \in \mathcal{M}$*.*

Given the implicit description of $\mathcal{M}$, the projection operator is often formulated as a constrained minimization problem:

$$\begin{aligned} \min_{q} \quad & \frac{1}{2}\|q_0 - q\|_2{}^2 \\ \text{subject to} \quad & F(q) = \mathbf{0} \end{aligned}$$

The formulation of the projection operator as a constrained minimization problem lends itself to a formulation as a Lagrangian with Lagrange multipliers $\lambda \in \mathbb{R}^k$:

$$\Lambda(q, \lambda) = \frac{1}{2}\|q_0 - q\|^2 - F(q)^T\lambda$$

We can solve for a solution to this system using gradient descent, with the update $\Delta q_{i+1} = -J(q_i)^+ F(q_i)$, where $J(q)^+$ is the pseudoinverse of the Jacobian (Nocedal and Wright 2006). The full algorithm for the projection operator via gradient descent (P) is shown in Alg. 2. This formulation to is similar to numerical inverse kinematic techniques (Buss 2004), which also can be thought of as projection operators (e.g., task space regions (Berenson et al. 2011b))

Algorithm 2 P, a Projection Operator

Input q, an initial configuration to project from
Output $P(q) \in \mathcal{M}$, the projected state configuration. On failure, "false" is returned.

```
1: procedure P(q)
2:     x ← F(q)
3:     while ‖x‖_2 > ε and iterations remain do
4:         q ← q − J(q)^+ x
5:         x ← F(q)
6:     if ‖x‖_2 ≤ ε then
7:         return q
8:     else
9:         return false
```

We show in Section 5.1 that the projection operator covers the constraint manifold. That is, for any configuration on the manifold, there exist configurations within the ambient space that will be projected onto the configuration.

4.2.2 Sampling We assume that the ambient configuration space $\mathcal{Q}$ has some sampler with non-zero density distribution over $\mathcal{Q}$, i.e., there is a chance to draw a sample from any open set in $\mathcal{Q}$. It is simple to define the projection-based sampler: repeatedly sample a point using the sampler from the ambient configuration space and project the point. Note that this routine may fail due to the projection operator failing to converge.

Similarly, for sampling nearby existing configurations, the neighborhood sampler from the ambient space is used to sample a new point which is projected to the manifold. Because the constraint manifold uses the ambient metric, all points that are within some distance r to a point in the ambient space are also within r on the manifold.

4.2.3 Local Planning As stated in Section 4.1.3, generating geodesics on the manifold is essential for local planning. For the projection-based local planning, we use a method inspired by integrating an ODE on a manifold (Hairer et al. 2006). In Alg. 3, we integrate the ODE given in Eq. 1 using projection to correct for drift off the manifold. Alg. 3 uses a fixed maximum step size s with respect to the Euclidean norm, and a tolerance ε. This method is very similar to the constrained extension steps employed in other projection-based sampling-based planning methods in the literature, such as CBIRRT2.

Generally, it is not enough to have an arbitrary discrete curve. Continuity must be enforced to some

Algorithm 3 Projection Integrator

Input $q_a \in \mathcal{M}$, an initial configuration, and $q_b \in \mathcal{Q}$, a goal configuration.

Output $\{q_k\}_{k=1}^{j}$ such that $\forall k, q \in \mathcal{M}$. Upon success, $\|q_b - q_j\|_2 \leq s$.

procedure INTEGRATE(q_a, q_b)
  $q_1 \leftarrow q_a, j \leftarrow 1$
  while $\|q_j - q_b\|_2 > s$ **do**
    // Interpolate in $\mathcal{C}$-space
    $q_{j+1} \leftarrow P(q_j + (s/\|q_b - q_j\|_2)(q_b - q_j))$
    $d \leftarrow \|q_{j+1} - q_j\|_2$
    // Check distortion & progress
    if $d > \lambda s$ or $d < s$ **then**
      break
    $j \leftarrow j + 1$
  return $\{q_k\}_{k=1}^{j}$

degree. Alg. 3 is modified slightly from a pure integration approach to respect conditions of continuity in regards to a maximum distortion parameter λ and to terminate early if no forward progress is made (a saddle point, e.g., the antipode of a point on a sphere or a sufficient curve in the manifold). For efficiency, collision checking can also be done within the integration step, but is left out of Alg. 3 for clarity.

4.3 Continuation-Based Method

As discussed in Section 3.3.2, the constraint manifold $\mathcal{M}$ can be approximated by a set of *tangent spaces*. A few recent planners such as AtlasRRT (Jaillet and Porta 2013b) and TB-RRT (Kim et al. 2016) use tangent space approximations for efficient sampling near the manifold. Both of these methods are emulated within IMACS.

Common between these methods is the construction and maintenance of an *atlas*. Definitions of the underlying mathematical concepts and implementation of operations associated with the atlas are presented in Section 4.3.1. The definitions and routines presented here are taken and reformulated from (Rheinboldt 1996; Henderson 2002; Jaillet and Porta 2013b; Kim et al. 2016). The primary difference is their use within IMACS as subroutines in a representation of a space, rather than as subroutines within a specific planner.

4.3.1 Definitions Both AtlasRRT and TB-RRT are higher-dimensional continuation methods: they construct a piecewise-linear approximation of the constraint manifold. This approximation is referred to as an *atlas*, in reference to the mathematical objective described below.

Definition 7. Atlas. *A family of homeomorphisms whose domains cover a manifold $\mathcal{M}$ is called an* atlas *($\mathcal{A}_\mathcal{M}$) for $\mathcal{M}$. A particular homeomorphism and domain $(\omega, U) \in \mathcal{A}_\mathcal{M}$ is called a* chart *for the atlas, where $\omega : U \rightarrow V \subseteq \mathbb{R}^n$ is a homeomorphism.*

In the case of AtlasRRT and TB-RRT, the atlas is formed of charts that are represented by *tangent spaces*, which are $\mathbb{R}^{n-k}$-hyperplanes embedded in the ambient configuration space.

Definition 8. Tangent Space. *The* tangent space *of $\mathcal{M}$ at x (written $T_x\mathcal{M}$) is the set of all tangent vectors at x. For a n-dimensional manifold, the tangent space is an n-dimensional real vector space.*

Details of how tangent spaces are computed are provided below. All together, the collection of tangent spaces at each point on the manifold form the *tangent bundle*, a *vector bundle* of tangent spaces.

Definition 9. Vector Bundle. *A vector bundle on a manifold $\mathcal{M}$ is given by $\xi = (\pi, E, \mathcal{M})$. E is the* total space*, a disjoint union of a collection of vector spaces, $E = \bigsqcup_{x \in \mathcal{M}} E_x$, where E_x is the vector space at $x \in \mathcal{M}$ and $\bigsqcup$ denotes disjoint union. $\pi : E \rightarrow \mathcal{M}$ is a continuous surjective function called the* bundle projection*, mapping $\pi(E_x) = x$.*

Definition 10. Tangent Bundle. *The* tangent bundle *of an n-dimensional manifold $\mathcal{M}$ (written $T\mathcal{M}$) is a vector bundle composed of the disjoint union of the tangent spaces $\mathbb{R}^n$ for each $x \in \mathcal{M}$, $T\mathcal{M} = (\pi, \bigsqcup_{x \in \mathcal{M}} T_x\mathcal{M}, \mathcal{M})$*

The following sections describe the various operations that utilize tangent spaces and charts.

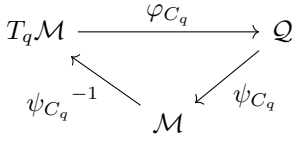
Creating New Charts: An atlas $\mathcal{A}_\mathcal{M}$ for the constraint manifold $\mathcal{M}$ contains charts $(\omega, U) \in \mathcal{A}_\mathcal{M}$ which locally parameterize $\mathcal{M}$. We would like to explicitly define the chart that locally parameterizes the manifold for a given point.

A *chart* is defined by $C_q = (q, \Phi(q))$ for a point q on a constraint manifold $\mathcal{M}$, and consists of the point q and an orthonormal basis $\Phi(q)$ for the tangent space $T_q\mathcal{M}$, $\Phi(q) \in \mathbb{R}^{(n-k)\times n}$ From (Rheinboldt 1996), the orthonormal basis can be constructed by finding a solution to the equation:

$$\begin{bmatrix} J(q) \\ \Phi(q)^T \end{bmatrix} \Phi(q) = \begin{bmatrix} \mathbf{0} \\ I \end{bmatrix} \tag{2}$$

Where $\mathbf{0} \in \mathbb{R}^{k\times(n-k)}$, $I \in \mathbb{R}^{(n-k)\times(n-k)}$. The basis for the tangent space can be computed by finding an orthonormal basis for the nullspace of the Jacobian $\Phi(q) = \ker(J(q))$. Typically, this is solved using some matrix decomposition (e.g., QR (Press et al. 2007)). The computation of new charts is done in Alg. 5.

To and From Charts: Once charts have been created on the manifold, they can be used to generate configurations that adhere to the constraint. There are three functions to consider for a chart C_q: φ_{C_q}, a basis change that embeds a point from chart space into the configuration space (Eq. 3), ψ_{C_q}, a mapping from the space around the chart to the manifold (Alg. 4), and ${\psi_{C_q}}^{-1}$, which maps from the manifold back into chart space (Eq. 5). The cyclical relation of these functions is illustrated in Figure 4a and the following diagram:



These functions are explained in the following paragraphs.

For a given chart around a configuration $q \in \mathcal{M}$, points within a chart C_q, $u_q \in T_q\mathcal{M}$, are embedded into the ambient configuration space with $\varphi_{C_q} : T_qM \rightarrow \mathbb{R}^n$:

$$q_u = \varphi_{C_q}(u_q) = q + \Phi(q)u_q \tag{3}$$

Although q_u will be close to the manifold, the configuration must still be projected onto $\mathcal{M}$. The exponential map $\psi_{C_q} : \mathbb{R}^n \rightarrow \mathcal{M}$ projects points embedded in $\mathbb{R}^n$ from the chart onto $\mathcal{M}$. A key difference from the projection operator presented in Section 4.2.1 is that ψ_{C_q} projects *orthogonally* relative to the embedding of T_qM in $\mathbb{R}^n$, and is applicable only to points within the chart C_q's space. The projected point q_m is found by solving the system of equations:

$$F(x) = 0 \qquad \Phi(q)^T(q_m - q) = 0$$

This system of equations is solved with gradient descent, shown in Alg. 4, similar to Alg. 2. In Alg. 4, the following equations are used for clarity, given the initial point q_u and chart C_q:

$$A_u(q_m) = \begin{bmatrix} J(q_m) \\ \Phi(q)^T \end{bmatrix}, \; b_u(q_m) = \begin{bmatrix} F(q_m) \\ \Phi(q)^T(q_m - q_u) \end{bmatrix} \tag{4}$$

A_u is invertible and full rank, as the Jacobian is assumed to be of full rank on $\mathcal{M}$, and thus is also of full rank near $\mathcal{M}$ (by the inverse function theorem). Note that Alg. 4 can fail due to lack of convergence similarly to the projection operator, but this is less likely due to the proximity of the charts to the manifold.

The inverse mapping, or logarithmic map, ${\psi_{C_q}}^{-1} : \mathcal{M} \rightarrow T_q\mathcal{M}$ is defined as a projection onto the tangent space:

$${\psi_{C_q}}^{-1}(q_m) = \Phi(q)^T(q_m - q) \tag{5}$$

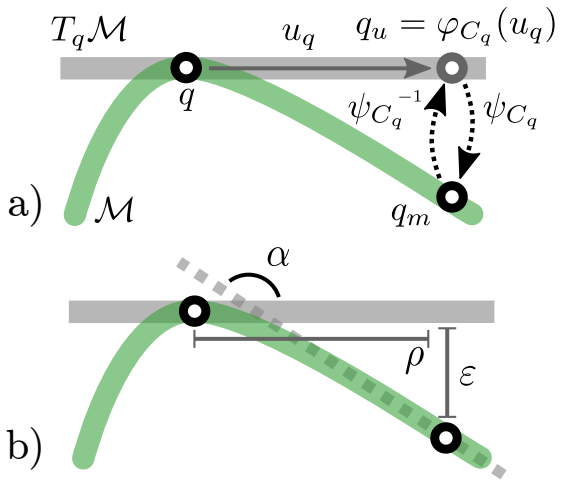


Figure 4. a) Illustration of the various chart operators as described in Section 4.3.1. **b)** Illustration of the parameters used in a chart's validity region.

Algorithm 4 ψ_{C_q}, the exponential map

Input q_u, an initial configuration from C_q's tangent space

Output $q_m = \psi_{C_q}(q_u)$, the orthogonally projected state configuration. On failure, "false" is returned.

