Sampling-Based Motion Planning with Reachable Volumes: Theoretical Foundations

Troy McMahon, Shawna Thomas, Nancy M. Amato

***Abstract*— We introduce a new concept, reachable volumes, that denotes the set of points that the end effector of a chain or linkage can reach. We show that the reachable volume of a chain is equivalent to the Minkowski sum of the reachable volumes of its links, and give an efficient method for computing reachable volumes. We present a method for generating configurations using reachable volumes that is applicable to various types of robots including open and closed chain robots, tree-like robots, and complex robots including both loops and branches. We also describe how to apply constraints (both on end effectors and internal joints) using reachable volumes. Unlike previous methods, reachable volumes work for spherical and prismatic joints as well as planar joints.**

Visualizations of reachable volumes can allow an operator to see what positions the robot can reach and can guide robot design. We present visualizations of reachable volumes for representative robots including closed chains and graspers as well as for examples with joint and end effector constraints.

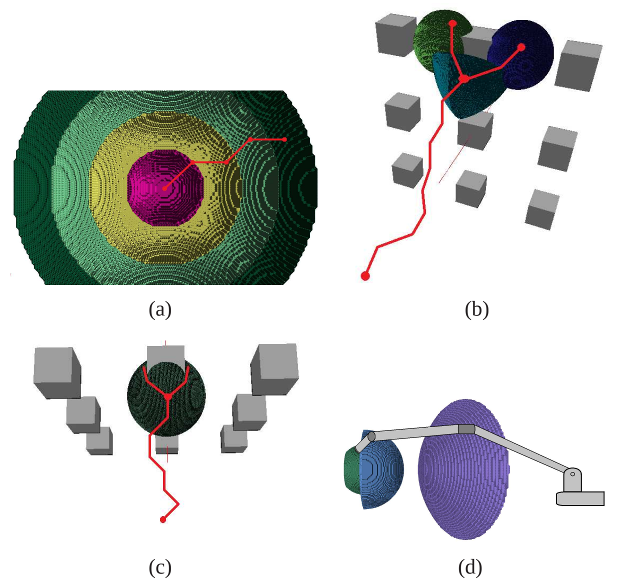


Fig. 1. The reachable volume (a) of a 4 link chain with spherical joints. The reachable volumes of a 16 dof fixed-base grasper with spherical joints is affected by constraints placed either (b) on the end effectors to be within spherical regions of the object or (c) on the base to be at a point. Note that in (c) the end effector reachable volumes are identical so only one is shown. (d) The reachable volume of a WAM robot grasping a spherical object. Example configurations shown in red (a,b,c) and gray (d).

I. Introduction

Constrained motion planning problems place constraints on the motion of an object (robot). These constraints could require that the robot remain in contact with a surface or that it maintain a specific clearance. They could also require that certain joints of the robot remain in contact with each other (e.g., closed chains). Motion planning with constraints is applicable to parallel robotics [20], grasping and manipulation [22], computational biology and molecular simulations [2], and animation [15].

Randomized motion planning methods such as the graph-based PRM method [16] and the tree-based RRT method [17] have had a good deal of success solving traditional motion planning problems. Unfortunately, these methods are poorly suited for constrained problems where the probability of randomly generating a sample satisfying the constraints approaches zero [18]. Previous methods have developed specialized samplers that generate samples that satisfy constraints [6, 11, 25]. Such samplers can be used in combination with existing PRM-based methods to solve problems with constraints, however these methods are either unable to handle high degree of freedom (dof) systems or unsuited for systems with spherical or prismatic joints or systems that combine different types of joints. One of the most effective previous methods is reachable distance sampling [25]. It produces samples more efficiently than random sampling for constrained systems, however it is limited to planar robots with 1D articulated joints and cannot be applied to prismatic or spherical joints or combinations of different joint types.

We propose the concept of *reachable volumes* that generalizes the concept of reachable distances so that it can be applied to linkages, closed chains and tree-like robots with prismatic and spherical as well as planar joints. Most previous methods assume a planar robot with 1D articulated joints and are not applicable to systems with spherical or prismatic joints. Reachable volumes also has potential application in robot control and operation because it allows an operator to see what a robot can reach, thereby letting the operator determine feasible actions or guiding robot design.

We define the reachable volume of a linkage to be the volume of space that the end effector of the linkage can reach while satisfying any constraints present. We present a method for efficiently computing reachable volumes and a family of samplers that uses them to generate constraint satisfying configurations of linkages, closed chains and tree-like robots. Unlike many previous methods (e.g., cyclic coordinate decent [28], reachable distances [25], and inverse kinematics [6]) which focus on end effector constraints, our method allows for constraints on internal joints as well as end effectors. Moreover, our method allows constraints to simultaneously be applied to multiple joints/end effectors, while most previous methods only allow for constraints on a single end effector. We show that the running time of these samplers is linear with respect to the size of the robot.

The main contributions of this work include:

- the concept of *reachable volume*, which denotes the volume of space that the joints and end effectors of a linkage can reach while satisfying the problem constraints, and
- a family of linear time sampling methods for robots with spherical, planar and prismatic joints that is applicable to tree-like linkages, closed chain systems, and combinations thereof.

This paper focuses on the theoretical foundations of reachable volumes. More detail, including a description of how reachable

This research supported in part by NSF awards CNS-0551685, CCF-0833199, CCF-0830753, IIS-0916053, IIS-0917266, EFRI-1240483, RI-1217991, by NIH NCI R25 CA090301-11, by Chevron, IBM, Intel, Oracle/Sun and by Award KUS-C1-016-04, made by King Abdullah University of Science and Technology (KAUST). [1]Parasol Lab., Dept. of Comp. Sci. and Eng., Texas A&M Univ., College Station, Texas, USA. {tmcmahon,sthomas,amato}@cse.tamu.edu

Method	**Constraint Types**	**High dof Robots**	**Closed Chains**	**Tree-like Robots**	**Joint Types**	**Multi-Query Problems**
Basic PRM [16]	None	Yes	No	Yes	Planar, Spherical, Prismatic	Yes
Inverse Kinematics [6]	End Effector	Yes	Yes	Yes	Planar	Yes
Constrained Dynamics [4]	End Effector	Yes	Yes	No	Planar	Yes
I-CD	None	Yes	No	Yes	Planar, Spherical, Prismatic	Yes
DDRRT [33, 32], Atlas RRT [14], Tangent Bundle RRT [24]	Only End Effector Shown	Not shown	Yes	Not shown	Only Planar Shown	No
CCD [28]	End Effector	No	Yes	No	Planar, Spherical	Yes
CHOMP [35]	*H*ard and *S*oft	Yes	Not Shown	Yes	Planar, Spherical, Prismatic, Combinations	No
Reachable Distances [25]	End Effector	Yes	Yes	Not shown	Planar, Prismatic	Yes
Reachable Volumes (this paper)	**End Effector, Internal Joints, Multiple Joints**	**Yes**	**Yes**	**Yes**	**Planar, Spherical, Prismatic, Combinations**	**Yes**

TABLE I

COMPARISON OF METHOD CAPABILITIES.

volumes can be computed efficiently and application to a range of systems including arbitrary linkages, closed chain systems, and mobile manipulators can be found in [19].

