Appeared in Int. Symp. Robotics Research 2017

Decoupling Constraints from Sampling-Based Planners

Zachary Kingston, Mark Moll, and Lydia E. Kavraki

Abstract We present a general unifying framework for sampling-based motion planning under kinematic task constraints which enables a broad class of planners to compute plans that satisfy a given constraint function that encodes, e.g., loop closure, balance, and end-effector constraints. The framework decouples a planner's method for exploration from constraint satisfaction by representing the implicit configuration space defined by a constraint function. We emulate three constraint satisfaction methodologies from the literature, and demonstrate the framework with a range of planners utilizing these constraint methodologies. Our results show that the appropriate choice of constrained satisfaction methodology depends on many factors, e.g., the dimension of the configuration space and implicit constraint manifold, and number of obstacles. Furthermore, we show that novel combinations of planners and constraint satisfaction methodologies can be more effective than previous approaches. The framework is also easily extended for novel planners and constraint spaces.

Key words: Sampling-Based Motion Planning, Constrained Motion Planning

1 Introduction

Motion planning is an essential tool for a robotic system with any level of autonomy. With planning, a robot's movements can be specified with start and goal configurations, rather than a full prescription of intermediate states [7]. *Task constraints* are an important mechanism to concisely specify complex motions for a robot. For example, a robot tasked with transferring a glass of water may have to be constrained to keep the glass level. Another example of common task constraints are loop-closure constraints, such as in parallel manipulators or a bi-manual system carrying a tray [25] (shown in Figure 1). Recently, there has been rapid development in creating robotic

Zachary Kingston, Mark Moll, and Lydia E. Kavraki
Rice University, 6100 Main St., Houston TX 77005, e-mail: {zak, mmoll, kavraki}@rice.edu

Appeared in Int. Symp. Robotics Research 2017

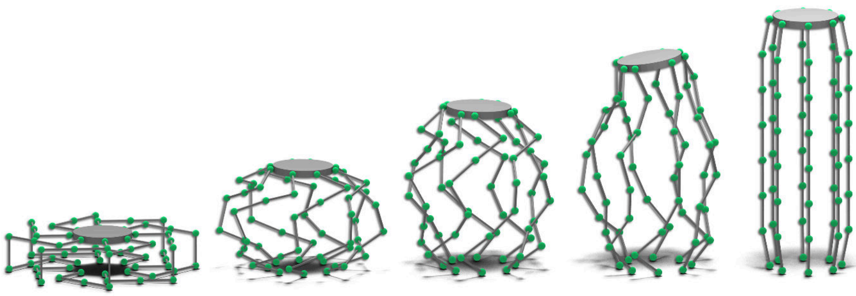

Fig. 1: An plan in the "implicit parallel manipulator" environment. The goal is to move from a flat, rotated configuration to upright. This path was computed using KPIECE in a projection-based constrained space in a median time of 14.5 seconds.

systems that are high-dimensional such as humanoid robots, mobile manipulators, and redundant arms, but planning for these high-dimensional systems is hard due to the inherent difficulty of the motion planning problem [25]. Planning for these systems with constraints is even more important and relevant as complex systems are tasked with more complex objectives.

In general, sampling-based planners have been effective at planning motions for high-dimensional systems [7]. 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 constraints in planning is still difficult, as finding configurations that satisfy constraints is a challenging task. Recently, several algorithms have been developed for planning with constraints that are effective for realistic problems [2, 18, 23]. These algorithms are somewhat limited in the sense that they adapt a specific sampling-based algorithm to also satisfy task constraints, convolving constraint satisfaction with planning methodology.

This paper presents a solution to the design of constrained sampling-based algorithms for more complex systems by means of a framework that decouples constraint satisfaction from space exploration in the planner. With this framework, a broad class of sampling-based planners can utilize many previously proposed constraint satisfaction methods and leverage the tools developed by the community, such as asymptotically optimal planners [20], path optimization [12], or domain specific planners for high-dimensional problems [37]. The conceptual framework encapsulates and extends previous approaches in the literature. We show that within our framework different constraint satisfaction methodologies can all use the same underlying constraint representation. Furthermore, we show that different problems can be solved more successfully using novel combinations of planning algorithms and implicitly defined constrained spaces.

This paper is organized as follows. Section 2 contains a survey of related work for constrained motion planning. Section 3 defines constraints and the constrained planning problem. Our framework which decouples constraints from sampling-based algorithms is presented in Section 4. Empirical results are shown and discussed in Section 5. Section 6 contains concluding remarks and directions for future work.

2 Related Work

In this work we are studying sampling-based planning with geometric task constraints [25], which has a wide breadth of literature concerning both techniques to plan motions and represent constraints. Using task constraints to specify the motion of a robotic system is not a new problem, and has its roots within industrial control [27, 22]. Task constraints on robot motion can be used to specify many useful manipulation tasks [34], parallel manipulators and closed chains [41, 25] and even structural biology [46]. Most early work with task constraints did not focus on geometric constraints, and was directed at non-holonomic constraints, such as differential drive cars [1]. However, as planners were applied to more complex, high-dimensional systems, geometric constraints were revisited as a difficult addition to the motion planning problem.

Non-Sampling Methods While not the focus of this paper, a short survey of non-sampling-based methods for planning with constraints is given for completeness.