// $A_u(q_m), b_u(q_m)$ defined in Eq. 4
procedure $\psi_q(q_u)$
  $q_m \leftarrow q_u$
  $b \leftarrow b_u(q_m)$
  // Orthogonally project until convergence
  while $\|b\|_2 > \epsilon$ **and** iterations remain **do**
    $q_m \leftarrow q_m - A_u(q_m)^{-1}b$
    $b \leftarrow b_u(q_m)$
  if $\|b\|_2 \leq \epsilon$ **then**
    return q_m
  else
    return false

Chart Validity: Due to the constraint manifold's non-Euclidean structure, the tangent spaces of the charts that form the atlas are only good local approximations. To maintain accuracy of the approximation, each chart has a *validity region*, wherein the chart is applicable to the manifold. A chart C_q's validity region $\mathcal{V}_q \subset T_q\mathcal{M}$ is defined by parameters:

- ε: maximum distance of $T_q\mathcal{M}$ to $\mathcal{M}$
- ρ: radius of ball in $T_q\mathcal{M}$
- α: maximum curvature of $T_q\mathcal{M}$ to $\mathcal{M}$

The following inequalities define the validity region, for a point $u_q \in T_q\mathcal{M}$, $q_u = \varphi_{C_q}(u_q)$, $q_m = \psi_{C_q}(q_u)$:

$$\|q_u - q_m\|_2 \leq \varepsilon \quad \frac{\|u_q\|_2}{\|q - q_m\|_2} < \cos(\alpha) \quad \|u_q\|_2 \leq \rho$$

Respectively, these correspond to error, curvature, and maximum span. These parameters are shown in Figure 4b. Note that it is possible to adaptively change these parameters, but this is not discussed in this work.

If a point $q_m \in \mathcal{M}$ is such that $u_q = {\psi_{C_q}}^{-1}(q_m) \notin \mathcal{V}_q$ for any chart $C_q \in \mathcal{A}_\mathcal{M}$, then no existing chart is suitable and a new chart needs to be created. Given this new chart C_x, other charts near C_q can add a *separating halfspace* to their *applicability polyhedron* P_q. Given a new $u \in T_qM$ and a neighboring chart $C_x = (x, \Phi(x))$, with $u_x = \psi_q^{-1}(x)$, u is in the applicability region if it satisfies the inequality:

$$2u^T u_x \leq {\|u_x\|_2}^2$$

It is not strictly necessary to separate charts, as will be seen with the tangent bundle planning approach. Each of the charts C_x and C_q store the halfspace. We consider $u \in P_q$ for a chart C_q when a point u satisfies all inequalities, and thus is within the applicability polyhedron.

In our implementation, the atlas is stored as a collection of charts within a nearest neighbor structure, using the configuration space's metric between chart centers. The nearest neighbor structure is used in Alg. 5 in the method WithinRadius$(q, \mathcal{A}_\mathcal{M}, r)$, where $q \in \mathbb{R}^n$, $\mathcal{A}_\mathcal{M}$ is the atlas (a collection of charts) and $r \in \mathbb{R}^+$ is a radius. WithinRadius returns all charts within the radius of the configuration. In both cases the maximum radius of a chart, ρ, is used to obtain nearby charts.

Primarily, all operations that modify or access the atlas are done through GetChart (Alg. 5), which gets the chart that a point on the constraint manifold belongs to. Auxiliary to this method is FindChart, which searches existing charts for a chart that is feasible given a point. FindChart returns the chart with the least error to the point if multiple are feasible. If FindChart fails, a new chart is created and added to the atlas in GetChart. For efficiency in implementation, a configuration can store a reference to the last chart it was found to belong to, rather than needing a full lookup each time.

4.3.2 Sampling In imacs, new constraint-adhering configurations are generated by sampling within tangent spaces and projecting these points onto the manifold. A method for sampling the entire manifold and sampling near known configurations is given in Alg. 6. The method SampleAtlas is implemented to sample a chart from the collection of charts. Generally, this is uniform sampling of the collection of charts. As a heuristic improvement, bias can be introduced to SampleAtlas in order to do weighted sampling (ensuring that each chart always has a positive probability of being sampled). This is done in tb-rrt, where charts are biased by their distance to the goal (Kim et al. 2016). SampleBall(C, q, r) samples a point within radius r of q within the chart C's space. Not shown in Alg. 6 are failure conditions due to ψ_C not converging; in this case, the center of an existing chart is returned.

Algorithm 5 Find and Create Chart for q_a in $\mathcal{A}_\mathcal{M}$

Input q_a, the configuration to find a chart for, and $\mathcal{A}_\mathcal{M}$, the current atlas of charts

Output C, the best (distance-wise) chart for q_a. On failure (no valid charts for q_a), an empty object {} is returned.

```
procedure FindChart(q_a, A_M)
    best ← ε, C ← {}
    // Get all charts within ρ of q_a
    charts ← WithinRadius(q_a, A_M, ρ)
    for C_q ∈ charts do
        u_q ← ψ_{C_q}^{-1}(q_a)
        d ← ‖φ_{C_q}(u_q) − q_a‖_2
        // Check if C_q is a closer, valid chart
        if d < best and u_q ∈ P_q then
            best ← d, C ← C_q
    return C
```

Input q_a, the configuration to get a chart for, and $\mathcal{A}_\mathcal{M}$, the current atlas of charts

Output C, the best (distance-wise) chart for q_a. If there are no valid charts for q_a, a new chart is created using q_a as the origin.

```
procedure GetChart(q_a, A_M)
    // Check if chart already exists
    C_a ← FindChart(q_a, A_M)
    if C_a ≠ {} then
        return C_a
    // Create new chart (see Eq. 2)
    C_a ← (q_a, Φ(q_a))
    // Generate separating halfspaces (optional)
    if separate charts then
        adjacent ← WithinRadius(q_a, A_M, 2ρ)
        for C_q ∈ adjacent do
            ComputeHalfspaces(C_a, C_q)
    A_M ← A_M ∪ {C}
    return C
```

Algorithm 6 Continuation-based Sampling

Input $\mathcal{A}_\mathcal{M}$, the current atlas of charts.
Output $q \in \mathcal{M}$, a random configuration.

1: **procedure** SAMPLE($\mathcal{A}_\mathcal{M}$)
2: **do**
// Uniformly choose a chart
3: $C \leftarrow$ SAMPLEATLAS($\mathcal{A}_\mathcal{M}$)
4: $u \leftarrow$ SAMPLEBALL($C, \mathbf{0}, \rho\beta$)
5: **while** $u \notin P_C$ **or** $\neg\psi_C(u)$
6: **return** $\psi_C(u)$

Input q, the point to sample nearby, radius $r \in \mathbb{R}^+$, and $\mathcal{A}_\mathcal{M}$, the current atlas of charts.
Output $q \in \mathcal{M}$, a random configuration.

1: **procedure** NEIGHBORHOODSAMPLE(q, r $\mathcal{A}_\mathcal{M}$)
2: **do**
3: $C \leftarrow$ GETCHART($q, \mathcal{A}_\mathcal{M}$)
4: $u \leftarrow$ SAMPLEBALL($C, {\psi_C}^{-1}(q), r\beta$)
5: **while** $\neg\psi_C(u)$
6: **return** $\psi_C(u)$

Although at first biased towards explored areas, in the limit—once the manifold has been fully explored—sampling approaches uniform sampling (Jaillet and Porta 2013b). These methods can sample within hard-to-project areas, such as the interior surface of a highly curved manifold. This is empirically demonstrated in Section 6.2.

Neighborhood sampling is shown in NEIGHBORHOODSAMPLE in Alg. 6. For a given configuration q, a point is sampled within radius r of q's projection onto the chart. As the chart is a local approximation of the manifold, a ball with radius r in chart space mapped via ψ_C to $\mathcal{M}$ contains the ball of radius r in $\mathcal{M}$. However, due to the curvature of the manifold, parts of the ball may be invalid, resulting in a failure condition that simply returns the original configuration.

The validity parameter ρ is used within SAMPLE to generate samples within the current validity region of the sampled chart. Within the sampling process, note the β parameter on line 4 in both sampling routines. $\beta \geq 1$ is an exploration parameter that allows search to extend from current applicable charts, so that there is a chance of generating new configurations outside of the current collection of chart's validity regions. Higher values of β have potential to generate more invalid samples, but will also cause the manifold to be more rapidly explored.

4.3.3 Local Planning Generating geodesics using continuation-based methods is also inspired by

Algorithm 7 Continuation Integrator

Input $q_a \in \mathcal{M}$, an initial configuration, $q_b \in \mathcal{Q}$, a goal configuration, and $\mathcal{A}_\mathcal{M}$, the current atlas of charts.
Output $\{q_k\}_{k=1}^{j}$ such that $\forall k, q \in \mathcal{M}$. Upon success, $\|q_b - q_j\|_2 \leq s$.

1: **procedure** INTEGRATE(q_a, q_b, $\mathcal{A}_\mathcal{M}$)
2: $q_1 \leftarrow q_a$, $j \leftarrow 1$
3: $C_j \leftarrow$ GETCHART($q_j, \mathcal{A}_\mathcal{M}$)
4: $u_j \leftarrow \psi_{C_j}^{-1}(q_j)$ *// q_j projected into C_j*
5: $u_b \leftarrow \psi_{C_j}^{-1}(q_b)$ *// goal q_b projected into C_j*
6: **while** $\|u_j - u_b\|_2 > s$ **do**
// Interpolate in chart space
7: $u_{j+1} \leftarrow u_j + s(u_b - u_j)/\|u_b - u_j\|_2$
8: $q_{j+1} \leftarrow \psi_{C_j}(u_{j+1})$
9: $d \leftarrow \|q_{j+1} - q_j\|_2$
// Check distortion & progress
10: **if** $d > \lambda s$ or $d < s$ **then**
11: **break**
12: $j \leftarrow j + 1$
// Check if u_j is in validity region
13: **if** $\|\varphi_{C_j}(u_j) - q_j\|_2 > \varepsilon$
14: **or** $s/d < \cos\alpha$
15: **or** $\|u_j\|_2 > \rho$
16: **or** $u_j \notin P_q$ **then**
// Create or get new valid chart
17: $C_j \leftarrow$ GETCHART($q_j, \mathcal{A}_\mathcal{M}$)
18: $u_j \leftarrow \psi_{C_j}^{-1}(q_j)$
19: $u_b \leftarrow \psi_{C_j}^{-1}(q_b)$
20: **return** $\{q_k\}_{k=1}^{j}$

1: **procedure** LAZYINTEGRATE(q_a, q_b, $\mathcal{A}_\mathcal{M}$)
2: $q_1 \leftarrow q_a$, $j \leftarrow 1$
3: $C_j \leftarrow$ GETCHART($q_j, \mathcal{A}_\mathcal{M}$)
4: $u_j \leftarrow \psi_{C_j}^{-1}(q_j)$ *// q_j projected into C_j*
5: $u_b \leftarrow \psi_{C_j}^{-1}(q_b)$ *// goal q_b projected into C_j*
6: **while** $\|u_j - u_b\|_2 > s$ **do**
// Interpolate in chart space
7: $u_{j+1} \leftarrow u_j + s(u_b - u_j)/\|u_b - u_j\|_2$
8: $q_{j+1} \leftarrow \varphi_{C_j}(u_{j+1})$
9: $j \leftarrow j + 1$
// Check if u_j is close to manifold
10: **if** $\|\varphi_{C_j}(u_j) - q_j\|_2 > \varepsilon$
11: **or** $u_j \notin P_q$ **then**
// Create or get new valid chart
12: $q_j \leftarrow \psi_{C_j}(u_j)$ *// project to $\mathcal{M}$*
13: $C_j \leftarrow$ GETCHART($q_j, \mathcal{A}_\mathcal{M}$)
14: $u_j \leftarrow \psi_{C_j}^{-1}(q_j)$
15: $u_b \leftarrow \psi_{C_j}^{-1}(q_b)$
16: **return** $\{q_k\}_{k=1}^{j}$

integrating an ODE on a manifold. In the continuation-based method, geodesic generation is accomplished by walking along the tangent spaces of the approximation, switching tangent spaces once certain criteria are met. In Alg. 7, two integration methods are shown: INTEGRATE which checks for constraint adherence at every step, and LAZYINTEGRATE which takes a lazy approach to constraint validation.

INTEGRATE is very similar to the extension method presented by AtlasRRT (Jaillet and Porta 2013b); INTEGRATE projects at every step along the approximation, and generates separating half-spaces to create polytopes of the tangent space for more accurate geodesic traversal, at the cost of additional computation (shown in Figure 3c).

LAZYINTEGRATE is very similar to the extension method presented by TB-RRT; LAZYINTEGRATE takes a lazy approach to geodesic generation, projecting to the manifold only when necessary to switch tangent spaces (shown in Figure 3b). This has the benefit of performing less work computing projections, but it is harder to do correctly. Extra consideration is needed when performing collision checking as lazy evaluation generates a relaxed geodesic, which might hit or miss obstacles.

4.4 Sampling-Based Planning

Above, we have presented the details of how projection- and continuation-based spaces are emulated within IMACS. These spaces are primarily formed by their methods of sampling and geodesic generation. Note that these spaces are not the only methods for constraint adherence that can be captured with IMACS: for example, tangent space-based planning would fit within this framework as well (as described in the first part of Section 3.3.2).

Within the context of sampling-based planning, IMACS is used as the *space* that a planner is exploring, it is an overlay for an underlying ambient configuration space with a constraint function. Each method for constraint adherence's sampler is used in place of the sampler normally offered by the ambient configuration space. Note that for some planners (e.g., RRT) the ambient configuration space sampler can be used, as these planners do not directly use samples in their graph construction. Geodesic generation is used in local planning and interpolation: whenever a motion is validated between states or a new state is generated by interpolation, INTERPOLATE (Alg. 1) is used with the method's integrator (e.g., Alg. 3 for the projection-based space).

We show in Section 5 some of the theoretical guarantees afforded by IMACS. In particular, if a sampling-based planner is probabilistically complete, it remains probabilistically complete while within IMACS. Additionally, asymptotically optimal planners such as RRT* and PRM* retain their asymptotic optimality within IMACS.