II. RELATED WORK

We give an overview of previous methods that are applicable to motion planning systems with constraints. A complete overview of related work can be found in [19].

Table I summarizes the capabilities of the various methods. Reachable volume sampling is unique in that it is shown to be applicable to problems with internal joint constraints and problems with constraints on multiple joints. As detailed below, most existing methods are limited to end effector constraints, and none have been explicitly shown to be applicable to such problems. Reachable volumes are also capable of handling high dof problems. While many existing methods are limited to problems with planar articulated joints, reachable volumes can handle prismatic and spherical joints and combinations of different types of joints. Reachable volumes also have an advantage over RRT-based methods in that they are applicable to multi-query problems.

A. Adaptations of the PRM and RRT methods

PRMs and RRTs have been adapted for use in spatially constrained systems. *Gradient decent methods* push randomly generated configurations onto a constrained surface [18, 30]. They are capable of solving problems with single-loop, articulated joint, closed chains. PRM-MC combines PRM and Monte Carlo methods to generate samples that satisfy closure constraints for single loop closed chains up to 100 links [8]. In [26, 21], Trinkle and Milgram develop a method that uses C-space analysis for path planning while ignoring self collisions. They show results for a set of planar parallel star-shaped manipulators. Alternative Task-space and Configuration-space Exploration (ATACE) for path planning with constrained manipulators uses a randomized gradient decent method for constrained manipulators [31]. They present results for a 9 dof manipulator robot with a set of end effector constraints. In [34] Zhang and Hauser present a Monte Carlo method for generating closed chain samples.

Atlas-RRT [14] and Tangent Bundle based RRT [24] construct RRTs along manifolds in the environment, while DDRRT [33, 32] can be used to construct RRTs along regions of space that satisfy a problem's constraints. They have been shown to solve constrained problems but are not applicable to more than 18 dof. Unfortunately, it would be difficult to adapt these methods to work in a PRM framework because they use an approximation of the constraint manifolds to generate new samples and these approximations are only accurate near existing samples.

1) Kinematics-based samplers: An alternative approach is to use inverse kinematics to produce constraint-satisfying samples. Kinematics-based PRM utilizes a two step process [6] that separates the difficulties of sampling constraint-satisfying configurations and collision-free configurations. Cortés *et. al.* developed a sampling method for closed chain linkages with kinematic constraints [2]. It is shown to be faster than previous kinematics-based sampling methods. Kinematics-based methods have been extended to large linkages [29] and multiple loops [2]. Inverse kinematic methods for 3D, 5D, and 6D end effector constraints are shown to efficiently generate samples for linkages with as many as 1000 links [11, 9].

It has been shown that for any planar polygonal loop there exist two special configurations such that any connectable pair of configurations can be connected by a sequence of straight line paths through them [7]. This method has been extended to produce paths guaranteed to be self-collision free [13]. They show that any two convex configurations of a closed chain can be connected by a path comprised of two straight line segments consisting only of convex configurations.

While inverse kinematics-based methods have had a great deal of success, they also have a number of major limitations. Most of these methods assume a planar robot with 1D planar joints. None of these methods can handle problems with prismatic joints or combinations of different joint types. In addition, these methods are only applicable to end effector constraints; they cannot handle problems with constraints on internal joints or constraints on multiple joints.

2) Optimization methods: Another approach is to iteratively optimize samples or paths until they satisfy a problem's constraints. Cyclic coordinate decent (CCD) [28] moves the end effector of a robot to a specified end effector position by iteratively cycling through the robot's coordinates and adjusting them so that the end effector converges to the goal position. CCD can also be used to generate closed chain samples or samples that satisfy a specified end effector constraint for linkages with as many as 7 dof.

CHOMP[35] uses gradient based techniques to improve paths by optimizing a function that balances obstacle avoidance and path smoothness. This method can be used to generate paths which satisfy *h*ard constraints and optimize adherence to *s*oft constraints.

3) Enforcing constraints during sampling: Some approaches explicitly enforce constraints while sampling. Closed chain problems may by transformed into a system of linear inequalities [10]. Constrained dynamics enforce constraints such as joint connectivity, spatial relationships, and obstacle avoidance for manipulators up to 6 dof [4]. Other planners require the end effector to traverse a predefined trajectory by generating samples that satisfy the end effector constraints given by the trajectory [22]. Other methods can generate samples with self-contact [12]. The *reachability grid* is a voxel-based representation that consists of a workspace grid in which each grid cell is denoted by the minimum time required to reach that cell [1]. It is possible to produce accurate reachability grids in real time; errors in estimates are almost always biased towards optimistic ones.

B. Reachable workspace and reachable distance

Reachable workspace [3] is the volume of workspace that can be reached by the center point of the end effector of a fixed base manipulator. It differs from reachable volumes in that it is only defined for serial linkages and it does not take into consideration a problem's constraints. Moreover, reachable workspace is only defined for end effectors so it cannot be used to generate samples in the same manner as reachable volumes.

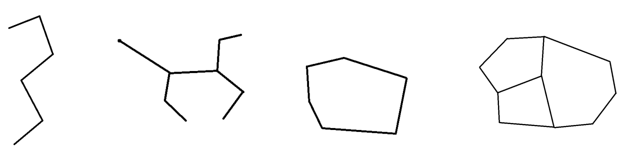
The reachable distance of an articulated linkage is the range of distances that its end effector can reach with respect to its base [25]. Reachable distance is computed by recursively computing the reachable distances of subsets of the linkage. This method efficiently produces samples for linkages, single and multiple loop closed chains, and constrained motion planning problems but is limited to planar joints.