One approach is to plan in the robot's workspace, so geometric constraints can be directly evaluated and satisfying poses can be sampled. Post-planning a path in the robot's configuration space is generated using inverse kinematics (IK) [33, 19]. However, these methods may not be efficient as re-planning is required if a found path cannot be mapped into the configuration space of the robot. Completeness is also not guaranteed unless all feasible IK solutions can be generated given the constraints.

Another approach that operates within the robot's workspace is reactive control, which uses convex optimization to find local satisfying motions, such as those used at the DARPA Robotics Challenge (e.g., [10]). While effective with operator supervision, these controllers are usually incomplete and risk local minima. As local controllers are optimization-based methods, hard constraints are relaxed into soft constraints, and invalid motions can be generated. Trajectory optimization approaches (e.g., [47, 32]) optimize within trajectory space and are effective for everyday manipulation tasks, but suffer from many of the same shortfalls as reactive control. Comprehensive comparison of constrained non-sampling-based methods to sampling-based planners has not been done, and a thorough analysis is left as future work.

Sampling-Based Planning Sampling-based planners fall broadly into two categories: graph-based methods such as PRM [21] and tree-based methods such as RRT [26]. Graph-based methods construct a "roadmap" within the configuration space that can be queried multiple times.Tree-based methods build a tree of motions rooted from the start or goal. Many techniques perform a bidirectional search for efficiency (e.g., [24]) or use coverage estimates to bias search towards unexplored space (e.g., [37]). While sampling-based planners have been shown to be very efficient in finding feasible paths, paths are often far from optimal. One approach to improve path quality is to post-process and locally optimize paths [12]. Sampling-based algorithms can also provide asymptotic optimality guarantees [20] such that the solution path converges to a globally optimal path for a cost function, or “soft” constraint.

The first methods capable of solving constrained problems dealt with specialized cases of constraints, specifically loop closures in parallel manipulators. Planning with loop closures is very relevant in structural biology [43], and complex loop closure problems in robotics were solved with PRM variants using active/passive chain methods [13, 44, 8]. Key to loop closure methods is a *projection operator*, which maps an unsatisfying configuration into a satisfying one. Active/passive chain methods use IK as a projection operator to join the passive chain to the active chain, closing the loop and creating a satisfying configuration. Cyclic-Coordinate Descent [5] is another loop closure method that, unlike numerical IK, does not require the computation of Jacobian (pseudo-)inverses.

The idea of projection to satisfy constraints was applied to general end-effector constraints in [45]. Task Constrained RRT [36] further generalized the idea of constraints and utilized Jacobian gradient descent [4] for projection. More recently, CBIRRT2 [2] and the motion planner implemented for the Humanoid Path Planner System [29] utilize projection with general constraints and can solve complex combinations of constraints. Additionally, the projection methodology has been extended to handle “soft” constraints with GradienT-RRT [2]. We show our framework can emulate CBIRRT2 and other previous approaches in Section 4.2.

Projection, while effective at satisfying constraints, utilizes very little information from the constraint. A constraint function defines an *implicit manifold* within the robot’s configuration space composed of all constraint satisfying configurations. This manifold is typically of lower dimension than the configuration space. The projection operator described above projects a point from the ambient space to this manifold. As the constraint defines a manifold, it is possible to locally approximate the manifold using a *tangent space* of a satisfying configuration. The tangent space can be used to generate new configurations that are close to the manifold. 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. [36, 6, 30] use local tangent space approximations to generate new samples.

Furthering the idea of local parameterization with tangent spaces is the concept of building an *atlas* of the manifold, a concept borrowed from the definition of differentiable manifolds [35]. Here, the atlas is defined as a piece-wise linear approximation of the manifold using tangent spaces, which fully cover and approximate the manifold [15]. TB-RRT [23] and AtlasRRT [18] both construct an atlas, incrementally building a 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 asymptotically optimal (AtlasRRT* [17]) and kinodynamic planning [3]. Like CBIRRT2, both TB-RRT and AtlasRRT are emulated within our framework, as shown in Section 4.3.

Carrying the concept of approximation even farther, recent works delve into complete parameterizations of the manifold. The constraint function is used to build a representation that only contains configurations that satisfy or very nearly satisfy the constraint. “Deformation space” [14] and “Reachable volume space” [28] are both reparameterization-based approaches.

The techniques discussed above cover a spectrum of methods to compute satisfying configurations for constrained motion planning. The spectrum describes the amount of effort the constraint methodology is using to more closely plan using the true implicit manifold. On one end of the spectrum are projection-based methods, which use little information about the constraint. On the other end lie approaches such as atlas-based methods, which compute considerable information about the constraint in order to approximate the implicit manifold.

3 Constrained Sampling-Based Planning

A *configuration* of the robot is denoted by $q \in \mathcal{Q}$, where $\mathcal{Q}$ is the *configuration space*, a metric space. The number of degrees of freedom of a robot, i.e., the dimensionality of its configuration space, is denoted by n. The motion planning problem is defined as finding a continuous, collision-free path from q_{start} to q_{goal} in configuration space $\tau : [0,1] \to \mathcal{Q}$, $\tau(0) = q_{start}$, $\tau(1) = q_{goal}$. In many cases, avoiding collisions is the only concern for computing a valid path. For constrained motion planning, we also want to satisfy a *constraint function* $F(q) : \mathcal{Q} \to \mathbb{R}^k$ over the configuration space, which evaluates $F(q) = 0$ when q satisfies the constraint. Here, k is the number of equality constraints imposed. The constraint function defines an $(n-k)$-dimensional implicit constrained configuration space within the ambient configuration space:

$$\mathcal{X} = \{\, q \in \mathcal{Q} \mid F(q) = 0 \,\}$$

For this work, we assume F is continuous and differentiable everywhere, and therefore $\mathcal{X}$ is a manifold. This assumption is stronger than strictly necessary for much of this work, but is imposed for ease of presentation. The constrained motion planning problem, with a constraint function F and configuration space $\mathcal{Q}$, is a problem of finding $\tau : [0,1] \to \mathcal{X}$. For example, consider a point robot with a configuration space $\mathcal{Q} \subset \mathbb{R}^3$. Given $F(q) = \|q\| - 1$, the robot is constrained to the surface of a unit sphere, a two-dimensional manifold in $\mathbb{R}^3$. Operations on $\mathcal{X}$ are described in Sections 4.2 and 4.3, which require additional definitions to be understood.

A *projection operator* is a continuous idempotent mapping $P(q) : \mathcal{Q} \to \mathcal{X}$, where if $q \in \mathcal{X}, P(q) = q$. Projection takes a configuration and *projects* it onto the surface of the implicit manifold, solving for a root of the constraint function. Typically this is implemented using Jacobian gradient descent, using the Jacobian of the constraint function $J(q)$. The descent stops when $F(q) = 0$. A more comprehensive look at projection for constrained motion planning is found in [2].

Local parameterization of the implicit manifold can be accomplished by a *tangent space* (alternatively, a *chart* from [18]). The tangent space is constructed by finding the basis for the nullspace of $J(q)$, which can be computed through a matrix decomposition. The tangent space T_q is a $(n-k)$-dimensional space with its origin at a configuration $q \in \mathcal{X}$, with an $n \times (n-k)$ orthonormal basis Φ_q. A point $t \in T_q$ can be mapped into $q_t \in \mathcal{Q}$ by $q_t = q + \Phi_q t$. To map the configuration q_t onto the manifold (an exponential map), an orthonormal projection can be computed by solving the system of equations:

$$F(q) = 0 \qquad \text{and} \qquad \Phi_q^T (q - q_t) = 0$$

The opposite mapping from the manifold to T_q is much simpler: $t = \Phi_q^T (q - q_t)$. Tangent spaces are composed into an approximation of the manifold by the AtlasRRT and TB-RRT constrained spaces within the framework, and are discussed in Section 4.3. A more comprehensive look at manifold approximation for planning can be found in [18], with many operations on implicit manifold discussed in [31]. The concepts of differential manifolds outlined here are covered in depth in [35].

4 Representing Implicit Spaces

Despite their differences, sampling-based planners have similar requirements from the robot's configuration space [25]. The primary capabilities we are concerned with are the following:

- Computation of distance between states, to select nearby states in the motion graph to either extend from or connect to.
- "Projection" for estimating configuration space coverage in relation to a task, so that the planner can measure progress and sampling can be directed towards uncovered regions (Note this is not a projection operator as described before).
- Linear interpolation on a geodesic, or moving between two states, so that new states can be created or validated through extension or connection.
- Sampling "uniformly" over the space or nearby known states to generate new configurations, which can be grown towards or connected to the motion graph.

These can be defined as operations on the space itself and need not be specific to any planner. The contribution of this paper is a conceptual framework, outlined within Section 4.1, that enables a broad class of motion planners to plan in many constrained spaces by exploiting the commonality of the spaces' primitive operations. This decouples constraints from a planner by augmenting the space with primitives that automatically satisfy imposed constraints.

This section is organized as follows. First we discuss the framework at an abstract level in Section 4.1 and describe how each of the space primitives utilized by a sampling-based planner are conceptualized. Then, we show our emulations of three

Appeared in Int. Symp. Robotics Research 2017

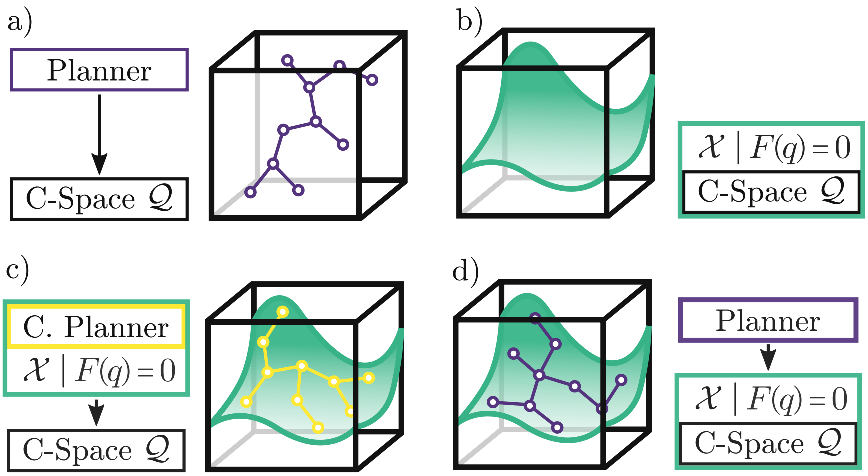

Fig. 2: A depiction of the framework and its relation to sampling-based planners. **a)** A box configuration space $\mathcal{Q}$ is shown in black. A sampling-based planner (purple) plans in $\mathcal{Q}$ using primitives afforded by the space. **b)** A constraint function $F(q) = 0$ defines a implicit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

successful and widely known methodologies within our framework: CBIRRT2 [2] in Section 4.2, and Tangent Bundle RRT [23] and AtlasRRT [18] in Section 4.3.