5 Theoretical Guarantees

Many sampling-based planners provide some sort of formal guarantee about their completeness or optimality of solution. Typically, sampling-based planners provide probabilistic completeness guarantees (that is, the probability that a planner will find a solution if one exists goes to 1 as the number of iterations goes to infinity), and many recent planners provide asymptotic optimality guarantees (that is, the solution found will converge to the global optimum as the number of iterations goes to infinity). In this section, we provide building blocks that would compose a larger proof of probabilistic completeness or asymptotic optimality for planners that run within IMACS. These proofs leverage the modular nature of sampling-based planners and provide guarantees about the certain individual components, namely the samplers and local planners.

We first present proofs that show a probabilistically complete planner within IMACS remains probabilistically complete. Samplers and local planners form the core of probabilistic completeness for any motion planner. This is similar to how samplers and local planners form the core of a constrained sampling-based planning method, as well as the core of the methods of constraint adherence within IMACS. The analysis presented follows from the measure theoretic analysis presented in (Ladd and Kavraki 2004), which, among other properties of the planner itself, requires two properties from the space representation:

- The sampler used must have a non-zero chance of sampling any configuration in the configuration space. For neighborhood samplers, this corresponds to having a non-zero chance of sampling any configuration within the specified radius.
- The local planner must be a measurable relation. Additionally, the local planner must be such that its transitive closure is a measurable relation (i.e, the path reachability condition). That is, for any two configurations q_a, q_b that are path connected, there exists some sequence of configurations $q_a, q_1, \ldots, q_m, q_b$ such that the local planner reports validity between each successive configuration.

We will show that the samplers and local planners from the projection- and continuation-based methodologies

satisfy the above properties, and thus sampling-based planners maintain probabilistic completeness within the framework.

Additionally, we provide proofs that asymptotically optimal motion planners maintain their asymptotic optimality guarantees within the framework. We show that under the assumptions above about the sampler and local planner, the proofs of asymptotic optimality of RRT* and PRM* given in (Karaman and Frazzoli 2011) hold, and that connection radii γ exist to maintain asymptotic optimality.

The key intuition of why probabilistic completeness and asymptotic optimality are preserved within IMACS is that the structure of the implicit space is that of a smooth manifold. Manifolds are such that they locally appear to be like a Euclidean space, and thus in a sense are "nice" for planning. It is possible to provide guarantees about sampling and local planning under constraints as, locally, there is always space around the manifold that is well behaved.

The rest of this section is organized as follows. In Section 5.1, we show that projection-based sampling (as defined in Section 4.2.2) which uses a projection operator (as defined in Section 4.2.1) satisfies the criteria of a sampler. Next, Section 5.2 shows that the geodesic generators of the projection- and continuation-based spaces are indeed local planners as defined above. Together, these facts lead to the probabilistic completeness of a planner within IMACS, discussed in Section 5.3. Finally, we discuss how the proofs of asymptotic optimality remain unaffected by IMACS in Section 5.4.

5.1 Manifold Coverage and Sampling

Key to the projection-based methods is the projection operator. Critically, the projection operator must guarantee that it will project a state onto the constraint manifold. We show that the projection operator, as presented, *covers* the manifold. That is, we show in Prop. 1 that any open set on the manifold $U \subseteq \mathcal{M}$ has an associated open set in the ambient configuration space $V \subseteq \mathbb{R}^n$ such that $P(V) = U$. With this fact, the projection-based sampler is shown to be a sampler, as all measurable sets on the manifold will be sampled.

This proof is inspired by the proof of manifold coverage used in CBIRRT2 (Berenson et al. 2011b) and a proof of continuity of the projection operator given in (Mirabel and Lamiraux 2016). However, note that the proof offered in Prop. 1 is more general than the proof of manifold coverage given in (Berenson et al. 2011b). The difference lies in the class of constraints considered: (Berenson et al. 2011b) gives a proof of manifold coverage for a class of end-effector constraints for manipulators (task space regions), while our proof is for any constraint of the form $F(q) = \mathbf{0}$ for any robot (as defined in Section 2.2), generalizing the prior result.

Recall that we assume the configuration space $\mathbb{R}^n$ is a measure space with the Borel σ-algebra. We assume that there is a probability measure over $\mathbb{R}^n$ with non-zero measure for any non-empty open set. We assume the sampler for the ambient configuration space has a distribution with non-zero density on all of $\mathcal{Q}$.

We show that the projection operator covers the manifold by showing that there exists a region (potentially small) around the manifold such that all points in this region project onto the manifold. A natural concept for the immediate region closest to the smooth constraint manifold is that of a *tubular neighborhood*.

Definition 11. Tubular Neighborhood (Spivak 1999; Lee 2003). *Let $\mathcal{M} \subset \mathbb{R}^n$ be a at least a C^1-smooth $(n-k)$-dimensional compact manifold. A* tubular neighborhood *of $\mathcal{M}$ is a pair (f, ξ) where:*

- *$\xi = (\pi, E, \mathcal{M})$ is a vector bundle (Def. 9) over $\mathcal{M}$. E is the disjoint union of a collection of vector spaces, $E = \bigsqcup_{x \in \mathcal{M}} E_x$, where E_x is k-dimensional. $\pi : E \rightarrow \mathcal{M}$ is the bundle projection, a continuous surjective mapping where $\pi(E_x) = x$.*
- *$f : E \rightarrow \mathbb{R}^n$ is a smooth embedding such that $f(E)$ is an open neighborhood of $\mathcal{M}$ in $\mathbb{R}^n$.*

A partial tubular neighborhood *of $\mathcal{M}$ is a triple (f, ξ, E_U) where $E_U \subset E$ and is a neighborhood of $U \subset \mathcal{M}$ such that $f(E_u)$ is open in $\mathbb{R}^n$.*

Intuitively, the tubular neighborhood of the manifold can be visualized as "inflating" the manifold, e.g., for a line in $\mathbb{R}^3$, a possible embedding of the tubular neighborhood would be a cylinder in $\mathbb{R}^3$ oriented around the line. It is known that in particular if $\mathcal{M}$ is manifold and $\mathcal{X} = \mathbb{R}^n$, then $\mathcal{M}$ has a tubular neighborhood in $\mathbb{R}^n$ as a consequence of the *tubular neighborhood theorem*.

Theorem 1. Tubular Neighborhood Theorem (Spivak 1999; Lee 2003). *Every embedded submanifold of $\mathbb{R}^n$ has a tubular neighborhood. Without loss of generality this tubular neighborhood is the* normal tubular neighborhood, *where the vector bundle ξ is the normal bundle of the embedded manifold.*

As the manifold is smooth, it makes sense that it can be "inflated" without the tubular region self-intersecting; a smooth manifold can only twist and curve so much, and has no self-intersections. However, note that a manifold can get arbitrarily close to itself, and thus while the tubular neighborhood does

exist it can be arbitrarily small. The volume of the tubular neighborhood, and thus the effectiveness of the projection operator, are dependent on the regularity of the constraint function.

We would like to show that all configurations within this region around the manifold, when projected, end up on the manifold. The gradient descent within the projection operator (Alg. 2) can be phrased as the flow of an ODE. Thus, we would like to show that, for some desired point on $\mathcal{M}$, that it is a stable solution, or *equilibrium point*, to the projection operator's ODE. The concept of *Lyapunov Stability* states this precisely.

Definition 12. Lyapunov Stability. *Let an* ODE $\dot{q} = g(q(t)), q(0) = q_0$ *be given, such that* $q(t) \in D \subseteq \mathbb{R}^n$ *for all* t *and* $g : \mathbb{R}^n \to \mathbb{R}^n$ *is continuous on* D. *An* equilibrium point $\bar{q}$ *is a point such that* $f(\bar{q}) = 0$. *Additionally, an equilibrium point* $\bar{q}$ *is:*

1. *Lyapunov Stable, if for every* $\varepsilon > 0$, *there exists a* $\delta > 0$ *such that* $\|q(0) - \bar{q}\|_2 < \delta$, *then for every* $t \geq 0$, $\|q(t) - \bar{q}\|_2 < \varepsilon$.
2. *Asymptotically Stable, if* $\bar{q}$ *is Lyapunov stable and there exists a* $\delta > 0$ *such that if* $\|q(0) - \bar{q}\|_2 < \delta$, $\lim_{t\to\infty} \|q(t) - \bar{q}\|_2 = 0$.

The Lyapunov Stability of an equilibrium point can be proven with the Lyapunov's theorem (Luenberger 1979). With these tools, we are now ready to prove that the projection operator covers the manifold.

Proposition 1. Projection Operator Coverage. *Let an ambient space* $\mathbb{R}^n$ *and a* C^2 *constraint function* F *be given, defining a* C^2 *implicit constraint manifold* $\mathcal{M}$. *For any open set* $U \subseteq \mathcal{M}$, *there exists an open set* $D_U \subseteq \mathbb{R}^n$ *(with non-zero measure) such that for all* $x \in U$, *there exists* $y \in D_U$ *such that* $P(y) = x$.

Proof. Let any open set $U \subseteq \mathcal{M}$ be given. By Thm. 1, as $\mathcal{M}$ is a C^2-smooth compact manifold in $\mathbb{R}^n$ there exists (h, ξ), the normal tubular neighborhood of $\mathcal{M}$, where $\xi = (\pi, E, \mathcal{M})$, E is the normal bundle of $\mathcal{M}$, and h is the smooth embedding of E in $\mathbb{R}^n$ such that $h(E)$ is an open neighborhood of $\mathcal{M}$ in $\mathbb{R}^n$. Note that as U is open in $\mathcal{M}$, we can restrict (h, ξ) to a partial tubular neighborhood (h, ξ, E_U), where $E_U = \bigcup_{x\in U} \pi^{-1}(x)$. Also, the Jacobian is of full rank over a non-empty open neighborhood $W \subseteq \mathbb{R}^n, \mathcal{M} \subset W$ by the inverse function theorem.

For all $x \in U$, we would like to show that there is some set $D_x \subset W$ such that for all $y \in D_x$, $P(y) = x$. Let $B_x = h(\pi^{-1}(x))$ denote all points in the tubular neighborhood that are closest to x (the embedding of the normal plane at x in $\mathbb{R}^n$). Put $D_x = B_x \cap W$, a non-empty open set in $\mathbb{R}^n$.

Following from the definition of a projection operator (Def. 6), define an ODE g over the domain D_x:

$$g(q) = -J(q)^+ F(q) \qquad \dot{q} = g\left(q(t)\right) \tag{6}$$

Note that g is continuous, as F is C^2 and thus the Jacobian J is a continuous map, and the pseudoinverse of the Jacobian is continuous as the Jacobian is of full rank for all $q(t)$ (Stewart 1969). Equilibrium points of (6) occur when $F(x) = 0$, i.e., when $x \in \mathcal{M}$.

Put V, the candidate Lyapunov function as:

$$V(q(t)) = \frac{1}{2}\|F\left(q(t)\right)\|_2{}^2.$$

V is continuous as the norm is a continuous map and g is continuous from above. V is also differentiable, with:

$$\nabla V(q(t)) = \left(J\left(q(t)\right)^T F\left(q(t)\right)\right)^T = F\left(q(t)\right)^T J\left(q(t)\right)$$

V satisfies the properties of the Lyapunov theorem on the domain D_x, as:

1. $V(q) = 0$ for $q \in \mathcal{M}$, and by construction D_x only contains x such that $x \in \mathcal{M}$,
2. $V(q) > 0$ as norms are positive definite, and
3. $\nabla V(q)g(q) < 0$ as:

$$\begin{aligned}\nabla V(q)g(q) &= -F\left(q\right)^T J\left(q\right) J\left(q\right)^+ F\left(q\right) \\ &= -\|F\left(q\right)\|_2{}^2.\end{aligned}$$

Thus x is an asymptotically stable equilibrium point on D_x for the ODE. Thus, for any $y \in D_x$, $P(y) = x$.

Put $D_U = \bigcup_{x\in U} D_x$. It is clear to see that $D_U = h(E_U) \cap W$ by construction, and thus D_U is a non-empty open set in $\mathbb{R}^n$. Additionally, for all $x \in U$, there exists $y \in D_U$ such that $P(y) = x$, as $D_x \subset D_U$ for all x. Thus, there exists an open set D_U for any $U \subseteq \mathcal{M}$ such that for all $y \in D_U$, $P(y) \in U$. □

With Prop. 1, it is clear to see that the ambient configuration space sampler paired with Alg. 2 covers the manifold, and is a surjective function. Projection-based sampling can be used for probabilistically-complete sampling of any manifold defined by a constraint function. Note that the projection operator acts like a topological retraction from a neighborhood D_U to U in the constraint manifold. As the ambient space sampler has a non-zero chance of sampling from any open set, there is a non-zero chance that for any open set $U \subset \mathcal{M}$, its corresponding open set $D_U \subset \mathbb{R}^n$ will be sampled. However, this also means that the probability of sampling from a particular set $U \subset \mathcal{M}$ is not related the measure of the set within the manifold, but the volume of its converging region, $\mu(D_U)$. Note

that D_U does not have to be a subset of $\mathcal{Q}_{\text{free}}$. For nearby sampling, as the distance metric used is the induced distance metric from $\mathbb{R}^n$, sampling within a ball in ambient space centered at a point on the manifold will at least contain a ball of the same radius on the manifold.

For the continuation-based space, theoretical guarantees about sampling are provided by (Henderson 2002; Jaillet and Porta 2013b), and are preserved within the context of IMACS. We assume that, if a sampling-based planner is expansive and has a non-zero probability of expanding to a state that will create a new chart, eventually the entire manifold will be covered by charts. Recall that continuation-based sampling will generate samples that create new charts due to the exploration parameter β (Section 4.3.2). As all portions of the manifold will be covered by charts whose validity regions apply to the manifold, continuation-based sampling (Alg. 6) will cover the manifold.