III. Problem Formulation

We describe the types of linkages studied in this work and state the motion planning problem for linkages.

A. Robot types

We explore linkage systems with planar, spherical, and prismatic joints and combinations thereof. These systems consist of a set of links connected to each other by joints that can form a chain (Figure 2(a)), a tree (Figure 2(b)), or closed chains with one or many loops (Figure 2(c,d)).



(a) Chain (b) Tree-like robot (c) Closed chain (d) Multiple loop closed chain

Fig. 2. Examples of linkage systems.

Robot links are assumed to be rigid bodies connected at the ends by joints of various types. Planar joints are 1 dof articulated joints with a single parameter θ. Spherical joints are 2 dof joints in which any possible angle between adjacent links is valid denoted by an inclination θ and a rotation ϕ. Prismatic joints are 1 dof linear sliding joints represented by a single value d denoting the length it is extended.

B. Motion planning with linkages

The motion planning problem is to locate a valid path between a start and goal configuration. For linkage robots, paths consist of changes in the relative position of the links due to altering their dof. Sampling-based methods address this problem by sampling valid configurations and connecting them using a local planner. Constrained motion planning applies a set of constraints $\mathcal{S}$ to some or all of the joints. Configurations must satisfy $\mathcal{S}$ along with any other validity conditions associated with the problem.

C. Constraints

We define a constraint S_j to be a subset of space in which joint j must be located. In much of the previous work, constraints were assumed to be placed only on a linkage's end effectors. Our work is unique in that we allow constraints to be placed on any of the joints. Multiple constraints can be applied to the same joint by constructing a single constraint which is their intersection.

Joint position constraints can be used to model a wide variety of constrained systems. For example, one can require that a humanoid robot remain upright by constraining its torso to be above its base or require that an object such as a bucket of water remain upright by constraining the vertices of the object to be parallel to the robot's base. Grasping can be modeled by requiring that the end effectors of the robot be located at specified handle positions.

IV. Reachable Volumes

In this section we define the concept of reachable volumes for unconstrained systems. In Section IV-C we extend this definition to incorporate constraints.

A chain's reachable volume is the region of space that its end effector can reach. Below, we show that the reachable volume of a chain is equal to the Minkowski sum of the reachable volumes of the links in the chain. This allows us to develop a recursive method for computing the reachable volume of each of the robot's joints. We show how this approach applies not only to chain linkages, but also to more complex linkages such as trees and closed chains.

A. Reachable volume space

The *reachable volume space* (*RV-space*) for a linkage is a 3-dimensional space in which the origin is located at one of the joints or end effectors of the robot (referred to as the root). Points in RV-space represent possible locations of the joints and end effectors in the chain with respect to the root. RV-space is analogous to C-space in that it does not include any obstacles and that it can be used to generate configurations that are tested for validity using a validity checker (e.g. [5]).

The reachable volume of a chain is the set of points in RV-space that can be reached by the other end of the chain while satisfying any constraints associated with the problem (Figure 3(a)). First note that if you can position the end effector at one point that is r away from the origin, then you can reach all other points of distance r from the origin by rotating the the robot about the origin (Figure 3(b)). For chains with a single link of length l where the adjacent joint is not prismatic, the reachable set is the set of points that are a distance l from the origin. Thus, the reachable set can be represented by the radii $r_{min} = r_{max} = l$. If the link has an adjacent prismatic joint that ranges between d_{min} and d_{max}, then the reachable set is the set of points represented by the radii $r_{min} = l + d_{min}$ and $r_{max} = l + d_{max}$. Based on this, we observe the following:

Observation 1: If a point of distance r from the origin is reachable, then all points that are a distance of r from the origin must be reachable.

Proof: Because our definition of RV-space allows the base of a robot to rotate freely about the origin, this observation holds for chains that include planar, spherical, and prismatic joints. ■

Observation 2: If a chain can reach a point that is r_1 from the base and a point that is r_2 from the base, where $r_1 \leq r_2$, then it can reach points that are $r_1 \leq r \leq r_2$.

Based on these observations, there must exist a minimum radius (r_{min}) and maximum radius (r_{max}) such that the reachable volume of a chain is the set of points p whose distance from the origin O is between r_{min} and r_{max}.

$$\texttt{RV}(\texttt{Chain}) = \{p \mid r_{min} \leq \texttt{distance}(p, O) \leq r_{max}\}$$

Both observations hold for chains with planar, spherical, and prismatic joints, however they do not hold for chains with constraints. This is addressed below in Section IV-C.

An RV-space configuration consists of a position in RV-space for each of the joints and end effectors. A configuration in RV-space is composed of a position for each joint and end effector in the robot. In an RV-space configuration, the position of a joint or end effector in RV-space is equal to the difference between the position of that joint or end effector in workspace and the position of the root in workspace. An RV-space configuration captures the relative position

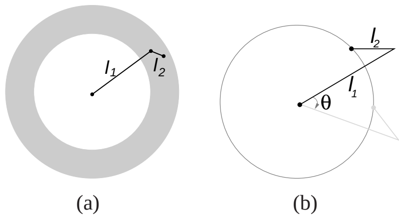


Fig. 3. (a) The reachable volume (gray region) of a 2 link chain robot, l_1 rotates about the point in the center while l_2 rotates about the endpoint of l_1. (b) If the end effector (black) can reach a point, then it can reach all other points that are the same distance from the base (gray circle).

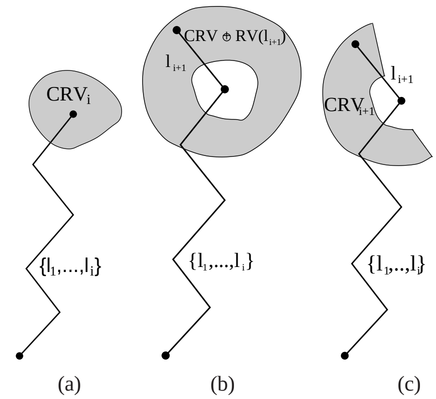


Fig. 4. (a) CRV_i is the constrained reachable volume of $\{l_1, \ldots, l_i\}$. (b) The region that is reachable by the endpoint of link l_{i+1} is $CRV_i \bigoplus$ ReachableVolume(l_{i+1}). (c) The constrained reachable volume of the linkage $\{l_1, \ldots, l_{i+1}\}$ is therefore ($CRV_i \bigoplus$ ReachableVolume(l_{i+1})) $\cap S_{i+1}$.

of the joints and end effectors. In [19] we show how reachable volume configurations in RV-space can be transformed into a C-space configuration including specifics for each joint type.