4.1 Conceptual Framework

A sampling-based planning algorithm plans within a configuration space $\mathcal{Q}$, and generates a collision-free path by using a validity checker along with properties of the configuration space, shown in Figure 2a. Prior works augmented the planning algorithm with a means to find constraint satisfying motions, shown in Figure 2c. In contrast, our framework is a layer of abstraction that lies between the representation of the robot's configuration space and the sampling-based planner used to find valid motions, shown in Figure 2d. The framework can be thought of as a representation of the implicit manifold $\mathcal{X}$ defined by the constraint function F, and a means for a sampling-based planner to plan within this space.

Normally, the distance metric utilized by a sampling-based planner is defined by the configuration space. This metric is primarily for nearest-neighbor computation, by which states nearby novel states can be found (e.g., `Select` and `SelectNghbrs` in Figure 3). For example, a point robot in $\mathbb{R}^3$ and a manipulator arm with $\mathcal{Q} \subseteq \mathbb{R}^n$

Appeared in Int. Symp. Robotics Research 2017

procedure TREE PLANNER(q_{start}, q_{goal})
  $\mathcal{T}$.`init`(q_{start});
  while no path from q_{start} to q_{goal} **do**
    $q_{rand} \leftarrow$ `Sample`();
    $q_{near} \leftarrow$ `Select`($\mathcal{T}$, q_{rand})
    $q_{new} \leftarrow$ `Extend`(q_{near}, q_{rand});
    if `Connect`(q_{new}, q_{near}) **then**
      $\mathcal{T}$.`Add`(q_{near}, q_{new})

procedure GRAPH PLANNER(q_{start}, q_{goal})
  $\mathcal{G}$.`init`(q_{start}, q_{goal});
  while no path from q_{start} to q_{goal} **do**
    $q_{rand} \leftarrow$ `Sample`();
    $Q \leftarrow$ `SelectNghbrs`($\mathcal{G}$, q_{rand})
    for all $q_{near} \in Q$ **do**
      if `Connect`(q_{near}, q_{rand}) **then**
        $\mathcal{G}$.`Add`(q_{near}, q_{rand})

Fig. 3: Prototypical examples of tree- and graph-based sampling-based planners. Many sampling-based planners can be cast into this mold, such as RRT for the tree-based planner, and PRM for the graph-based planner. This illustrates the underlying similarities in sampling-based planners, as they use the same primitive operations. Note that both `Connect` and `Extend` interpolate a geodesic.

commonly use the Euclidean norm. However, the notion of distance in the ambient space has little meaning on the implicit manifold, as the manifold can twist and curve relative to the ambient space [40]. As the framework represents the implicit manifold of a constraint, a natural metric to use would be the Riemannian metric, or the length of the geodesic between points. However, this is computationally infeasible, as nearest-neighbor computations would require many geodesic computations. As such, the metric from the configuration space is used unless otherwise specified, defining a semi-metric on the manifold, as the triangle inequality may not hold given sufficient curvature. This is still "good enough" for most motion planning algorithms in practice, but some theoretical guarantees may not hold, such as asymptotic optimality.

Similarly, "projection" for coverage estimates is left unaffected by the framework. As they are heuristics to bias sampling, projections are problem specific and are typically hard to devise. Random linear projections perform well in many cases, but do not incorporate constraint information [39]. Interesting future work would be to utilize information about the implicit manifold as a projection for coverage estimates.

Computing geodesics from configuration q_a to q_b normally has an analytic form, such as linear interpolation in $\mathbb{R}^n$. In sampling-based planners, geodesic movement underlies `Extend` and `Connect`, as shown in Figure 3. For implicit manifolds, traversing geodesics is one of the biggest hurdles to cross. Traversing a geodesic in configuration space and attempting to "fix-up" the new configuration ignores the manifold's curvature and can generate invalid motions. Thus, geodesic interpolation within the framework is akin to a local motion planner, computing a discretized geodesic by growing from one state to another, taking small enough steps to accurately traverse the manifold's curvature. The way this traversal is accomplished is up to the instantiation of the methodology behind the framework, and is one of the defining traits of a constrained space. Figure 4 shows three local planning methodologies to compute discretized geodesics used by the three spaces in the framework. Once the discretized geodesic is computed, an interpolated state can be computed along the found geodesic, by doing piece-wise interpolation.

Critical to sampling-based planners is the ability to sample new configurations in the configuration space (`Sample` in Figure 3). This is normally as simple as drawing

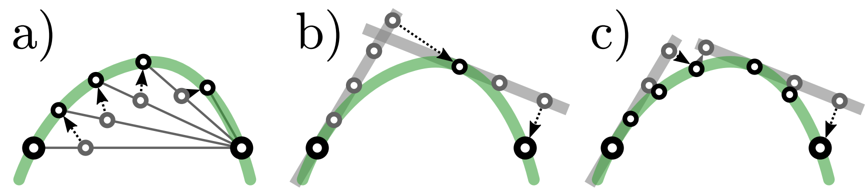

Fig. 4: Projection-, tangent bundle-, and atlas-based geodesic interpolation. Between points (large black) on the implicit manifold (green), the discretized geodesic is computed (black). **a)** Projection-based (CBIRRT2). Small extensions are taken (grey) and projected using a projection operator (arrow). **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.