5.2 Integrator Convergence and Local Planning

In Ladd and Kavraki (2004), a local planner is defined as a measurable relation (on the Borel σ-algebra generated from the subset topology on $\mathcal{M}$) whose transitive closure is the path reachability relation. That is, the local planner is always capable of reaching points within a neighborhood of an original point, i.e., for any point we would like to reach q_b, there is an open set $U \subset \mathcal{M}$ such that all $q_a \in U$ can reach q_b. As discussed in Section 4.1.3, geodesic generation underlies local planning. The properties of the geodesic generation are thus properties of the local planner. Recall that the geodesic generators used in the projection- and continuation-based spaces are based upon integrating an ODE, in this case, the flow defined in Eq. 1. In this section, we show that formulating geodesic generation as an ODE satisfies the properties of being a measurable relation whose transitive closure is the path reachability relation. Thus, local planners in IMACS satisfy the necessary properties for probabilistic completeness of a planner.

We first present a theorem about the existence of solutions to ODEs on manifolds.

Theorem 2. Uniqueness and Existence Theorem for ODEs on Manifolds (Rheinboldt 1991). *Let $\bar{q}$ be an equilibrium point of a C^1-ODE g on a C^2-smooth manifold $\mathcal{X}$. There exists an open neighborhood of $\bar{q}$, $U \subset \mathcal{X}$ such that for all $q \in U$, $\bar{q}$ is a equilibrium point of g with q as an initial point.*

This theorem captures the intuition that, as manifolds are locally Euclidean, there is always a local neighborhood that behaves nicely for local planning. Now, we can show geodesic generation is a measurable relation, i.e., for any point in $\mathcal{M}$ there exists an open neighborhood that can reach it.

Proposition 2. Measurable Local Planners. *Let $q_b \in \mathcal{M}$ be given. There exists an open neighborhood of q_b, $U_b \subset \mathcal{M}$, such that for all $q_a \in U_b$, q_b can be reached by integrating Eq. 1 with $q(0) = q_a$ with either Alg. 3 or 7.*

Proof. Recall that $\mathcal{M}$ is a C^2-smooth manifold and Eq. 1 is a C^1 ODE by assumption. Set q_b as the constant in Eq. 1. It is clear that q_b is an equilibrium point of Eq. 1. Thus, the conditions of Thm. 2 are met, and for any $q_b \in \mathcal{M}$, there exists an open neighborhood $U_b \subset \mathcal{M}$ such that all $q_a \in U_b$ will converge to q_b. Additionally, as a bounded step size is used in integration (Alg. 3 and 7), the integrators will converge to the solution of Eq. 1 (Hairer et al. 2006). □

Thus, the local planners in the projection- and continuation-based space are measurable relations. Recall that $\mathcal{M}$ is imbued with the induced metric from $\mathbb{R}^n$. An obvious consequence of Prop. 2 is that for all points $q \in \mathcal{M}$, there exist open balls $B(q, \epsilon_l) \subseteq U_q$ of radius $\epsilon_l > 0$ such that all $q' \in B(q, \epsilon_l)$ can connect to q. The range of the local planner ϵ_l is highly sensitive to the volume of regions that converge, which is related to the regularity of the constraint function. That is, the more "curved" the manifold, the less likely the local planner is to succeed, but there always exists a region where it will succeed.

We now wish to show that, using these local planners, any configuration can be reached. Given that a path exists on $\mathcal{M}$, it must be realizable by a finite number of local plans.

Proposition 3. Local Planning To Path Reachability. *Let $q_a, q_b \in \mathcal{M}_{free}$ be given, and let a path $\sigma \in \Sigma_{\mathcal{M}_{free}}$ with strong δ-clearance be given such that $\sigma(0) = q_a, \sigma(1) = q_b$. There exists a finite sequence of points $\{q_i\}_{i=1}^k$ such that for all $i = 1, \ldots, k$ there is a t_i such that $\sigma(t) = q_i$, $t_i < t_{i+1}$ for all i. Additionally, q_{i+1} is reachable from q_i using a local planner (i.e., Alg. 3 or 7). Thus the transitive closure of Alg. 3 and 7 is the path reachability condition.*

Proof. By assumption, $q_a, q_b \in \mathcal{M}_{\text{free}}$ are given with a robust path $\sigma \in \Sigma_{\mathcal{M}_{\text{free}}}$ such that $\sigma(0) = q_a, \sigma(1) = q_b$.

From Prop. 2, we know that for all $t \in [0, 1]$, there exists an open neighborhood U_t of $\sigma(t)$ in $\mathcal{M}$ such that $\sigma(t)$ can be reached via a local planner from all $q \in U_t$. Additionally, as σ is robust, we know that there exists a ball of radius δ that is collision-free around all $\sigma(t)$. Define $\nu = \min\{\epsilon_l, \delta\}$, $\nu > 0$. Moreover, there exists open balls $B(\sigma(t), \nu) \subseteq U_t$.

As σ is of finite length L, it is possible to tile σ with a finite number $k = L/\nu$ of open balls of radius ν, such that $\sigma(t_n)$ is contained within the ball centered at $\sigma(t_{n+1})$ for all $n = 1, \ldots, k$ and $\sigma(t_1) = q_a, \sigma(t_k) = q_b$. From a configuration $\sigma(t_n)$, $\sigma(t_{n+1})$ is reachable without collision by a local planner for all t. This yields $\{q_i\}_{i=1}^{k}$, where $q_i = \sigma(t_i)$, a discrete path realizable by local planners. □

There are many planners (e.g., RRT, RRT*) that assume that a local planner only "steers" for a certain duration, $\eta > 0$. That is, the local planner needs to, at most, be able to connect to a point within a distance of η from any point, similar to the bound ϵ_l for local planners in IMACS. Thus, for guarantees to hold $\epsilon_l \geq \eta > 0$ as that is as far as the local planner is guaranteed to travel for any point. This fact is used in Section 5.3 and Section 5.4.

5.3 Probabilistic Completeness

In the sections above, we presented facts about sampling and local planning in IMACS for both the projection- and continuation-based spaces. These facts concerned the critical properties necessary of a sampler and local planner for a sampling-based planner to have probabilistic completeness. For concreteness, we now give a proof of probabilistic completeness for PRM within the context of IMACS. The proof of completeness for PRM follows from a proof offered in (Ladd and Kavraki 2004).

5.3.1 PRM within IMACS

Ladd and Kavraki (2004) provide a proof of the probabilistic completeness of PRM within a measure theoretic framework, offering the following theorem.

Theorem 3. Probabilistic Completeness of PRM (Thm. IV.2 in (Ladd and Kavraki 2004)). *If it is possible to find a path from q_a, q_b with a random walk using k points using a local planner, the expected number of samples needed for* PRM *to succeed (N) satisfies the following inequality:*

$$\mathbb{E}[N] \leq \frac{H(k)}{p}$$

Where $H(k)$ is the k^{th} harmonic number and $p > 0$ is the minimum probability of sampling one of the necessary k points of the path (a problem specific constant).

Following from analysis presented in (Ladd and Kavraki 2004) and Prop 3, it is easy to see that a random walk from q_a, q_b on $\mathcal{M}$ using sampling and local planning within IMACS is probabilisitically complete for finding paths with strong δ-clearance. This is because 1) sampling in IMACS has a distribution with non-zero density over $\mathcal{M}$ and 2) it always possible to travel at least ϵ_l with local planning in IMACS (i.e., local planning is a measurable relation). Thus, Thm. 3 applies to the projection- and continuation-based spaces.

For the projection-based space, it is not known *a priori* what the volume is of an open neighborhood D_U that projects to an open ball $B_{\mathcal{M}}(q, \epsilon) \subseteq \mathcal{M}$. A conservative estimate can be made by taking the minimum volume over all open balls of radius ϵ on $\mathcal{M}$:

$$\zeta_{\mathcal{M},\epsilon} = \min_{q \in \mathcal{M}} \mu \left(\bigcup_{q \in B_{\mathcal{M}}(q,\epsilon)} D_q \right) \tag{7}$$

That is, for any $q \in \mathcal{M}$, there is a volume of measure $\zeta_{\mathcal{M},\epsilon}$ that projects to the open ball of radius ϵ around q. We can now state a bound for the projection-based space:

Proposition 4. PRM in projection-based space. *For a path with strong δ-clearance and length L, with projection-based sampling and a local planner with range ϵ_l, set $\gamma = \min\{\delta, \epsilon_l\}$. The expected number of samples needed for* PRM *to succeed (N) satisfies the following inequality:*

$$\mathbb{E}[N] \leq \frac{H(L/\gamma)\mu(\mathcal{Q})}{\zeta_{\mathcal{M},\gamma}}$$

Proof. Let σ be a path with strong δ-clearance. As σ has finite length L, can be tiled with $k = L/\gamma$ balls of radius γ centered at configurations $q_1, \ldots q_k$. The projection-based local planner can connect to the center of each of these balls, as they are no farther than ϵ_l away, and thus a random walk can realize the path. By definition, $\zeta_{\mathcal{M},\gamma} \leq \min_{i=1,\ldots,k} \mu \left(\bigcup_{q \in B_{\mathcal{M}}(q_i,\gamma)} D_q \right)$. Thus, set $p = \zeta_{\mathcal{M},\gamma}/\mu(\mathcal{Q})$ and invoke Thm. 3. □

For the atlas-based space, we use theory developed in (Jaillet and Porta 2012). Importantly, the following bound is presented for two points $q_i, q_j \in \mathcal{M}$ parameterized by a chart C, $u_i = {\psi_C}^{-1}(q_i), u_j = {\psi_C}^{-1}(q_j)$:

$$\|q_i - q_j\| \leq \sec(\alpha) \|u_i - u_j\| \tag{8}$$

Where α is the maximum curvature of a chart against the manifold.

Proposition 5. PRM in continuation-based space. *For a path with strong δ-clearance and length L, with continuation-based sampling and a local planner with range ϵ_l using an atlas A, set $\gamma = \min\{\delta, \epsilon_l\}$. The expected number of samples needed for* PRM *to succeed (N) satisfies the following inequality:*

$$\mathbb{E}[N] \leq \frac{H(L/\gamma)\mu_A(A)}{\mu_A(B_A(\cdot, \sec(\alpha)\gamma)}$$

Where $\mu_A(\cdot)$ is the Lebesgue measure in the charts $C \subset \mathbb{R}^{n-k}$ and B_A is a ball in a chart C, $\mu_A(B_A(\cdot, r)$ is the measure of an open ball of radius r in the atlas, and $\mu_A(A) = \sum_{C \in \mathcal{A}_\mathcal{M}} \mu_A(C)$ is the total measure of the atlas.

Proof. As before, let σ be a path with strong δ-clearance. As σ has finite length L, can be tiled with $k = L/\gamma$ balls of radius γ centered at configurations $q_1, \dots q_k$. The continuation-based local planner can connect between each of these balls, as they are no farther than ϵ_l away, and thus a random walk can realize the path. Without loss of generality, assume a ball on the manifold $B_\mathcal{M}(q, \gamma)$ is fully parameterized by a chart C. From Eq. 8, the ball on the manifold is contained within a ball in the chart $B_A({\psi_C}^{-1}(q), \sec(\alpha)\gamma)$. Thus, set $p = \mu_A(B_A(\cdot, \sec(\alpha)\gamma))/\mu_A(A)$ and invoke Thm. 3. □

5.3.2 Discussion Although here we present a proof of probabilistic completeness for PRM, the core of the proof applies to proofs for many other planners. This is because sampling, local planning, and metrics all behave nicely over $\mathcal{M}$, and local properties hold. It is always possible to sample in a particular ball on $\mathcal{M}$ as IMACS affords sampling with non-zero probability over all of $\mathcal{M}$. The local planner always is capable of connecting to points within a range $\eta > 0$. Finally, the induced metric is still a metric which underestimates with respect to geodesic distance on $\mathcal{M}$, and the triangle inequality holds.

We briefly illustrate how the building blocks of sampling and local planning can be applied to other proofs of completeness. Similar to the proofs presented here, (Kleinbort et al. 2019) presents a proof of probabilistic completeness of RRT. Without going into to details, the proof of probabilistic completeness of geometric RRT hinges on Lemma 1, which states that a local plan from a nearby random sample to its nearest neighbor lies entirely in $\mathcal{Q}_{\text{free}}$. As local planning in IMACS can connect all points within a distance η and uses the induced Euclidean metric, Lemma 1 in holds. Thus, Thm. 1 in (Kleinbort et al. 2019) (the convergence guarantee of RRT) holds as well (albeit with different constants that related to the probability of sampling a specific).

5.4 Asymptotic Optimality

Asymptotically optimal sampling-based planners not only find a feasible solution, but as the number of iterations of the planner go to infinity, find the globally optimal solution. The algorithms RRT* and PRM* (Karaman and Frazzoli 2011) are of particular interest, as they are the basis for many other asymptotically optimal algorithms and good case studies for how IMACS affects the necessary properties of an asymptotically optimal planner. Within this work, we describe how IMACS affects the proofs of asymptotic optimality for RRT* and PRM*; the features relevant that are affected by IMACS are the sampler, metric, and local planner. We walk through the affected portions of the proofs presented in (Karaman and Frazzoli 2011) and show that their results hold given our assumptions about the metric and local planner within IMACS.