B. Reachable volumes and Minkowski sums

In this section we show that if you attach the base of a chain to the end effector of a second chain, then the reachable volume of the resulting chain is equal to the Minkowski sum of the reachable volumes of the original chains. We also show that the reachable volume of a chain is equivalent to the Minkowski sums of the reachable volumes of the links in that chain.

Lemma 1: If a chain C can be subdivided into two subchains C_1 and C_2, then the reachable volume of C is equal to the Minkowski sum of of the reachable volumes of C_1 and C_2.

Proof: Observe that a point in the reachable volume can be seen as an offset that is achievable by the chain. For example, if the point (x,y,z) is in the reachable set, then the corresponding chain can reach a point that is (x,y,z) from the base of the chain. This is a result of defining the origin to be the first point of the chain.

If C_1 can reach the point (x_1,y_1,z_1) and C_2 can reach the point (x_2,y_2,z_2), then we can attach a configuration of C_2 that reaches (x_2,y_2,z_2) to the end of a configuration of C_1 that reaches (x_1,y_1,z_1) to obtain a configuration of C that reaches (x_1+x_2,y_1+y_2,z_1+z_2). Consequently, if (x_1,y_1,z_1) is in the reachable set of C_1 and (x_2,y_2,z_2) is in the reachable set of C_2, then (x_1+x_2,y_1+y_2,z_1+z_2) must be in the reachable set of C.

Observe that if C can reach a point (x, y, z), then we can take a configuration of C that reaches (x, y, z) and split it into configurations of C_1 and C_2 in which the points that C_1 and C_2 reach (in their respective RV-spaces) sum to (x, y, z). We can therefore conclude that for a point (x, y, z) to be in C, there must exist a point (x_1, y_1, z_1) in the reachable set of C_1 and a point (x_2, y_2, z_2) in the reachable set C_2 such that $x_1 + x_2 = x$, $y_1 + y_2 = y$ and $z_1 + z_2 = z$.

The reachable set of C is therefore the following:

$$\texttt{Reachable}(C) = \{(x_1 + x_2, y_1 + y_2, z_1 + z_2) \mid (x_1, y_1, z_1) \in \texttt{Reachable}(C_1)\texttt{and}(x_2, y_2, z_2) \in \texttt{Reachable}(C_2)\}$$

This is equivalent to the Minkowski sum of the reachable volumes of C_1 and C_2:

$$\texttt{Reachable}(C) = \texttt{Reachable}(C_1) \bigoplus \texttt{Reachable}(C_2)$$

■

Corollary 1: The reachable volume of a chain is the Minkowski sum of the reachable volumes of the links in the chain.

$$\texttt{Reachable}(C) = \texttt{Reachable}(l_1) \bigoplus \texttt{Reachable}(l_2) \bigoplus \ldots \bigoplus \texttt{Reachable}(l_N)$$

Corollary 1 implies that the reachable volume of a chain can be computed by calculating the Minkowski sum of the reachable volumes of the links in the chain. These may be easily computed (see [19]). Note that Minkowski sums are commutative [23], which implies that the order in which the links occur in a linkages has no impact on the reachable volume of the chain.

C. Reachable volumes for constrained systems

The constrained reachable volume of a chain is the portion of RV-space that the end effector can reach without violating the constraints. Consider a chain C that is comprised of the links $L_c = \{l_1, l_2, \ldots, l_n\}$, joints $J_c = \{j_0, j_1, \ldots, j_m\}$, and RV-space constraints $S_c = \{S_0, S_2, \ldots, S_m\}$ placed on its joint positions. As a base case, the constrained reachable volume of the chain l_1 is Reachable(l_1) $\cap S_1$. Note that if S_1 is null, then the constrained reachable volume is the empty set. In this case, no configuration will satisfy the constraint. Now we make the inductive assumption that the constrained reachable volume of the linkage $\{l_1, \ldots, l_i\}$ is CRV_i (Figure 4(a)). In the linkage $\{l_1, \ldots, l_{i+1}\}$, the base of link l_{i+1} coincides with the end of link l_i, and CRV_i is the set of possible locations of this endpoint. The set of points the endpoint of l_{i+1} can reach is therefore $CRV_i \bigoplus$ Reachable(l_{i+1}) (Figure 4(b)), and the set of points that this endpoint can reach while satisfying the constraint S_{i+1} is ($CRV_i \bigoplus$ Reachable(l_{i+1})) $\cap S_{i+1}$ (Figure 4(c)). By induction, the constrained reachable volume of the chain C must be $RV(l_1) \cap S_1$ if $|L_c| = 1$ and $(RV(L_c - l_{|L_c|}, S_c - S_{|S_c|}) \bigoplus RV(l_{|L_c|})) \cap S_{|L_c|}$ otherwise.

V. Reachable Volume Visualization

We next show a set of examples which illustrate the nature of reachable volumes and demonstrate their capabilities. The method in Section IV-B for computing the reachable volume of a single joint could be applied to every joint, but that would require $\mathcal{O}(|J|^2)$ time, where $|J|$ is the number of joints. Here we use a method described in [19] that runs in $\mathcal{O}(|J|$*diameter(R)), where diameter(R) is the robot's diameter. More examples may be found in the supplemental video.

Figure 1(a) shows the reachable volumes for each link in a simple 4 linkage chain with spherical joints (totaling 9 units long), where each link is the same length and no constraints are present. Each link has its own reachable volume sphere.

Consider a 16 dof fixed-base grasper with spherical joints in an environment with a set of cubic objects. The reachable volume of various parts changes significantly when it is either constrained to grasp an object (Figure 1(b) or the base is constrained to a point (Figure 1(c)).

Figure 1(d) displays the reachable volume of a WAM robot [27] with 15 dof and a combination of spherical and planar joints whose end effectors are constrained to grasp a spherical object. To reach the object, the elbow joint must occupy the rightmost region, the second arm joint must be located in the middle region, and the wrist must be within the left region. The reachable volumes of the knuckles are inside this reachable volume.