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 instantiation of the framework will almost-surely sample any volume of non-zero measure within the manifold.

In summary, the key idea of our framework is to imbue the implicit constraint manifold with primitives that closely approximate those that exist for regular configuration spaces. This allows any sampling-based planner to plan with constraints without any special consideration. The next two sections describe two approaches to sampling and interpolation that are on opposite ends of the spectrum in terms of amount of information they maintain about the constraint manifold.

4.2 Projection-Based Space

One of the simplest methods to sample the constraint manifold is to sample from the configuration space and use the projection operator to retract samples onto the manifold. It was shown in [2] that sampling with projection will eventually cover the manifold, albeit with no guarantees on uniformity. Interpolation on the manifold using projection is achieved using a method similar to the extension method of the CBIRRT2 algorithm [2] (shown in Figure 4a). We emulate the projection-based space within our framework using the aforementioned methodology. We conjecture the framework retains the probabilistic completeness of the overlying planners, following the proof of probabilistic completeness of projection-based RRT-like planners in [2].

4.3 Tangent Space-Based Spaces

As discussed in Section 3, an implicit manifold $\mathcal{X}$ can be approximated by a set of tangent spaces. A few recent planners use tangent space approximations for efficient sampling nearby the manifold, such as TB-RRT [23] and AtlasRRT [18]. The planners both sample new points by sampling within tangent spaces and projecting these points onto the manifold. Although at first biased towards explored areas, in the limit once the manifold has been fully explored sampling can approach uniform sampling [18]. These methods can sample within hard to project areas, such as the interior surface of a highly curved manifold. Geodesic interpolation is accomplished by walking along the tangent spaces of the approximation, switching tangent spaces once certain criteria are met. TB-RRT takes a lazy approach to interpolation, projecting to the manifold only when necessary to switch tangent spaces (shown in Figure 4b). 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 miss obstacles. AtlasRRT projects at every step along the approximation, and generates separating half-spaces to create polytopes of the tangent space for more accurate sampling and interpolation, at the cost of additional computation (shown in Figure 4c). Both of these methods are emulated within our framework.

5 Empirical Results

The framework was implemented within the Open Motion Planning Library [38] (OMPL), which has implementations of many popular sampling-based planning algorithms. Our framework fits neatly within OMPL's notion of a *state space*, and no modification was necessary to the implementation of any of the planning algorithms for them to work with the constrained planning framework. Moreover, 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 planning system as well as illustrate concepts that help put the work in context.

Within the literature of constrained motion planning, most planners are adaptations of sampling-based planners augmented with a constraint methodology. CBIRRT2 [2], TB-RRT [23], and AtlasRRT [18], the planners that we have chosen to emulate within the framework, all are augmentations of RRT-Connect [24]. Figure 5 shows the "sphere" environment, a two-dimensional manifold embedded within $\mathbb{R}^3$, defined by the constraint function $F(q) = \|q\| - 1$. The planner must traverse three longitudinal obstacles each with a narrow passage to move from the south to the north pole.

Appeared in Int. Symp. Robotics Research 2017

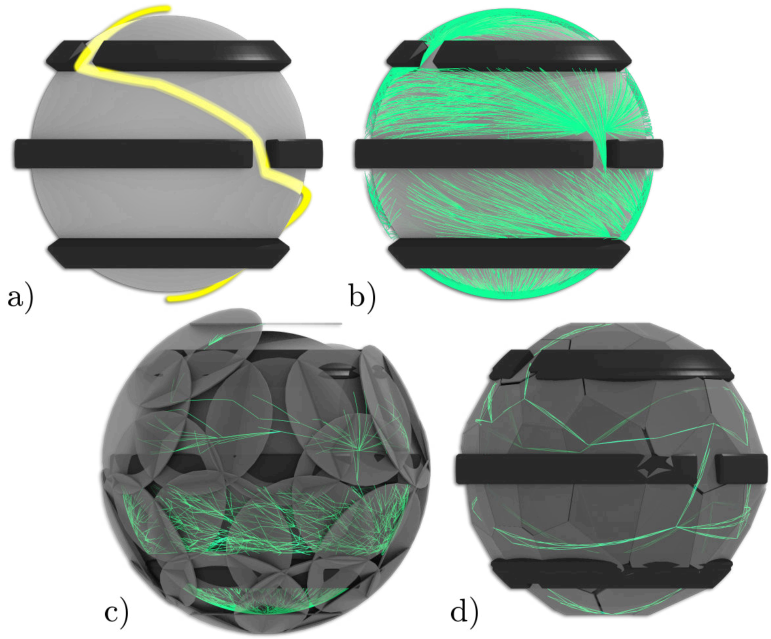

Fig. 5: The "sphere" environment. **a)** The sphere constraint manifold (grey) with obstacles (black). The solution path (yellow) runs from the south to north pole. **b)** Projection-based RRT* [20] motion graph (green) (Section 4.2). **c)** Tangent bundle-based BIT* [11] motion graph and tangent spaces (grey) (Section 4.3). **d)** Atlas-based SPARS [9] motion graph and tangent polytopes (Section 4.3).