5.4.1 Prior Results Karaman and Frazzoli (2011) prove the following bound as the connection radius of RRT*:

$$\gamma_{\text{RRT}^*} > \left(2\left(1 + \frac{1}{n}\right)\frac{\mu(\mathcal{Q}_{\text{free}})}{\zeta_n}\right)^{\frac{1}{n}}, \tag{9}$$

where $\zeta_n = \mu(B(\cdot, 1))$ is the volume of the unit ball in $\mathbb{R}^n$, and n is the dimension of the configuration space. Additionally, Karaman and Frazzoli (2011) proves the following bound for the connection radius for PRM*:

$$\gamma_{\text{PRM}^*} > 2\left(\left(1 + \frac{1}{n}\right)\frac{\mu(\mathcal{Q}_{\text{free}})}{\zeta_n}\right)^{\frac{1}{n}} \tag{10}$$

With respect to manifold constrained motion planning, Jaillet and Porta (2012) presented AtlasRRT*, an asymptotically optimal constrained sampling-based algorithm. AtlasRRT* adapts the RRT* algorithm with a continuation-based method for local planning and sampling. The following bound is given for the critical radius of AtlasRRT*:

$$\gamma_{Atlas\text{RRT}^*} > \left(2\left(1 + \frac{1}{n-k}\right)\frac{\mu(A_{\text{free}})}{\zeta_{n-k}}\sec(\alpha)\right)^{\frac{1}{n-k}}, \tag{11}$$

where μ_A is the Lebesgue measure in the atlas (i.e., in $\mathbb{R}^{n-k}$), $n-k$ is the dimension of the manifold, and $\mu_A(A_{\text{free}}) = \sum_{C \in \mathcal{A}_\mathcal{M}} \mu_A(C_{\text{free}})$ is the total measure of the free space of the applicability regions of the charts within the atlas $\mathcal{A}_\mathcal{M}$.

$\gamma_{Atlas\text{RRT}^*}$ is in terms of the measure on the atlas, an approximation of the $n-k$ dimensional manifold. Thus, the dimension exponent of $n-k$ is used together with $\sec(\alpha)$ to account for approximation error from maximum allowed curvature of a chart, α (see Eq. 8).

5.4.2 RRT within IMACS* As mentioned in (Jaillet and Porta 2012), the asymptotic optimality of RRT* is dependent on the ability of the planner to sample and connect configurations that are close to the optimal path. Importantly, the proof of RRT* is not dependent on the local planner being able to connect any two configurations. In fact, within (Karaman and Frazzoli 2011), the proof of RRT*'s asymptotic optimality only

relies on the local planner's steering parameter η being positive. We showed within Section 5.2, for any configuration $q \in \mathcal{M}$, the local planner within IMACS always has an open neighborhood of configurations that can connect to q. For the purposes of the proof of RRT*'s asymptotic optimality, the minimum radius of all such open neighborhoods corresponds to some steering parameter $\eta' > 0$. What value η' takes is determined by the curvature of the manifold, but this value only affects the convergence of RRT*, not its guarantees of asymptotic optimality.

For the projection-based space, the connection radius is affected by the measure of set of states that project onto the free parts of the manifold, as samples are drawn from $\mathbb{R}^n$. The set of states that projects onto $\mathcal{M}_{\text{free}}$ is $\bigcup_{q \in \mathcal{M}_{\text{free}}} D_q$, where D_q is the set of configurations that project to a configuration $q \in \mathcal{M}$ (from Prop. 1). Thus, $\bigcup_{q \in \mathcal{M}_{\text{free}}} D_q$ is used rather than $\mathcal{Q}_{\text{free}}$ as the bound in Eq. 9. Additionally, note that ζ_n appears in the bound as a ratio with the measure of the free space to capture the probability that two vertices have been connected by RRT*. Thus, $\zeta_{\mathcal{M},1}$ (Eq. 7) is used rather than ζ_n, as $\zeta_{\mathcal{M},1}$ is the guaranteed volume that will project to any open unit ball on the manifold. Within IMACS, we claim that the connection radius for RRT* in IMACS is the following.

$$\gamma_{\text{RRT}^*\text{+IMACS}} > \left(2 \left(1 + \frac{1}{n} \right) \frac{\mu\left(\bigcup_{q \in \mathcal{M}_{\text{free}}} D_q \right)}{\zeta_{\mathcal{M},1}} \right)^{\frac{1}{n}} \tag{12}$$

Moreover, Eq. 11 can be applied to RRT* within the continuation-based space within IMACS, as the fundamentals of sampling and local planning are equivalent to those within AtlasRRT*.

5.4.3 PRM within IMACS* Critically for PRM*, IMACS affects the ability for a planner to connect subsequent vertices along the optimal path, as local planning within IMACS is only guaranteed for a neighborhood around a configuration. This is captured in the proof of PRM*'s asymptotic optimality (Karaman and Frazzoli 2011) as an arbitrary constant $\theta_1 > 0$, which determines the radius and distance of balls that tile the optimal path. Specifically, for a path with strong δ-clearance, the center of the balls used within throughout the proof are at most $\frac{\theta_1}{1+\theta_1}\delta$ away from each other. Thus, for a local planner in IMACS with a given range, there is a value of θ_1 such that each successive configuration can be connected; the proof of PRM*'s asymptotic optimality is not affected by the local planners within IMACS. However, as with RRT*, convergence of the method will be affected due to failures from local planning.

A bound similar to RRT* is claimed for PRM* in the projection-based space in IMACS:

$$\gamma_{\text{PRM}^*\text{+IMACS}} > 2 \left(\left(1 + \frac{1}{n} \right) \frac{\mu\left(\bigcup_{q \in \mathcal{M}_{\text{free}}} D_q \right)}{\zeta_{\mathcal{M},1}} \right)^{\frac{1}{n}} \tag{13}$$

Again, the critical features of sampling points are affected by the projection-based sampling method as well as the use of the ambient metric. Thus, the volume of $\bigcup_{q \in \mathcal{M}_{\text{free}}} D_q$ is relevant rather than the volume $\mathcal{Q}_{\text{free}}$. A bound is claimed for the continuation-based space as well, following (Jaillet and Porta 2012):

$$\gamma_{\text{PRM}^*\text{+IMACS}} > 2 \left(\left(1 + \frac{1}{n-k} \right) \frac{\mu(A_{\text{free}})}{\zeta_{n-k}} \sec(\alpha) \right)^{\frac{1}{n-k}} \tag{14}$$

Observe that the bound is again in terms of measure on the atlas (giving the distortion constant $\sec(\alpha)$), and that the dimension exponent in $n - k$.

6 Experimental Results

IMACS is implemented within the Open Motion Planning Library (OMPL) (Şucan et al. 2012), an abstract library with implementations of many popular sampling-based planning algorithms. IMACS fits neatly within OMPL's notion of a *state space*, and is implemented as such. In particular, IMACS is implemented as a state space that takes as input the ambient configuration space (also a state space) and a constraint function. Each of IMACS's state spaces use the underlying subroutines offered by the ambient state space in tandem with their method of constraint adherence. No modification was necessary to any of the planning algorithms for them to work with IMACS. All of the code for the framework is available within OMPL, along with code to reproduce the most of the results presented in this section*.

All benchmarks were done with a single set of parameters for each constrained space and planning algorithm, to preserve fairness across multiple environments. More performance could have been gained by tuning these for each problem, but a set of reasonable defaults is desirable, especially from a naïve user's perspective. All benchmarks were performed on workstations with an Intel® Core™ i7-6700K processor and 32GB of DDR4 RAM at 2400MHz. The experiments shown here are meant to both demonstrate the effectiveness of the

*Available at http://ompl.kavrakilab.org. Tutorials and documentation specific to IMACS can be found at http://ompl.kavrakilab.org/constrainedPlanning.html.

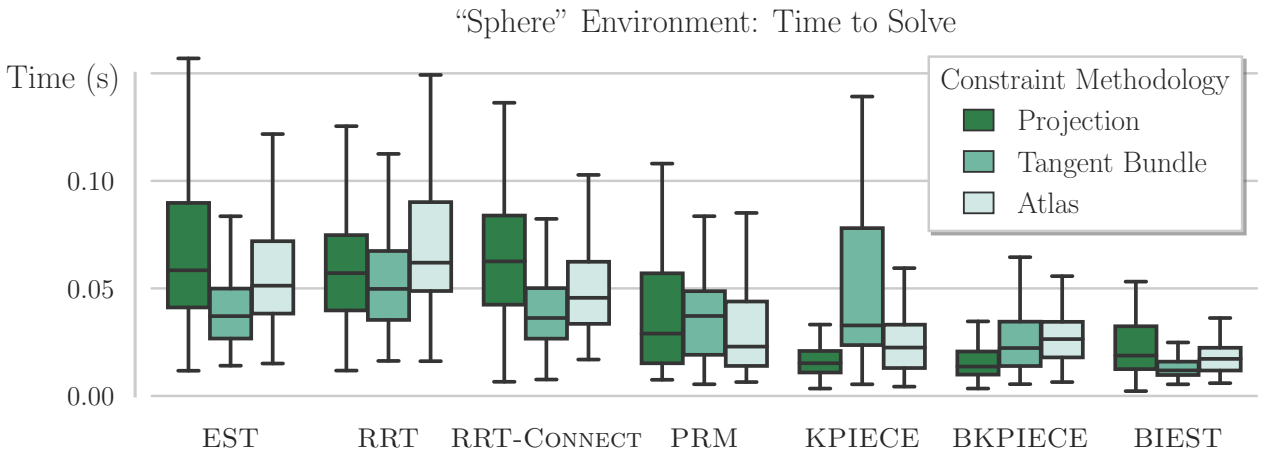


Figure 5. Timing results from 100 runs of each planner in the "sphere" environment (Figure 6) for the projection-, tangent bundle-, and atlas-based spaces in IMACS. Planners tested are EST, KPIECE, their bidirectional variants BIEST and BKPIECE (Hsu et al. 1999; Şucan and Kavraki 2008), RRT (LaValle and Kuffner 2001) and RRT-Connect (Kuffner and LaValle 2000), and PRM (Kavraki et al. 1996). CBIRRT2, TB-RRT, and AtlasRRT are emulated by RRT-Connect in their respective constrained space, and have relatively poor performance.

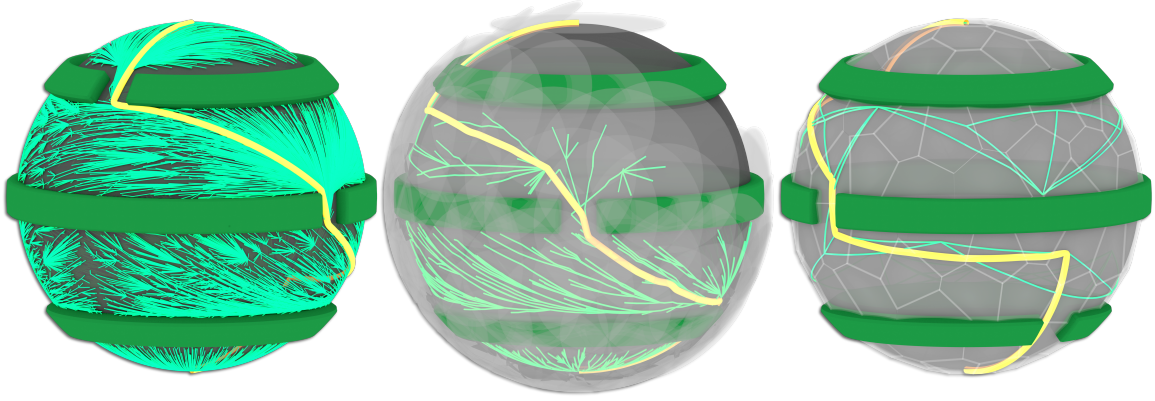


Figure 6. The "sphere" environment, from three perspectives. The sphere constraint manifold (grey) with obstacles (dark green). The solution path (yellow) runs from the south to north pole. From left to right, projection-based RRT* (Karaman and Frazzoli 2011) motion graph (green), tangent bundle-based BIT* (Gammell et al. 2015) motion graph and tangent spaces (grey), and atlas-based SPARS (Dobson and Bekris 2014) motion graph and tangent polytopes.

planning system as well as illustrate concepts that help put the work in context.

Within the presented results, we refer to three methods of constraint adherence running in IMACS: projection-, tangent bundle-, and atlas-based spaces. The projection-based space is as described in Section 4.2. The tangent bundle-based space is as described in Section 4.3 and uses the LAZYINTEGRATOR from Alg. 7 for geodesic generation. Similarly, the atlas-based space uses INTEGRATOR from Alg. 7 for geodesic generation, and generates . Note that the tangent bundle-based space does not generate separating halfspaces between charts, but the atlas-based space does (see Section 4.3.1).

A critical aspect of constrained planning is the computation of the Jacobian of the constraint function as well as computing solutions to the (pseudo)inverse of the Jacobian. Generally, analytic solutions to the Jacobian are preferable for efficiency. However, within our experiments, a few problems use numerical differentiation to compute the Jacobian when an analytical solution is not provided. Within our implementation of IMACS, the SVD decomposition is used in the implementation of the projection operator

(Alg. 2), and the QR decomposition is used within ψ_{C_q} (Alg. 4).

We performed a collection of experiments that demonstrate the important contribution IMACS represents: different planners and different methods of constraint adherence are better at different problems, IMACS is efficient and can be used for realistic problems, and planners within IMACS have similar performance to specialized constrained planners. The experiments include simple point robots in $\mathbb{R}^3$, high-dimensional manipulators implicitly defined by a constraints, and a realistic application with NASA's Robonaut 2.