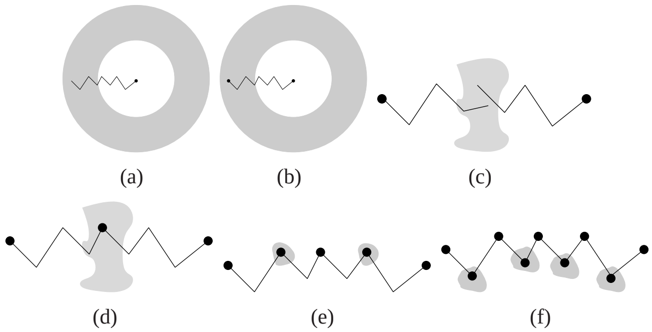


Fig. 5. Generating a configuration for a chain robot: (a) Compute the reachable volume of the chain. (b) Set the position of the end effector of the chain to be a point from this volume. (c) Bisect the chain and compute intersection of the reachable volumes of the two pieces. (d) Set the midpoint of the bisected chain to be a point from the intersection of these reachable volumes. (e,f) Continue until all joints are placed.

Algorithm 1 Generating configurations for chains

Input: A chain C

Output: A randomly sampled configuration of C by setting values for its dof

Compute the reachable volume RV_C of C
Set end effector of C to be a random point from RV_C
`SampleInternal(`C`)`
Convert Sample to C-Space Sample
Randomly sample translational + rotational coordinates

VI. Sampling with Reachable Volumes

We first describe how reachable volumes can be used to compute configurations for chains, tree-like robots, and closed chains without constraints and then with constraints.

A. Generating configurations for chains

To generate samples of a chain robot without constraints, we first compute the reachable volume of the end effector of the chain (Section IV-B). When generating multiple samples, this computation can be performed once as a prepossessing step. We then recursively position the internal joints of the chain by selecting a joint from the chain, "breaking" the chain at this joint, computing the reachable volumes of both pieces of the chain, and selecting a point from the intersection of their reachable volumes (see Figure 5). We recurse on the subchains formed by breaking the chain at the sampled joint. This process is given in Algorithms 1 and 2. We parse planar and prismatic joints first to ensure that they will be sampled after any spherical joints. In [19] we discuss how to sample joint positions from the intersection of reachable volumes.

B. Complexity

We next show that the running time of the reachable volume sampler is linear in the complexity of the reachable volumes and has a linear running time for unconstrained problems.

Proof: The sampler first computes the reachable volume of the chain by recursively breaking the chain. At the bottom level of this recursion, the sampler computes and returns the reachable volume of a single link which can be done in constant time and is done once per link. It then computes the Minkowski sums of these reachable volumes, performing a total of $\mathcal{O}(|L|)$ Minkowski sum operation (where $(|L|)$ is the number of links in the robot). In [19] we show that the complexity of computing the Minkowski sums of two reachable volumes is proportional to the complexity of the reachable volumes and that this complexity is $\mathcal{O}(1)$ in problems without constraints. The cost of this step is therefore $\mathcal{O}(|L|)$.

Algorithm 2 SampleInternal

Input: A chain C whose end effectors have already been sampled/set

Output: A randomly sampled configuration of C in RV-space by setting values for its dof

if C only has 1 link **then**
  return
Let j be the joint at the midpoint of C
Let C_l be the portion of C to the left of j
Let C_r be the portion of C to the right of j
RV_l = reachable volume of C_l
RV_r = reachable volume of C_r
The position of j = random point from $RV_l \cap (RV_r + \text{base}_r - \text{base}_l)$ + position of the base of C_l in RV-space of the robot
`SampleInternal(`C_l`)`
`SampleInternal(`C_r`)`

Once the reachable volume is computed, the algorithm samples each joint in the order that they were subdivided when computing the reachable volume of the chain. Consider an internal joint j which breaks the chain j_l through j_r. When computing the reachable volume of the chain in the first step of Algorithm 2, we recursively compute the reachable volumes of the chain j_l through j (which we will denote as $RV_{j_l,j}$) and j through j_r (RV_{j,j_r}). We next recall that reachable volumes are symmetric which means that the reachable volume of the chain j_r through j ($RV_{j_r,j}$) is equal to RV_{j,j_r}. Algorithm 2 samples j by placing j in the intersection of $RV_{j_l,j}$ and $RV_{j_r,j}$. Because $RV_{j_l,j}$ and $RV_{j_r,j}$ were computed while computing the reachable volume of the chain, we don't need to compute them during sampling. We only need to translate $RV_{j_l,j}$ by j_l and translate $RV_{j_r,j}$ by j_r, which is done by defining the bases of $RV_{j_l,j}$ and $RV_{j_r,j}$ to be at j_l and j_r (which can be done in constant time). j is then sampled by selecting a position from the intersection of $RV_{j_l,j}$ and $RV_{j_r,j}$ as described in Section VI-A.

Samples are generated by computing a bounding box or patch around the intersection of $RV_{j_l,j}$ and $RV_{j_r,j}$ (which can be done in constant time), and then repeatedly generating samples and testing if they are in $RV_{j_l,j}$ and $RV_{j_r,j}$. Testing if a point is in a reachable volume is equivalent to testing if the distance between the point and the base of the linkage (i.e., j_l or j_r) is between the minimum and maximum values for that linkage (as described in [19]) which can be done in constant time. Because we limit the total number of attempts to be a predefined constant, the total time to sample a joint's position (or return failure) is $\mathcal{O}(1)$. The total time to sample all the internal joints in the linkage is therefore $\mathcal{O}(|L|)$. In the final step, we convert the sample to a joint angle sample which can be done in $\mathcal{O}(|L|)$ time as described in Section VI-A. The total running time of the reachable volume sampler for problems without constraints is therefore linear with respect to the number of links in the robot. ■

C. Generating configurations for tree-like robots

To generate configurations for linkages with branches, we partition the linkage into a set of disjoint subchains (see Figure 6(d)). We then use use Algorithm 1 to generate a reachable volume configuration for each chain. We translate each reachable volume configuration into the RV-space of the root of the robot. We then convert this into a C-space configuration and randomly sample any rotational and translational coordinates. Algorithm pseudocode is provided in [19].