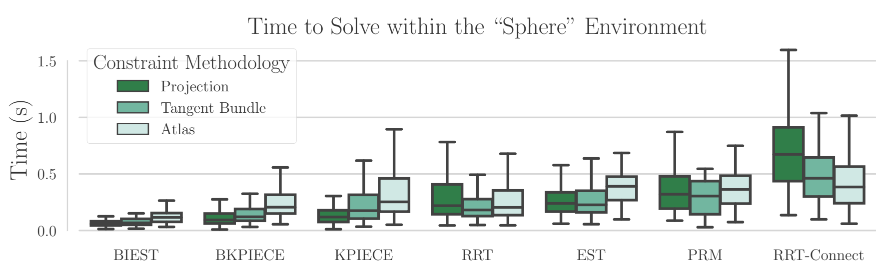

Fig. 6: Timing results from 100 runs of each planner in the "sphere" environment (Figure 5) using the three constrained spaces in the framework. Planners tested are EST, KPIECE, their bidirectional variants BIEST and BKPIECE [16, 37], RRT [26] and RRT-Connect [24], and PRM [21]. CBIRRT2, TB-RRT, and AtlasRRT are emulated by RRT-Connect in their respective constrained space, and perform the worst overall.

Appeared in Int. Symp. Robotics Research 2017

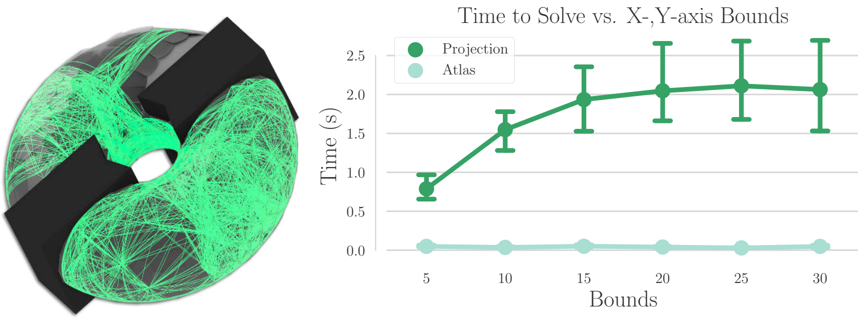

Fig. 7: The "torus" environment (grey) with obstacles (black) and timing results from 100 runs of PRM using the atlas- and projection-based constrained space versus the size of the x- and y- axes of the configuration space. On the left is a PRM motion graph (green) using the atlas-based space (tangent polytopes in grey). Projection-based PRM performs orders of magnitude worse than its atlas-based counterpart.

We show the results of 100 runs of various motion planners within the framework in Figure 6. As shown in the figure, combinations of planners and constrained spaces within the framework have dramatically different outcomes on planning time. Previous approaches in the literature are emulated by RRT-Connect within the framework, which is shown to have the poorest performance overall within the "sphere" environment. For this problem, any of the other tested planners would be a better selection of planner if speed was the primary concern.

More so, it is not just the planner that matters when approaching a constrained problem, the ambient configuration space can dramatically effect performance. Consider a "torus" environment (Figure 7), which is a two-dimensional manifold embedded within $\mathbb{R}^3$, with a constraint function $F(q) = (3 - \sqrt{x^2+y^2})^2 + z^2 - 2$. The planning problem is to traverse from one end of the torus to the other. There are obstacles bound around the outer surface of the torus, allowing passage only through the inner hole to traverse from one end to the other. Timing results for the PRM planner using the projection- and atlas-based methodologies is also shown in Figure 7, where the total volume of the configuration space is varied while the size of the torus remains constant. As shown by the results, 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 methodology, which samples directly off of 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 constraint methodology is equipped to handle

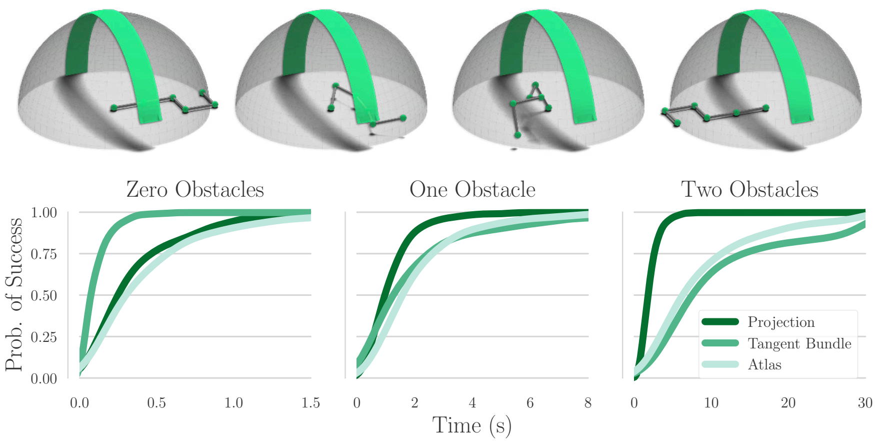

Fig. 8: A sample solution path with one obstacle is shown in the top figure. The bottom graphs show the cumulative probability of finding a path versus time for KPIECE using each constrained space with no surface obstacles, one obstacle, and two obstacles with antipodal narrow passages. 100 runs were used for each cumulative probability curve. Qualitatively similar results were obtained for RRT-Connect and PRM. Note that the X-axis on each plot is different.

every case. Therefore, the ability to combine and change constraint methodology with a planner is essential to efficiently planning within different environments.