6.1 "Sphere" Environment

Figure 6 shows the "sphere" environment, a two-dimensional manifold embedded within $\mathbb{R}^3$, defined by the constraint function $F(q) = \|q\| - 1$. For this environment, the robot is a point robot. The robot must traverse three longitudinal obstacles, each with a narrow passage, to move from the south to the north pole. We show the results of 100 runs of various motion planners within IMACS in Figure 5. As shown in the figure, combinations of planners and constrained spaces within the framework have dramatically different effects on planning time. Previous approaches in the literature are emulated by RRT-Connect within IMACS (e.g., CBIRRT2, TB-RRT, AtlasRRT), which is shown to have the relatively poor performance overall within the "sphere" environment. For this problem, other combinations of planners and methods of constraint adherence (e.g., tangent bundle-based BIEST) would be a better selection of planner if speed was the primary concern.

There is little work on optimizing with respect to some cost function in tandem with manifold constraints. AtlasRRT* (Jaillet and Porta 2013a) is an asymptotically optimal algorithm that adheres to manifold constraints, but requires specialized implementation and integration with a method of constraint adherence. Within IMACS, no additional overhead is necessary for asymptotically optimal planning, as shown in Figure 6, which shows motion graphs for three asymptotically optimal and near-optimal planners. Additionally, path smoothing, shortening, hybridization, and interpolation algorithms work with no knowledge of constraints, as all subroutines are handled within IMACS. As proven in Section 5.4, asymptotically optimal planners retain their guarantees within IMACS.

6.2 "Torus" Environment

In constrained planning problems, not just the planner matters when solving a constrained problem;

Figure 7. The "torus" environment. The torus constraint manifold (grey) with an overlaid obstacle maze and atlas. The resulting graph (green) of a run of atlas-based PRM along with a solution path (yellow) is shown on the torus.

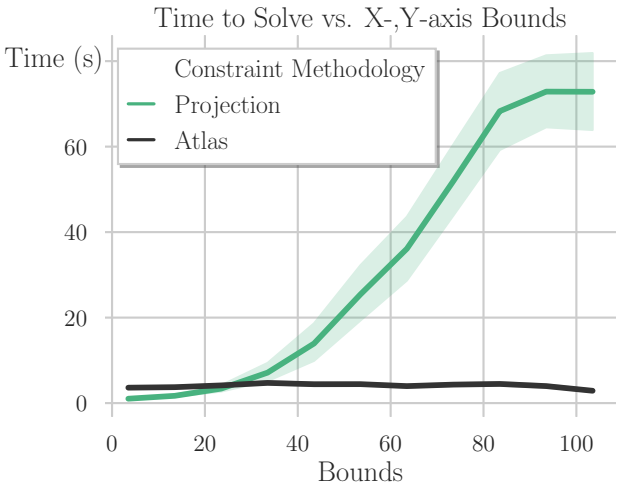


Figure 8. Timing results from 100 runs of PRM in the "torus" environment (Figure 7) using the atlas- and projection-based constrained space versus the size of the x- and y- axes of the ambient configuration space. Projection-based PRM performs orders of magnitude worse than its atlas-based counterpart if the ambient space has a large volume, but is better when the ambient space is a tight bounding box.

the ambient configuration space can dramatically affect performance. Consider a "torus" environment (Figure 7), which is a two-dimensional manifold embedded within $\mathbb{R}^3$, with a constraint function

$$F(q) = (3 - \sqrt{x^2 + y^2})^2 + z^2 - 2.$$

For this environment, the robot is a point robot. The planning problem is to traverse a maze that is wrapped onto the surface of the torus. Timing results for the PRM planner using the projection- and atlas-based methods are shown in Figure 8, where the total volume of the configuration space is varied while the size of the torus remains constant. The volume of the configuration space is varied by increasing the bound b on the $X-$ and $Y-$ axes of the ambient configuration space (given by $[-b, b] \times [-b, b] \times [-2, 2] \subset \mathbb{R}^3$), with the bound b ranging from a tight bounding box with $b = 3.5$ to a very expansive box with $b = 103.5$.

As shown by the results in Figure 8, projection-based planning performs orders of magnitude worse than its atlas-based counterpart and worsens as the volume of the space expands, due to the inefficiency of sampling configurations that mostly project to the outer surface of the torus. The atlas-based method, which samples directly from an approximation of the manifold, is unaffected by changes in the ambient configuration space. Projection to the inner surface of the torus requires sampling inside of the hole of the torus, which becomes less likely as ambient space expands. The torus example is illustrative of a problem that might arise on real robotic manipulators, as configuration spaces with revolute joints are toroidal in topology. It is unknown *a priori* how obstacles in the environment will interact with constraints, and no one method of constraint adherence is equipped to handle every case. Therefore, the ability to change method of constraint adherence independently of a planner is essential to efficiently planning within different environments.

6.3 "Implicit Chain" Environment

A general trend observed by the authors is that as a planning problem becomes more constrained or the implicit manifold more curved with respect to the ambient space, the atlas- and tangent bundle-based methods become more effective. This is because the extra computation to maintain the approximation pays off. However, as the dimensionality of the problem grows, the approximation is less helpful and requires a similar, amortized amount of work as projection does. These are not rules written in stone, and there are many problems which belie their guidance. Take for example the problem of an "implicit chain," shown in Figure 9. Here, the kinematics are modeled as distance constraints, one for each link, on a chain with 5 spherical joints. The ambient configuration space is thus $\mathbb{R}^{3\times5}$. To further increase problem complexity, we impose the following additional constraints: (a) the end-effector is constrained to the surface of a sphere of radius three, (b) joint 1 and 2 must have the same z-value, (c) joint 2 and 3 must have the same x-value, and (d) joint 3 and 4 must have the same z-value. This gives an implicit manifold dimension of six. Timing results for this problem are shown in Figure 9. When there are no obstacles in this scene, atlas- and tangent bundle-based methods perform the best with KPIECE, while other methods with both RRT-Connect and KPIECE are competitive. However, as obstacles are added to the surface of the outer sphere, RRT-Connect performs worse, with only the projection-based space solving the problem within time limit. The performance drop is likely due to the narrow passage created by the obstacles, a common problem for RRT-Connect. For KPIECE, the relative performance of the atlas- and tangent bundle-based methods drops in comparison to the projection-based space.

6.4 "Parallel Implicit Chain" Environment

One motivating factor of this work was extending constrained planning to high-dimensional spaces, taking advantage of previous approaches in high-dimensional planning without any additional cost. In Figure 10, we show the "implicit parallel manipulator" environment,

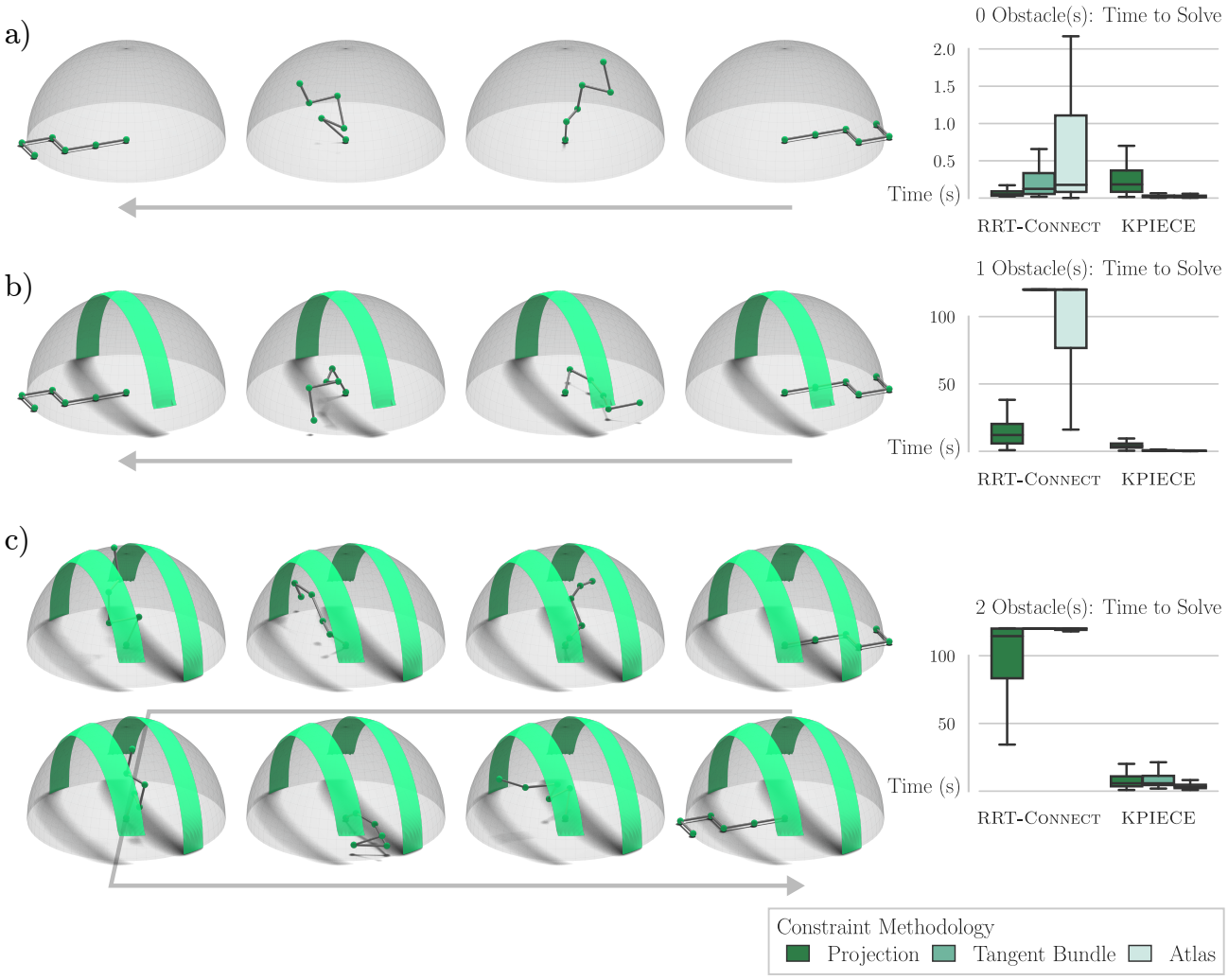


Figure 9. The "implicit chain" environment. Sample solution paths with zero (a), one (b), and two surface obstacles (c) with antipodal narrow passages are shown. Additionally, timing results for RRT-Connect and KPIECE using each constrained space for each obstacle amount are shown on the right. 100 runs were used for each combination of planner and constrained space, with a timeout of 120 seconds. Note that the Y-axis changes on the plots.

a parallel manipulator defined with a set of the "implicit chains" previously. The end-effectors of the chains are constrained to remain attached to a shared disk, creating dependencies in their motion. The environment shown has eight chains with seven links each, for a total ambient space dimensionality of 168. The constraint manifold is of dimension 99.

Recall that emulated prior works (CBIRRT2, TB-RRT, AtlasRRT) all use RRT-Connect as their base planner. RRT-Connect within the projection-, tangent bundle-, and atlas-based spaces was unable to successfully solve this system given 10 minutes of planning time over 100 trials for each space. Using the KPIECE planner designed for high-dimensional spaces with the projection-based space, we can quickly solve this problem while adhering to constraints, for a median time of 14.5 seconds over 100 runs. As IMACS decouples the choice of planner and method for constraint adherence, IMACS enables more effective planners to be used for complex high-dimensional planning problems such as this.

6.5 Robonaut 2

Finally, we demonstrate the framework on a real robotic system. Figure 11 shows steps through a sequence of constrained motion plans for NASA's Robonaut 2 (R2) (Diftler et al. 2011); R2 is tasked with walking through a narrow module on the International Space Station (ISS), with a few pieces of fixed, stray debris within. R2 is an excellent application for constrained motion planning due to the microgravity environment

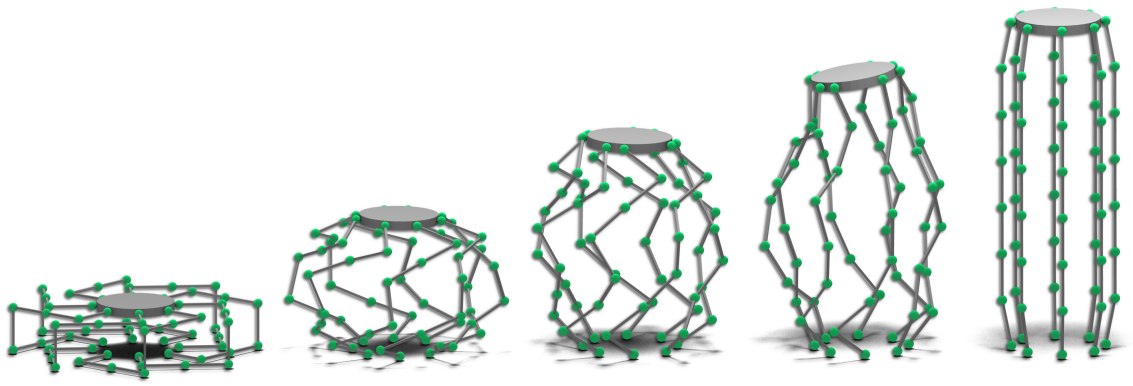


Figure 10. An plan in the "implicit parallel manipulator" environment. The goal is to move from a flat, rotated configuration to upright. This path was computed using KPIECE in a projection-based constrained space in a median time of 14.5 seconds.

of the ISS, which allows for R2 to climb across handrails without typical humanoid worries such as dynamic stability. R2 maintains the quasistatic assumption throughout all motions.