D. Generating configurations for closed chains

To compute configurations for closed chains, we decompose the closed chain into two open chains. We observe that if the two open

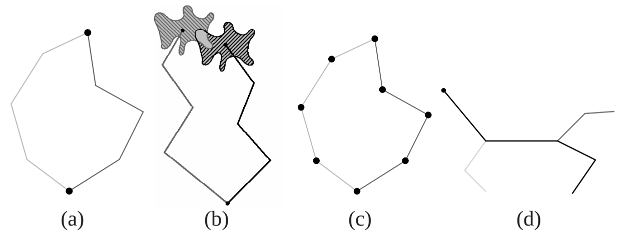


Fig. 6. Generating a closed chain configuration: (a) Break the closed chain into two open chains. (b) Compute the reachable volumes of the two chains (striped regions). (c) Randomly select a point from the intersection of their reachable volumes and use Algorithm 1 to sample the positions of the internal joints. (d) We generate a configuration for a tree-like robot by applying our method to each of the branches (black, dark gray, light gray).

chains are in configurations that share the same endpoints, then they can be combined to form a configuration of the closed chain. We now note that in RV-space both chains are rooted at the origin (we assume that the same endpoint is rooted for both linkages). In order for both chains to reach the same endpoint in workspace, they must reach the same point in RV-space (see Section IV). The set of possible positions for the end effectors of the two chains is therefore the intersection of the reachable volumes of the two chains. We can therefore select a point from this intersection to be the endpoint of the two chains (see Figure 6) and then sample the other points of the chains as described in Section VI-A. As with chains and tree-like robots, this method yields an RV-space configuration which can be transformed to a C-space configuration by setting the translational and rotational coordinates of the robot. Algorithm pseudocode is provided in [19].

E. Generating configurations for linkages and tree-like robots with constraints

It would be possible to compute samples for constrained problems in the same manner as unconstrained problems. However, our proof for the linear time complexity (Section VI-B) does not hold for problems with constraints. This proof relies on the ability to reuse the reachable volumes $RV_{jl,j}$ and $RV_{jr,j}$. Unfortunately, reachable volumes for problems with constraints are not symmetric and cannot be reused.

Algorithm 3 shows how to use reachable volumes to compute samples for problems with constraints. It first sets the position of the root of the robot to be (0,0,0) which is its location in RV-space by definition. It then selects an end effector j and calls the function ComputePartialRV (Algorithm 4) to set RV_j to be the reachable volume of $j \in J$ in the subset of the robot comprised of the children of j in the traversal. It then calls the function ComputeSampleHelper (Algorithm 5) which performs a second depth first traversal of this tree. During this traversal, the position of every joint j is set to be a random point in the intersection of RV_j and $P_{jprv} \bigoplus$ ReachableVolume(edge(j,parent(j))), which is the volume of space that j can occupy given the placement of its parent in the traversal. (See [19] for details on computing reachable volume intersections for different joint types.) After all of the joints have been sampled, Algorithm 3 transforms the resulting RV-space sample into a joint angle configuration and randomly samples the position and orientation of the robot to form a C-space sample.

Collectively, these algorithms perform two depth first traversals of the robot. During the first traversal we compute the partial reachable volume of each joint from the partial reachable volumes of its children in the traversal which requires one Minkowski sum operation and one intersection operation for each linkage in the robot. During the second traversal we sample each of the joints of the robot, compute the reachable volume of the link connecting the joint to its parent, translate that reachable volume by the position of the parent, compute the intersection of this reachable volume and the partial reachable volume of the joint (computed in the first traversal), and set the position of the joint to be a random point from

Algorithm 3 Compute sample for linkage or tree-like robot which satisfies constraints S

Input: A robot $R = (J, E)$ that contains no cycles, $root$ = an end effector of R, a set of constraints S on J
Output: A reachable volume configuration P that satisfies S

P_{root} = (0,0,0)
Let j be an arbitrary end effector that $\neq root$
RV_j = ComputePartialRV($j, \emptyset$,array($|J|$))
Let P_j be a random point from RV_j
$RVSample$ = ComputeSampleHelper($j, \emptyset, P_j, RV_j$)
c = CSpaceSample(RandomPosition, RandomOrientation, JointAngles($RVSample$))
return c

Algorithm 4 ComputePartialRV

Input: A robot $R = (J, E)$, a set of constraints S on J, and the subset P of J that have been assigned positions
Output: The reachable volume RV_j of j in the robot $R \setminus$ branch(j_{prv}) under the constraints S and the partial positioning P

ComputePartialRV(j, j_{prv}, RV)
if $j \in P$ **then**
  return Position(j)
$RV_j = S_j$
for all $j' \in$ Neighbors(j) $\setminus j_{prv}$ **do**
  $RV_{j'}$ = ComputePartialRV(j', j, RV)
  $RV_j = RV_j \cap RV_{j'}$
return RV

this intersection. The complexity of these operations is proportional to the complexity of the reachable volumes [19] so the running time of this method is $\mathcal{O}(|L|)$ in the complexity of the reachable volumes.

F. Sampling single loop closed chains with constraints

A single loop closed chain robot is a robot of genus 2. We first define the root of this robot to be one of the joints along the closed chain of the robot such that if the root were removed the robot would contain no cycles (Algorithm 6). We then select one of the root's neighbors j that is also located on the closed chain. Because the root is at the origin, edge($root, j$) imposes the constraint that j be the length of edge($root, j$)) away from the origin. We can therefore remove edge($root, j$) and replace it with a constraint that j be within ReachableVolume(edge($root, j$)) of the origin which is done by setting S_j to be $S_j \cap$ ReachableVolume(edge($root, j$)). (Recall that if there is no constraint on j, then S_j is the entire RV-space). Once the edge has been removed, the robot is a tree rooted at $root$. We then sample each of the branches in the same manner as with tree-like robots (Algorithm 3). We convert this sample to a joint angle configuration and randomly sample the translational and rotational coordinates of the robot. This algorithm performs two depth first traversals on each branch of the robot. Because complexity of these intersection and Minkowski sum operations is proportional to the complexity of the reachable volumes [19], the running time of this method is $\mathcal{O}(|L|)$ in the complexity of the reachable volumes.

VII. Probabilistic Completeness

In this section we discuss the sampling distribution of the reachable volume sampler and show that it is probabilistically complete.

Lemma 2: Joints are sampled uniformly in their reachable volume given the position of the joints that are already placed.