A general trend observed by the authors is that as a planning problem becomes more constrained and the implicit manifold more curved with respect to the ambient space, atlas- and tangent-bundle-based methods perform better as 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, and projection-based methods do well. 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 8. Here, the kinematics are modeled as distance constraints, one for each link, on a chain with 5 spherical joints. The configuration space is thus $\mathbb{R}^{3\times 5}$. To further increase problem complexity, we impose the additional constraints: (a) 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 8. When there are no obstacles in this scene, tangent bundle-based methods perform the best, while projection- and atlas-based methods perform equally less. Lazy evaluation of states works in favor of this problem, as the planner can quickly traverse the constraint manifold. However, as obstacles are added to the surface of the outer sphere, tangent-bundle performs worse, as the projection and atlas methods improve relative performance drastically.

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 1, we show the "implicit parallel manipulator" environment, a parallel manipulator defined with a set of the "implicit chains," defined analogously to the previous example. 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. Emulated prior works (with RRT-Connect) were unable to successfully solve this system given 10 minutes of planning time. Using the KPIECE planner designed for high-dimensional spaces, we can quickly solve (median 14.5 seconds over 100 runs) this problem while satisfying constraints.

There is little work in the literature on satisfying "soft" constraints in tandem with kinematic constraints. AtlasRRT* [17] and GradienT-RRT [2] both respect "soft" constraints, but require specialized implementation and integration with the constraint methodology to work. Within our framework, no additional overhead is necessary for asymptotically optimal planning, as shown in Figure 5, which shows motion graphs for three asymptotically optimal and near-optimal planners. Additionally, path smoothing, shortening, and interpolation algorithms work with no knowledge of constraints, as all operations are handled by the framework.

6 Conclusion

We have introduced a novel framework for constrained sampling-based planning that decouples constraint satisfaction from a motion planner's exploration of a configuration space. We have demonstrated the framework's capability by showing our emulations of the constraint satisfaction methodology employed by three constrained planners, CBIRRT2, TB-RRT, and AtlasRRT. Additionally, we have tested a broad range of sampling-based planners within the framework for a set of constrained problems and shown that each planner can operate within the framework's constraint spaces. The framework is easily extended to new planners, and new constraint spaces can be adapted to the framework as its concepts are general to constrained planning. 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. Future work for the framework is the implementation of other constraint spaces, such as local tangent space sampling, adapting the framework for kinodynamic planning with constraints, and addressing proofs of completeness in light of the framework. Finally, we are in the process of integrating the framework with real robotic platforms.

Acknowledgements We would like to thank Caleb Voss for his preliminary work [42]. ZK, MM, and LEK are supported by NSF IIS 1317849 and Rice University funds. ZK is also supported by a NASA Space Technology Research Fellowship 80NSSC17K0163.

References

1. Barraquand, J., Latombe, J.C.: Nonholonomic multibody mobile robots: Controllability and motion planning in the presence of obstacles. Algorithmica **10**(2), 121–155 (1993)
2. Berenson, D.: Constrained manipulation planning. Ph.D. thesis, CMU (2011)
3. Bordalba, R., Ros, L., Porta, J.M.: Kinodynamic planning on constraint manifolds. ArXiv e-prints (2017). URL http://arxiv.org/abs/1705.07637
4. Buss, S.R., Kim, J.S.: Selectively damped least squares for inverse kinematics. J. of Graphics, GPU, and Game Tools **10**(3), 37–49 (2005)
5. Canutescu, A.A., Dunbrack, R.L.: Cyclic coordinate descent: A robotics algorithm for protein loop closure. Protein Science **12**(5), 963–972 (2003)
6. Cefalo, M., Oriolo, G., Vendittelli, M.: Task-constrained motion planning with moving obstacles. In: IEEE/RSJ Int. Conf. on Intell. Robots and Syst., pp. 5758–5763 (2013)
7. Choset, H., Lynch, K., Hutchinson, S., et al.: Principles of Robot Motion: Theory, Algorithms, and Implementation. MIT Press (2005)
8. Cortés, J., Siméon, T., Laumond, J.P.: A random loop generator for planning the motions of closed kinematic chains using PRM methods. In: IEEE Int. Conf. Robot. Autom., vol. 2, pp. 2141–2146 (2002)
9. Dobson, A., Krontiris, A., Bekris, K.E.: Sparse roadmap spanners. In: Int. Wksp. on the Algorithmic Foundations of Robotics (2013)
10. Fallon, M., Kuindersma, S., Karumanchi, S., et al.: An architecture for online affordance-based perception and whole-body planning. J. of Field Robotics **32**(2), 229–254 (2015)
11. Gammell, J., Srinivasa, S.S., Barfoot, T.D.: Batch informed trees (BIT*): Sampling-based optimal planning via the heuristically guided search of implicit random geometric graphs. In: IEEE Int. Conf. Robot. Autom. (2015)
12. Geraerts, R.J., Overmars, M.H.: Creating high-quality paths for motion planning. Int. J. of Robotics Research **26**(8), 845–863 (2007)
13. Han, L., Amato, N.M.: A kinematics-based probabilistic roadmap method for closed chain systems. In: Int. Wksp. on the Algorithmic Foundations of Robotics (2000)
14. Han, L., Rudolph, L., Blumenthal, J., Valodzin, I.: 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 (2008)
15. Henderson, M.E.: Multiple parameter continuation: Computing implicitly defined k-manifolds. Int. J. of Bifurcation and Chaos **12**(3), 451–476 (2002)
16. Hsu, D., Latombe, J.C., Motwani, R.: Path planning in expansive configuration spaces. Int. J. of Computational Geometry and Applications **9**(4n5), 495–512 (1999)
17. Jaillet, L., Porta, J.M.: Efficient asymptotically-optimal path planning on manifolds. Robotics and Autonomous Syst. **61**(8), 797–807 (2013)
18. Jaillet, L., Porta, J.M.: Path planning under kinematic constraints by rapidly exploring manifolds. IEEE Trans. Robot. **29**(1), 105–117 (2013)
19. James, J., Weng, Y., Hart, S., et al.: Prophetic goal-space planning for human-in-the-loop mobile manipulation. In: IEEE-RAS Int. Conf. on Humanoid Robots, pp. 1185–1192 (2015)
20. Karaman, S., Frazzoli, E.: Sampling-based algorithms for optimal motion planning. Int. J. of Robotics Research **30**(7), 846–894 (2011)
21. Kavraki, L.E., Švestka, P., Latombe, J.C., Overmars, M.: Probabilistic roadmaps for path planning in high-dimensional configuration spaces. IEEE Trans. Robot. Autom. **12**(4), 566–580 (1996)
22. Khatib, O.: A unified approach for motion and force control of robot manipulators: The operational space formulation. IEEE J. Robot. Autom. **3**(1), 43–53 (1987)