In the above scenario, R2 is modeled as a 15 DOF system with two 7 DOF legs and a rotating waist joint. Additionally, R2 has a floating joint (free-flying motion in $SE(3)$) at the root of its kinematic tree so it can translate and rotate freely in space, bringing the total dimensionality of the system to 21 DOF. Within Figure 11, stills from a sequence of motion plans executed are shown. To walk through the ISS module, six motion planning problems compose a sequence of steps that walk R2 from the left to the right of the module. Each of these motions must avoid the fixed floating debris and the walls of the habitation module while respecting constraints. There are a variety of constraints imposed on each of the six motion planning problems:

- R2 must keep its torso upright at a fixed orientation,
- One leg end-effector must remain fixed in position and orientation where it has grasped a handrail, and
- Whenever stepping towards an end-effector goal with the same orientation as the start, the moving foot must remain facing the same fixed orientation, so cameras within the end-effector can more accurately track handrail location for grasping.

An implementation of CBIRRT2 was compared against RRT-Connect running in the projection-based IMACS space, using the *MoveIt!* motion planning framework (Şucan and Chitta 2011). Throughout the six motions, comparable timing performance was witnessed across 20 runs of each planner, with average planning times less than 5 seconds for both planners.

Functionally, the implementations of CBIRRT2 and RRT-Connect with the projection-based space in IMACS are nearly identical, with minor differences in how new states are generated via interpolation. Thus, the comparable performance of the planners is expected, and shows that in realistic scenarios there is no performance degradation using IMACS and an unconstrained planner.

7 Conclusion

We have introduced IMACS (Implicit MAnifold Configuration Space), a novel framework for sampling-based planning under manifold constraints that decouples constraint adherence from motion planning algorithms. With IMACS, the choice of motion planner and method for constraint adherence are orthogonal, enabling novel combinations of planner and method of constraint adherence that perform more effectively than previously proposed combinations. We have demonstrated IMACS's capability by showing projection- and continuation-based methods of constraint adherence inspired by three state-of-the-art constrained sampling-based planners, CBIRRT2, TB-RRT, and AtlasRRT. Additionally, we have tested a broad range of sampling-based planners within IMACS for a set of constrained problems and shown that each planner can operate effectively within IMACS. Furthermore, we have provided theoretical guarantees that IMACS preserves the probabilistic completeness and asymptotic optimality of planners running within IMACS. IMACS is easily extended to new planners, and new constraint spaces can be adapted to the framework as its concepts are general to constrained planning (e.g., local tangent space methods).

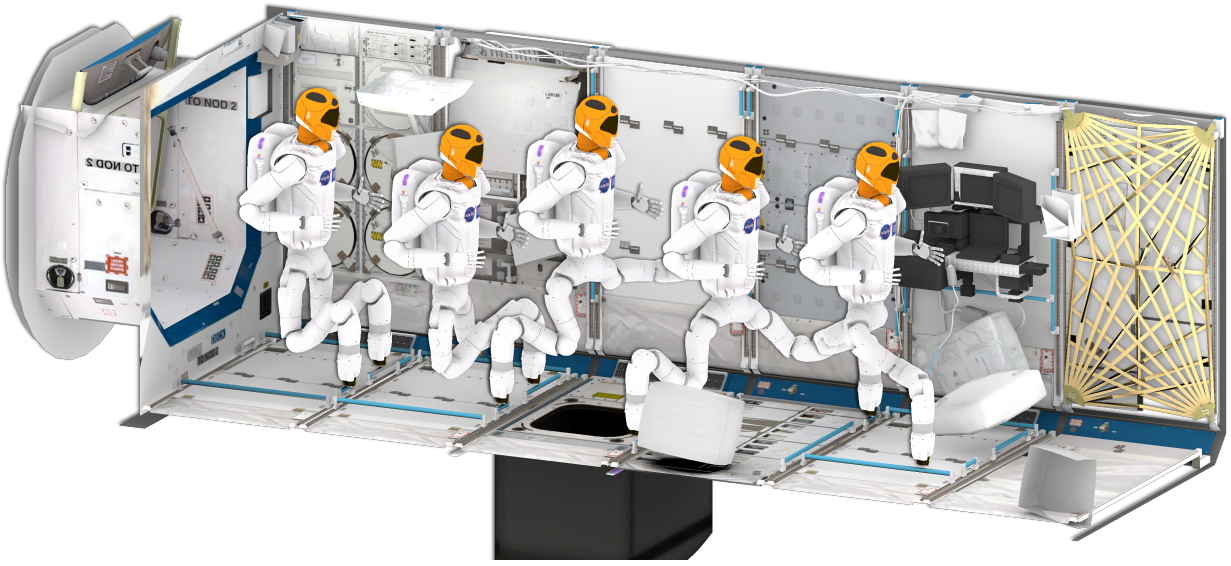


Figure 11. Stills from a sequence of motion plans executed by NASA's Robonaut 2 (R2) within a model of the International Space Station (sides of the module are removed for visualization purposes). R2 is modeled as a 15 DOF system with two 7 DOF legs and a rotating waist joint. Additionally, an $SE(3)$ element is added so the robot can float freely through the space. There are six separate motion planning problems in the sequence, each consisting of R2 taking a step. Each step with a variety of constraints imposed: (1) all steps required the torso to remain at a fixed orientation (2) all steps fixed the position and orientation of the foot grasped to a handrail and (3) where possible, the moving foot was required to face downward at a fixed orientation (a realistic requirement so the cameras within the feet can track handrail location).

Although there are rough guidelines on when different constrained planning approaches tend to work better than others, for specific problems it is difficult to predict which combination of constraint space and planner will work the best. This further highlights the benefit of decoupling constraints from planning. Additionally, the methods as presented here are dependent on some hyperparameters, such as the manifold integrator step size and chart validity parameters. Automatically tuning or adapting these parameters online while maintaining guarantees is left for future work.

Planning under manifold constraints is an essential capability for manipulation planning, task and motion planning, and multi-modal planing. Sampling-based planning with IMACS could be used as a subroutine for planning under constraints for these manipulation algorithms. The idea of IMACS is also extendable to the case of kinodynamic planning under manifold constraints. Finally, given the proof of concept on R2, we are in the process of integrating the framework for general robotic platforms.

Acknowledgements

We would like to thank the anonymous reviewers, Andrew Wells, and William Cannon Lewis III for their helpful comments. We would like to thank Caleb Voss for his preliminary work (Voss et al. 2017). We would also like to thank Dr. Julia Badger and the Robonaut 2 team. ZK, MM, and LEK are supported in part by NSF IIS 1317849 and Rice University funds. ZK is also supported by a NASA Space Technology Research Fellowship 80NSSC17K0163.

References

Alexopoulos C and Griffin PM (1992) Path planning for a mobile robot. *IEEE Trans. on Systems, Man, and Cybernetics* 22(2): 318–322.

Aristidou A and Lasenby J (2011) FABRIK: A fast, iterative solver for the inverse kinematics problem. *Graphical Models* 73(5): 243–260.

Asano T, Asano T, Guibas L, Hershberger J and Imai H (1985) Visibility-polygon search and Euclidean shortest paths. In: *Annual Symp. on Foundations of Computer Science*. pp. 155–164.

Atkeson CG, Babu BPW, Banerjee N, Berenson D, Bove CP, Cui X, DeDonato M, Du R, Feng S, Franklin P, Gennert M, Graff JP, He P, Jaeger A, Kim J, Knoedler K, Li L, Liu C, Long X, Padir T, Polido F, Tighe GG and Xinjilefu X (2015) NO FALLS, NO RESETS: Reliable humanoid behavior in the DARPA robotics challenge. In: *Int. Wksp. on the Algorithmic Foundations of Robotics*. Springer, pp. 623–630.

Barraquand J and Latombe JC (1993) Nonholonomic multibody mobile robots: Controllability and motion

planning in the presence of obstacles. *Algorithmica* 10(2): 121–155.

Berenson D, Siméon T and Srinivasa SS (2011a) Addressing cost-space chasms in manipulation planning. In: *IEEE Int. Conf. Robot. Autom.* pp. 4561–4568. DOI:10.1109/ICRA.2011.5979797.

Berenson D, Srinivasa SS and Kuffner JJ (2011b) Task space regions: A framework for pose-constrained manipulation planning. *Int. J. of Robotics Research* 30(12): 1435–1460. DOI:10.1177/0278364910396389.

Bonalli R, Bylard A, Cauligi A, Lew T and Pavone M (2019) Trajectory optimization on manifolds: A theoretically-guaranteed embedded sequential convex programming approach. In: *Robotics: Science and Syst.*

Bonilla M, Pallottino L and Bicchi A (2017) Noninteracting constrained motion planning and control for robot manipulators. In: *IEEE Int. Conf. Robot. Autom.* pp. 4038–4043. DOI:10.1109/ICRA.2017.7989463.

Bordalba R, Ros L and Porta JM (2018) Randomized kinodynamic planning for constrained systems. In: *IEEE Int. Conf. Robot. Autom.* pp. 7079–7086. DOI:10.1109/ICRA.2018.8460753.

Burget F, Hornung A and Bennewitz M (2013) Whole-body motion planning for manipulation of articulated objects. In: *IEEE Int. Conf. Robot. Autom.* pp. 1656–1662.

Buss SR (2004) Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods. *IEEE J. Robot. Autom.* 17(1-19): 16.

Canutescu AA and Dunbrack RL (2003) Cyclic coordinate descent: A robotics algorithm for protein loop closure. *Protein Science* 12(5): 963–972.

Cefalo M, Oriolo G and Vendittelli M (2013) Task-constrained motion planning with moving obstacles. In: *IEEE/RSJ Int. Conf. on Intell. Robots and Syst.* pp. 5758–5763.

Choset HM, Hutchinson S, Lynch KM, Kantor G, Burgard W, Kavraki LE and Thrun S (2005) *Principles of Robot Motion: Theory, Algorithms, and Implementation*. MIT Press. ISBN 0-262-03327-5.

Cortés J, Siméon T and Laumond JP (2002) A random loop generator for planning the motions of closed kinematic chains using PRM methods. In: *IEEE Int. Conf. Robot. Autom.*, volume 2. pp. 2141–2146.

Dai S, Orton M, Schaffert S, Hofmann A and Williams B (2018) Improving trajectory optimization using a roadmap framework. In: *IEEE/RSJ Int. Conf. on Intell. Robots and Syst.* IEEE, pp. 8674–8681. DOI:10.1109/IROS.2018.8594274.

Dantam NT, Kingston Z, Chaudhuri S and Kavraki LE (2018) An incremental constraint-based framework for task and motion planning. *Int. J. of Robotics Research* 37(10): 1134–1151. DOI:10.1177/0278364918761570.

Diftler MA, Mehling JS, Abdallah ME, Radford NA, Bridgwater LB, Sanders AM, Askew RS, Linn DM, Yamokoski JD, Permenter FA, Hargrave BK, Platt R, Savely RT and Ambrose RO (2011) Robonaut 2—the first humanoid robot in space. In: *IEEE Int. Conf. Robot. Autom.* IEEE, pp. 2178–2183.

Dobson A and Bekris KE (2014) Sparse roadmap spanners for asymptotically near-optimal motion planning. *Int. J. of Robotics Research* 33(1): 18–47. DOI:10.1177/0278364913498292.

Elbanhawi M and Simic M (2014) Sampling-based robot motion planning: A review. *IEEE Access* 2: 56–77. DOI:10.1109/ACCESS.2014.2302442.

Fallon M, Kuindersma S, Karumanchi S, Antone M, Schneider T, Dai H, D'Arpino CP, Deits R, DiCicco M, Fourie D, Koolen T, Marion P, Posa M, Valenzuela A, Yu KT, Shah J, Iagnemma K, Tedrake R and Teller S (2015) An architecture for online affordance-based perception and whole-body planning. *J. of Field Robotics* 32(2): 229–254.

Fonseca R, Budday D and van den Bedem H (2018) Collision-free poisson motion planning in ultra high-dimensional molecular conformation spaces. *J. of Computational Chemistry* DOI:10.1002/jcc.25138.

Gammell J, Srinivasa SS and Barfoot TD (2015) Batch informed trees (BIT*): Sampling-based optimal planning via the heuristically guided search of implicit random geometric graphs. In: *IEEE Int. Conf. Robot. Autom.*

Garrett CR, Lozano-Pérez T and Kaelbling LP (2018) Sampling-based methods for factored task and motion planning. *Int. J. of Robotics Research* 37(13-14): 1796–1825. DOI:10.1177/0278364918802962.

Geraerts RJ and Overmars MH (2007) Creating high-quality paths for motion planning. *Int. J. of Robotics Research* 26(8): 845–863.

Gomes AJP, Voiculescu I, Jorge J, Wyvill B and Galbraith C (2009) *Implicit Curves and Surfaces: Mathematics, Data Structures and Algorithms*. Springer Science & Business Media.

Hairer E, Lubich C and Wanner G (2006) *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations*, *Springer Series in Computational Mathematics*, volume 31. 2 edition. Springer-Verlag Berlin Heidelberg.

Han L and Amato NM (2000) A kinematics-based probabilistic roadmap method for closed chain systems. In: *Int. Wksp. on the Algorithmic Foundations of Robotics*. Springer.

Han L, Rudolph L, Blumenthal J and Valodzin I (2008) Convexly stratified deformation spaces and efficient

path planning for planar closed chains with revolute joints. *Int. J. of Robotics Research* 27(11-12): 1189–1212.