Algorithm 5 ComputeSampleHelper

Input: Joints j and j_{prv}, a set of partial positions P, and a reachable volume RV
Output: An updated set of partial positions P including j

```
if j ∈ P then
    return P
else
    if RV_j ∩ (P_jprv ⊕ ReachableVolume(edge(j, j_prv))) = ∅ then
        return Fail
    P_j = random point from RV_j ∩ (P_jprv ⊕ ReachableVolume(edge(j, j_prv)))
for all j' ∈ Neighbors(j) \ j_prv do
    ComputeSampleHelper(j', j, P, RV)
return P
```

Algorithm 6 Compute sample for closed chain robots which satisfies constraints S

Input: A robot $R = (J, E)$ that contains a single closed chain, $root$ = a joint on the closed chain in R, and a set of constraints S on J.
Output: A configuration that satisfies S

```
P_root = (0,0,0)
Let j be an arbitrary joint from Neighbors(root)
S_j = S_j ∩ ReachableVolume(edge(root, j))
Remove edge(root, j) from R
for all j' ∈ Neighbors(root) \ j do
    Let j be an end effector from the branch composed of j' and its descendants
    RV_j = ComputePartialRV(j, root, array(|J|))
    Let P_j be a random point from RV_j
    RVSample = ComputeSampleHelper(j, ∅, P, RV)
c = CSpaceSample(RandomPosition, RandomOrientation, JointAngles(RVSample))
return c
```

Proof: The joint sampling methods (see [19]) uniformly sample a domain that contains the reachable volume until they find a sample in the reachable volume resulting in a distribution that is uniform in the reachable volume. ■

Lemma 3: Reachable volume sampling is probabilistically complete.

Proof: The samplers iterate through a robot's joints and sample them in their reachable volume (the region they can reach given the position of the joints already sampled). The joint sampling methods sample over the entire reachable volume of a joint so we can inductively conclude that all possibly reachable volume space configurations can be sampled. There is a one to one correspondence between reachable volume samples and joint angle settings. Consequently the reachable volume sampler is complete over the range of joint angle coordinates. Our method uses the probabilistically complete reachable volume sampler to sample any joint angle coordinates and a uniform sampler (which is also probabilistically complete) to sample any translational and rotational coordinates resulting in a probabilistically complete sampler. ■

VIII. Results

In Figure 7 we show results for a **tunnel** environment in which a 70 dof chain must navigate within a tightly confined tunnel. We also show results for a **wheeled grasper** environment which consists of a 70 dof wheeled grasper that must navigate under a low hanging object while carrying a bucket that is constrained to remain upright. In the tunnel environment, we compare our method to uniform sampling and an Incremental CD checking method, while in the wheeled grasper environment we compare our method to Uniform sampling with filtering and CCD. Our results demonstrate that the reachable volume sampler gives a higher sampler success rate (Figure 7(c)) and requires less time to generate valid samples (Figure 7(d)) than existing methods. They also show that reachable volume sampling produces better connected roadmaps with a higher percentage of samples in the largest connected component (Figure 7(e)). A full set of results is contained in [19].

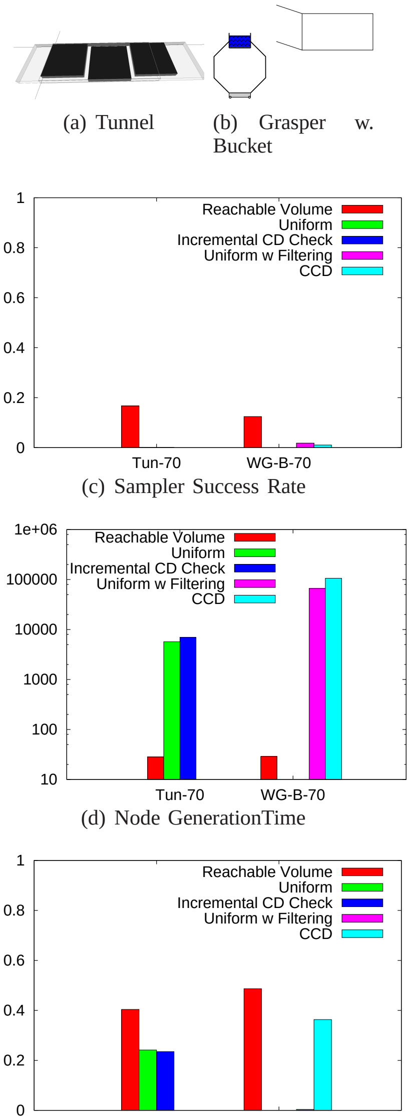


Fig. 7. Experimental results for tunnel(a) and wheeled-grasper w. bucket(b) environments. Stars indicate methods unable to generate samples in the allotted time. Note that (d) uses a log scale.

IX. Conclusion

In this work, we present the new concept of reachable volumes which denote the regions of workspace the joints and end effectors of a robot can reach. We show that the reachable volume of a chain can be computed from the Minkowski sum of the reachable volumes of the linkages composing the chain and provide a simple sampling method based on this property. Unlike many previous methods this sampler is applicable to complex robots (closed chains, tree-like robots) that include spherical and prismatic joints in addition to planar joints. Our method is also applicable to problems with end effector constraints, constraints on internal joints and problems with constraints on multiple joints/end effectors. We also present visualizations of reachable volumes for various constrained and unconstrained systems including closed chains and graspers. In the full paper [19] we show that our method can solve a wide variety of motion planning problems.

REFERENCES

[1] Peter Anderson-Sprecher and Reid Simmons. Voxel-based motion bounding and workspace estimation for robotic manipulators. In *International Conference on Robotics and Automation*, 2012.

[2] Juan Cortés and Thierry Siméon. Sampling-based motion planning under kinematic loop-closure constraints. In *Algorithmic Foundations of Robotics VI*, pages 75–90. Springer, Berlin/Heidelberg, 2005. book contains the proceedings of the International Workshop on the Algorithmic Foundations of Robotics (WAFR), Utrecht/Zeist, The Netherlands, 2004.

[3] John J. Craig. *Introduction to Robotics: Mechanics and Control, 2nd Edition*. Addison-Wesley Publishing Company, Reading, MA, 1989.

[4] Maxim Garber and Ming C. Lin. Constraint-based motion planning using Voronoi diagrams. In *Algorithmic Foundations of Robotics V*, pages 541–558. Springer, Berlin/Heidelberg, 2003. book contains the proceedings of the International Workshop on the Algorithmic Foundations of Robotics (WAFR), Nice, France, 2002.

[5] S. Gottschalk, M. C. Lin, and D. Manocha. OBB-tree: A hierarchical structure for rapid interference detection. *Comput. Graph.*, 30:171–180, 1996. Proc. SIGGRAPH '96.