23. Kim, B., Um, T.T., Suh, C., Park, F.C.: Tangent bundle RRT: A randomized algorithm for constrained motion planning. Robotica **34**(1), 202–225 (2016)
24. Kuffner, J.J., LaValle, S.M.: RRT-Connect: An efficient approach to single-query path planning. In: IEEE Int. Conf. Robot. Autom., vol. 2, pp. 995–1001 (2000)
25. LaValle, S.M.: Planning Algorithms. Cambridge University Press (2006)
26. LaValle, S.M., Kuffner, J.J.: Randomized kinodynamic planning. Int. J. of Robotics Research **20**(5), 378–400 (2001)
27. Mason, M.T.: Compliance and force control for computer controlled manipulators. IEEE Trans. on Syst., Man, and Cybernetics (1981)
28. McMahon, T.: Sampling based motion planning with reachable volumes. Ph.D. thesis, Texas A&M University (2016)
29. Mirabel, J., Tonneau, S., Fernbach, P., et al.: HPP: A new software for constrained motion planning. In: IEEE/RSJ Int. Conf. on Intell. Robots and Syst., pp. 383–389 (2016)
30. Pachov, D.V., van den Bedem, H.: Nullspace sampling with holonomic constraints reveals molecular mechanisms of protein Gαs. PLoS Computational Biology **11**(7) (2015)
31. Rheinboldt, W.C.: MANPAK: A set of algorithms for computations on implicitly defined manifolds. Computers & Mathematics with Applications **32**(12), 15 – 28 (1996)
32. Schulman, J., Duan, Y., Ho, J., et al.: Motion planning with sequential convex optimization and convex collision checking. Int. J. of Robotics Research **33**(9), 1251–1270 (2014)
33. Sentis, L., Khatib, O.: Synthesis of whole-body behaviors through hierarchical control of behavioral primitives. Int. J. of Humanoid Robotics **2**(4), 505–518 (2005)
34. Siméon, T., Laumond, J.C., Cortés, J., Sahbani, A.: Manipulation planning with probabilistic roadmaps. Int. J. of Robotics Research **32**(7-8), 729–746 (2004)
35. Spivak, M.: A Comprehensive Introduction to Differential Geometry. Publish or Perish (1999)
36. Stilman, M.: Global manipulation planning in robot joint space with task constraints. IEEE Trans. Robot. **26**(3), 576–584 (2010)
37. Şucan, I., Kavraki, L.E.: Kinodynamic motion planning by interior-exterior cell exploration. In: Int. Wksp. on the Algorithmic Foundations of Robotics (2008)
38. Şucan, I., Moll, M., Kavraki, L.E.: The Open Motion Planning Library. IEEE Robot. Autom. Magazine **19**(4), 72–82 (2012)
39. Şucan, I.A., Kavraki, L.E.: On the performance of random linear projections for sampling-based motion planning. In: IEEE/RSJ Int. Conf. on Intell. Robots and Syst. (2009)
40. Tenenbaum, J.B., de Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science (2000)
41. Tsai, L.W.: Robot Analysis: The Mechanics of Serial and Parallel Manipulators. John Wiley & Sons (1999)
42. Voss, C., Moll, M., Kavraki, L.E.: Atlas + X: Sampling-based planners on constraint manifolds. Tech. Rep. 17-02, Department of Computer Science, Rice University, Houston, TX (2017)
43. Wedemeyer, W.J., Scheraga, H.A.: Exact analytical loop closure in proteins using polynomial equations. J. of Computational Chemistry **20**(8), 819–844 (1999)
44. Yakey, J.H., LaValle, S.M., Kavraki, L.E.: Randomized path planning for linkages with closed kinematic chains. IEEE Trans. Robot. Autom. **17**(6), 951–958 (2001)
45. Yao, Z., Gupta, K.: 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 (2005)
46. Zhang, Y., Hauser, K.: Unbiased, scalable sampling of protein loop conformations from probabilistic priors. BMC Structural Biology **13**(1), S9 (2013)
47. Zucker, M., Ratliff, N., Dragan, A., et al.: CHOMP: Covariant hamiltonian optimization for motion planning. Int. J. of Robotics Research (2013)