Hauser K (2013) Fast interpolation and time-optimization on implicit contact submanifolds. In: *Robotics: Science and Syst.* DOI:10.15607/RSS.2013.IX.022.

Hauser K, Bretl T, Latombe JC, Harada K and Wilcox B (2008) Motion planning for legged robots on varied terrain. *Int. J. of Robotics Research* 27(11-12): 1325–1349. DOI:10.1177/0278364908098447.

Hauser K and Latombe JC (2010) Multi-modal motion planning in non-expansive spaces. *Int. J. of Robotics Research* 29(7): 897–915. DOI:10.1177/0278364909352098.

Hauser K and Ng-Thow-Hing V (2011) Randomized multi-modal motion planning for a humanoid robot manipulation task. *Int. J. of Robotics Research* 30(6): 678–698. DOI:10.1177/0278364910386985.

Henderson ME (2002) Multiple parameter continuation: Computing implicitly defined k-manifolds. *Int. J. of Bifurcation and Chaos* 12(3): 451–476.

Hsu D, Latombe JC and Motwani R (1999) Path planning in expansive configuration spaces. *Int. J. of Computational Geometry and Applications* 9(4n5): 495–512.

Jaillet L and Porta JM (2012) Asymptotically-optimal path planning on manifolds. In: *Robotics: Science and Syst.* DOI:10.15607/RSS.2012.VIII.019.

Jaillet L and Porta JM (2013a) Efficient asymptotically-optimal path planning on manifolds. *Robotics and Autonomous Syst.* 61(8): 797–807.

Jaillet L and Porta JM (2013b) Path planning under kinematic constraints by rapidly exploring manifolds. *IEEE Trans. Robot.* 29(1): 105–117.

James J, Weng Y, Hart S, Beeson P and Burridge R (2015) Prophetic goal-space planning for human-in-the-loop mobile manipulation. In: *IEEE-RAS Int. Conf. on Humanoid Robots*. pp. 1185–1192.

Janson L, Schmerling E, Clark A and Pavone M (2015) Fast marching tree: A fast marching sampling-based method for optimal motion planning in many dimensions. *Int. J. of Robotics Research* 34(7): 883–921.

Johnson M, Shrewsbury B, Bertrand S, Wu T, Duran D, Floyd M, Abeles P, Stephen D, Mertins N, Lesman A, Carff J, Rifenburgh W, Kaveti P, Straatman W, Smith J, Griffioen M, Layton B, de Boer T, Koolen T, Neuhaus P and Pratt J (2015) Team IHMC's lessons learned from the DARPA robotics challenge trials. *J. of Field Robotics* 32(2): 192–208.

Karaman S and Frazzoli E (2011) Sampling-based algorithms for optimal motion planning. *Int. J. of Robotics Research* 30(7): 846–894.

Kavraki LE and LaValle SM (2016) Motion planning. In: Siciliano B and Khatib O (eds.) *Springer Handbook of Robotics*, 2nd edition. Springer, Cham. DOI:10.1007/978-3-319-32552-1_7.

Kavraki LE, Švestka P, Latombe JC and Overmars M (1996) Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Trans. Robot. Autom.* 12(4): 566–580.

Khatib O (1987) A unified approach for motion and force control of robot manipulators: The operational space formulation 3(1): 43–53.

Kim B, Um TT, Suh C and Park FC (2016) Tangent bundle RRT: A randomized algorithm for constrained motion planning. *Robotica* 34(1): 202–225.

Kingston Z, Moll M and Kavraki LE (2017) Decoupling constraints from sampling-based planners. In: *Int. Symp. on Robotics Research*. Springer.

Kingston Z, Moll M and Kavraki LE (2018) Sampling-based methods for motion planning with constraints. *Annual Review of Control, Robotics, and Autonomous Systems* DOI:10.1146/annurev-control-060117-105226.

Kleinbort M, Solovey K, Littlefield Z, Bekris KE and Halperin D (2019) Probabilistic completeness of RRT for geometric and kinodynamic planning with forward propagation. *IEEE Robot. Autom. Letters* 4(2): 277–283.

Kuffner JJ and LaValle SM (2000) RRT-Connect: An efficient approach to single-query path planning. In: *IEEE Int. Conf. Robot. Autom.*, volume 2. pp. 995–1001.

Ladd AM and Kavraki LE (2004) Measure theoretic analysis of probabilistic path planning. *IEEE Trans. Robot. Autom.* 20(2): 229–242.

Ladd AM and Kavraki LE (2005) Motion planning in the presence of drift, underactuation and discrete system changes. In: *Robotics: Science and Syst.* pp. 233–240.

Latombe JC (1991) *Robot Motion Planning*. Kluwer Academic publishers. ISBN 978-0-7923-9206-4.

Laumond JP (1986) Feasible trajectories for mobile robots with kinematic and environment constraints. In: *Int. Conf. on Intelligent Autonomous Systems*. pp. 346–354.

LaValle SM (2006) *Planning Algorithms*. Cambridge University Press. ISBN 0521862051.

LaValle SM and Kuffner JJ (2001) Randomized kinodynamic planning. *Int. J. of Robotics Research* 20(5): 378–400.

Lee JM (2003) *Introduction to Smooth Manifolds*, *Graduate Texts in Mathematics*, volume 218. 1 edition. Springer-Verlag New York. ISBN 978-0-387-21752-9. DOI: 10.1007/978-0-387-21752-9.

Lozano-Pérez T (1983) Spatial planning: A configuration space approach. *IEEE Trans. on Computers* C-32(2):

108–120. DOI:10.1109/TC.1983.1676196.

Luenberger DG (1979) *Introduction to Dynamic Systems: Theory, Models, and Applications*. Wiley New York.

Luna R, Şucan IA, Moll M and Kavraki LE (2013) Anytime solution optimization for sampling-based motion planning. In: *IEEE Int. Conf. Robot. Autom.* Karlsruhe, Germany, pp. 5053–5059. DOI:10.1109/ICRA.2013.6631301.

Luo J and Hauser K (2014) An empirical study of optimal motion planning. In: *IEEE/RSJ Int. Conf. on Intell. Robots and Syst.*

Mason MT (1981) Compliance and force control for computer controlled manipulators. *IEEE Trans. on Syst., Man, and Cybernetics* 11(6): 418–432. DOI:10.1109/TSMC.1981.4308708.

McMahon T (2016) *Sampling Based Motion Planning with Reachable volumes*. PhD Thesis, Texas A&M University.

Mirabel J and Lamiraux F (2016) Manipulation planning: Building paths on constrained manifolds. URL https://hal.archives-ouvertes.fr/hal-01360409. Preprint.

Mirabel J and Lamiraux F (2018) Handling implicit and explicit constraints in manipulation planning. In: *Robotics: Science and Syst.* DOI:10.15607/RSS.2018.XIV.018.

Mirabel J, Tonneau S, Fernbach P, Seppälä AK, Campana M, Mansard N and Lamiraux F (2016) HPP: A new software for constrained motion planning. In: *IEEE/RSJ Int. Conf. on Intell. Robots and Syst.* pp. 383–389.

Mitchell JSB, Mount DM and Papdimitrious CH (1987) The discrete geodesic problem. *SIAM J. on Computing* 16(4): 647–668.

Nocedal J and Wright S (2006) *Numerical Optimization*. Springer-Verlag New York.

Oriolo G and Vendittelli M (2009) A control-based approach to task-constrained motion planning. In: *IEEE/RSJ Int. Conf. on Intell. Robots and Syst.* pp. 297–302.

Pachov DV and van den Bedem H (2015) Nullspace sampling with holonomic constraints reveals molecular mechanisms of protein Gαs. *PLoS Computational Biology* 11(7).

Porta JM, Ros L, Bohigas O, Manubens M, Rosales C and Jaillet L (2014) The CUIK suite: Analyzing the motion closed-chain multibody systems. *IEEE Robot. Autom. Magazine* 21(3): 105–114.

Press WH, Teukolsky SA, Vetterling WT and Flannery BP (2007) *Numerical Recipes: The Art of Scientific Computing*. 3rd edition. Cambridge University Press.

Rakita D, Mutlu B and Gleicher M (2018) RelaxedIK: Real-time synthesis of accurate and feasible robot arm motion. In: *Robotics: Science and Syst.* Pittsburgh, Pennsylvania. DOI:10.15607/RSS.2018.XIV.043.

Resnick SI (2014) *A Probability Path*. Birkhäuser Basel. DOI:10.1007/978-0-8176-8409-9.

Rheinboldt WC (1991) On the existence and uniqueness of solutions of nonlinear semi-implicit differential-algebraic equations. *Nonlinear Analysis: Theory, Methods & Applications* 16(7-8): 647–661.

Rheinboldt WC (1996) MANPAK: A set of algorithms for computations on implicitly defined manifolds. *Computers & Mathematics with Applications* 32(12): 15–28.

Sánchez G and Latombe JC (2003) A single-query bi-directional probabilistic roadmap planner with lazy collision checking. In: *Int. Symp. on Robotics Research*. Springer, pp. 403–417.

Schulman J, Duan Y, Ho J, Lee A, Awwal I, Bradlow H, Pan J, Patil S, Goldberg K and Abbeel P (2014) Motion planning with sequential convex optimization and convex collision checking. *Int. J. of Robotics Research* 33(9): 1251–1270.

Sentis L and Khatib O (2005) Synthesis of whole-body behaviors through hierarchical control of behavioral primitives. *Int. J. of Humanoid Robot.* 2(4): 505–518.

Siméon T, Laumond JC, Cortés J and Sahbani A (2004) Manipulation planning with probabilistic roadmaps. *Int. J. of Robotics Research* 32(7-8): 729–746.

Spivak M (1999) *A Comprehensive Introduction to Differential Geometry*. Publish or Perish.

Srivastava S, Fang E, Riano L, Chitnis R, Russell S and Abbeel P (2014) Combined task and motion planning through an extensible planner-independent interface layer. In: *IEEE Int. Conf. Robot. Autom.* pp. 639–646. DOI:10.1109/ICRA.2014.6906922.

Stewart G (1969) On the continuity of the generalized inverse. *SIAM J. Appl. Math.* 17(1): 33–45. DOI:10.1137/0117004.

Stilman M (2010) Global manipulation planning in robot joint space with task constraints. *IEEE Trans. Robot.* 26(3): 576–584.

Şucan IA and Chitta S (2011) *MoveIt!* URL http://moveit.ros.org.

Şucan IA and Chitta S (2012) Motion planning with constraints using configuration space approximations. In: *IEEE/RSJ Int. Conf. on Intell. Robots and Syst.* IEEE, pp. 1904–1910.

Şucan IA and Kavraki LE (2008) Kinodynamic motion planning by interior-exterior cell exploration. In: *Int. Wksp. on the Algorithmic Foundations of Robotics*. Springer.

Şucan IA and Kavraki LE (2011) Mobile manipulation: Encoding motion planning options using task motion multigraphs. In: *IEEE Int. Conf. Robot. Autom.* pp.

5492–5498. DOI:10.1109/ICRA.2011.5980212.

Şucan IA, Moll M and Kavraki LE (2012) The Open Motion Planning Library. *IEEE Robot. Autom. Magazine* 19(4): 72–82.

Tenenbaum JB, de Silva V and Langford JC (2000) A global geometric framework for nonlinear dimensionality reduction. *Science* 290(5500): 2319–2323. DOI:10.1126/science.290.5500.2319.

Tsai LW (1999) *Robot Analysis: The Mechanics of Serial and Parallel Manipulators*. John Wiley & Sons.

Vega-Brown W and Roy N (2016) Asymptotically optimal planning under piecewise-analytic constraints. In: *Int. Wksp. on the Algorithmic Foundations of Robotics*. Springer.

Vendittelli M and Oriolo G (2009) Task-constrained motion planning for underactuated robots. In: *IEEE Int. Conf. Robot. Autom.* pp. 2965–2970.

Voss C, Moll M and Kavraki LE (2017) Atlas + X: Sampling-based planners on constraint manifolds. Technical Report 17-02, Department of Computer Science, Rice University, Houston, TX.

Wedemeyer WJ and Scheraga HA (1999) Exact analytical loop closure in proteins using polynomial equations. *J. of Computational Chemistry* 20(8): 819–844.

Weghe MV, Ferguson D and Srinivasa SS (2007) Randomized path planning for redundant manipulators without inverse kinematics. In: *IEEE-RAS Int. Conf. on Humanoid Robots*. pp. 477–482.

Yakey JH, LaValle SM and Kavraki LE (2001) Randomized path planning for linkages with closed kinematic chains. *IEEE Trans. Robot. Autom.* 17(6): 951–958.

Yao Z and Gupta K (2005) Path planning with general end-effector constraints: Using task space to guide configuration space search. In: *IEEE/RSJ Int. Conf. on Intell. Robots and Syst.* pp. 1875–1880.

Zha F, Liu Y, Guo W, Wang P, Li M, Wang X and Li J (2018) Learning the metric of task constraint manifolds for constrained motion planning. *Electronics* 7(12): 395.

Zhang Y and Hauser K (2013) Unbiased, scalable sampling of protein loop conformations from probabilistic priors. *BMC Structural Biology* 13(1): S9.

Zucker M, Ratliff N, Dragan AD, Pivtoraiko M, Klingensmith M, Dellin CM, Bagnell JA and Srinivasa SS (2013) CHOMP: Covariant hamiltonian optimization for motion planning. *Int. J. of Robotics Research* 32(9-10): 1164–1193. DOI:10.1177/0278364913488805.