[6] L. Han and N. M. Amato. A kinematics-based probabilistic roadmap method for closed chain systems. In *New Directions in Algorithmic and Computational Robotics*, pages 233–246. A. K. Peters, Boston, MA, 2001. book contains the proceedings of the International Workshop on the Algorithmic Foundations of Robotics (WAFR), Hanover, NH, 2000.

[7] L. Han, L. Rudolph, J. Blumenthal, and I. Valodzin. Convexly stratified deformation spaces and efficient path planning for planar closed chains with revolute joints. In *Int. J. Robot. Res.*, pages 1189–1212, 2008.

[8] Li Han. Hybrid probabilistic roadmap — Monte Carlo motion planning for closed chain systems with spherical joints. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 920–926, New Orleans, LA, 2004.

[9] Li Han and Lee Rudolph. Inverse kinematics for a serial chain with joints under distance constraints. In *Robotics Science and Systems II*, pages 177–184. The MIT Press, Cambridge, MA, 2007. book contains the proceedings of Robotics: Science and Systems (RSS) conference, Philadelphia, PA, 2006.

[10] Li Han and Lee Rudolph. Simplex-tree based kinematics of foldable objects as multi-body systems involving loops. In *Robotics Science and Systems IV*, page to appear. The MIT Press, Cambridge, MA, 2009. book contains the proceedings of Robotics: Science and Systems (RSS) conference, Zurich, Switzerland, 2008.

[11] Li Han and Lee Rudolph. A unified geometric approach for inverse kinematics of a spatial chain with spherical joints. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 4420–4427, Roma, Italy, 2007.

[12] Li Han, Lee Rudolph, Sam Dorsey-Gordon, Dylan Glotzer, Dan Menard, Jon Moran, and James R. Wilson. Bending and kissing : Computing self-contact configurations of planar loops with revolute joints. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 1346–1351, 2009.

[13] Li Han, Lee Rudolph, Michael Chou, Emily Eagle, Dylan Glotzae, Jake Kramer, Jonathan Moran, Christopher Pietras, Ammar Tareen, and Mathew Valko. Configurations and path planning of convex planar polygonal loops. In *Proc. Int. Workshop on Algorithmic Foundations of Robotics (WAFR)*, 2012.

[14] L. Jaillet and J.M. Porta. Path planning with loop closure constraints using an atlas-based RRT. In *15th International Symposium on Robotics Research*, 2011.

[15] Marcelo Kallmann, Amaury Aubel, Tolga Abaci, and Daniel Thalmann. Planning collision-free reaching motion for interactive object manipulation and grasping. *Computer Graphics Forum*, 22(3):313–322, September 2003.

[16] L. E. Kavraki, P. Švestka, J. C. Latombe, and M. H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Trans. Robot. Automat.*, 12(4):566–580, August 1996.

[17] S. M. LaValle and J. J. Kuffner. Randomized kinodynamic planning. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 473–479, 1999.

[18] S.M. LaValle, J.H. Yakey, and L.E. Kavraki. A probabilistic roadmap approach for systems with closed kinematic chains. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 1671–1676, Detroit, MI, 1999.

[19] T. McMahon, S. Thomas, and N. M. Amato. Sampling based motion planning with reachable volumes. Technical Report TR13-008, Department of Computer Science, Texas A&M University, College Station Tx., 2013.

[20] Jean-Pierre Merlet. Still a long way to go on the road for parallel mechanisms. In *ASME Bienneal Mech. Rob. Conf.*, Montreal, Canada, 2002.

[21] R. James Milgram and Jeff C. Trinkle. The geometry of configuration spaces for closed chains in two and three dimensions. *Homology, Homotopy Appl.*, 6(1):237–267, 2004.

[22] Giuseppe Oriolo and Christian Mongillo. Motion planning for mobile manipulators along given end-effector paths. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 2154–2160, Barcelona, Spain, 2005.

[23] Rolf Schneider. *Convex bodies: the Brunn-Minkowski theory*. Cambridge University Press, 1993.

[24] Chansu Suh, Beobkyoon Kim, and Frank C. Park. The tangent bundle RRT algorithms for constrained motion planning. In *13th World Congress in Mechanism and Machine Science*, 2011.

[25] X. Tang, S. Thomas, P. Coleman, and N. M. Amato. Reachable distance space: Efficient sampling-based planning for spacially constrained systems. *Int. J. Robot. Res.*, 29(7):916–934, 2010.

[26] Jeff C. Trinkle and R. James Milgram. Complete path planning for closed kinematic chains with spherical joints. *Int. J. Robot. Res.*, 21 (9):773–789, 2002.

[27] WAM. http://www.barrett.com/robot/products-arm.htm.

[28] Li-Chun Tommy Wang and Chih Cheng Chen. A combined optimization method for solving the inverse kinematics problem of mechanical manipulators. *IEEE Trans. Robot. Automat.*, 7(4):489–499, 1991.

[29] D. Xie and N. M. Amato. A kinematics-based probabilistic roadmap method for high DOF closed chain systems. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 473–478, New Orleans, LA, 2004.

[30] Jeffery H. Yakey, Steven M. LaValle, and Lydia E. Kavraki. Randomized path planning for linkages with closed kinematic chains. *IEEE Trans. Robot. Automat.*, 17(6):951–958, 2001.

[31] Zhenwang Yao and K. Gupta. Path planning with general end-effector constraints: using task space to guide configuration space search. In *Proc. IEEE Int. Conf. Intel. Rob. Syst. (IROS)*, pages 1875–1880, Edmonton, Alberta, Canada, 2005.

[32] A. Yershova and S. M. LaValle. Motion planning for highly constrained spaces. In *Robot Motion and Control*, pages 297–306, 2009.

[33] A. Yershova, L. Jaillet, T. Simeon, and S. M. LaValle. Dynamic-domain RRTs: Efficient exploration by controlling the sampling domaine. In *ICRA*, pages 3867–3872, 2005.

[34] Y. Zhang, K. Hauser, and J. Luo. Unbiased, scalable sampling of closed kinematic chains. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2013.

[35] M. Zucker, N. Ratli, A. D. Dragan, M. Pivtoraiko, M. Klingensmith, C. M. Dellin, J. A. Bagnell, and S. S. Srinivasa. Chomp: Covariant hamiltonian optimization for motion planning. In *The International Journal of Robotics Research*, volume 32, pages 1164–1193, 2013.