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 grasper as well as for examples with joint and end effector constraints.

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 [28], and animation [3].

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 [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 [23], and inverse kinematics [4]) which focus on end effector constraints, our method allows for constraints on internal joints as well as on 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
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 [13].

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 [15].

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 [12, 20]. 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 [11]. In [22, 23], 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 [24] Zhang and Hauser present a Monte Carlo method for generating closed chain samples.

Atlas-RRT [44] and Tangent Bundle based RRT [24] construct RRTs along manifolds in the environment, while DDRT [33] 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 [25] and multiple loops [6]. 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, 26].

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 [8]. 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 hard constraints and optimize adherence to soft constraints.

3) Enforcing constraints during sampling: Some approaches explicitly enforce constraints while sampling. Closed chain problems may be 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 [8]. Other planners require the end effector to traverse a predefined trajectory by generating samples

<table>
<thead>
<tr>
<th>Method</th>
<th>Constraint Types</th>
<th>High dof Robots</th>
<th>Closed Chains</th>
<th>Tree-like Robots</th>
<th>Joint Types</th>
<th>Multi-Query Problems</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic PRM [16]</td>
<td>None</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Planar, Spherical, Prismatic</td>
<td>Yes</td>
</tr>
<tr>
<td>Inverse Kinematics [6]</td>
<td>End Effector</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Planar</td>
<td>Yes</td>
</tr>
<tr>
<td>Constrained Dynamics [4]</td>
<td>End Effector</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Planar</td>
<td>Yes</td>
</tr>
<tr>
<td>I-CD</td>
<td>None</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Planar, Spherical, Prismatic</td>
<td>Yes</td>
</tr>
<tr>
<td>DDRRT [33, 32], Atlas RRT [13], Tangent Bundle RRT [24]</td>
<td>Only End Effector Shown</td>
<td>Not shown</td>
<td>Yes</td>
<td>Not shown</td>
<td>Only Planar Shown</td>
<td>No</td>
</tr>
<tr>
<td>CCD [28]</td>
<td>End Effector</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Planar, Spherical</td>
<td>Yes</td>
</tr>
<tr>
<td>CHOMP [35]</td>
<td>Hard and Soft</td>
<td>Yes</td>
<td>Not Shown</td>
<td>Yes</td>
<td>Planar, Spherical, Combinations</td>
<td>No</td>
</tr>
<tr>
<td>Reachable Distances [25]</td>
<td>End Effector</td>
<td>Yes</td>
<td>Yes</td>
<td>Not shown</td>
<td>Planar, Prismatic</td>
<td>Yes</td>
</tr>
<tr>
<td>Reachable Volumes (this paper)</td>
<td>End Effector, Internal Joints, Multiple Joints</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Planar, Spherical, Prismatic, Combinations</td>
<td>Yes</td>
</tr>
</tbody>
</table>

TABLE I

COMPARISON OF METHOD CAPABILITIES.
that satisfy the end effector constraints given by the trajectory [22]. Other methods can generate samples with self-contact [14]. 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 \( A \) 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.

The reachable distance of an articulated linkage is the range of distances that its end effector can reach with respect to its base [23]. 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)).

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 \( \theta \). Spherical joints are 2 dof joints in which any possible angle between adjacent links is valid denoted by an inclination \( \theta \) and a rotation \( \phi \). 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 \( S \) to some or all of the joints. Configurations must satisfy \( S \) 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. \( \square \)).

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 2(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 2(b)). For chains with a single link of length \( l \) 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 \( r_{\text{min}} = r_{\text{max}} = l \). If the link has an adjacent prismatic joint that ranges between \( d_{\text{min}} \) and \( d_{\text{max}} \), then the reachable set is the set of points represented by the radii \( r_{\text{min}} = l + d_{\text{min}} \) and \( r_{\text{max}} = l + d_{\text{max}} \). 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 \( r_{\text{min}} \) and maximum radius \( r_{\text{max}} \) such that the reachable volume of a chain is the set of points \( p \) whose distance from the origin \( O \) is between \( r_{\text{min}} \) and \( r_{\text{max}} \).

\[
\text{RV(Chain)} = \{ p \mid r_{\text{min}} \leq \text{distance}(p, O) \leq r_{\text{max}} \}
\]

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...
of the joints and end effectors. In [13], 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 is subdivided into two subchains C1 and C2, then the reachable volume of C is equal to the Minkowski sum of the reachable volumes of C1 and C2.

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 C1 can reach the point (x1, y1, z1) and C2 can reach the point (x2, y2, z2), then we can attach a configuration of C2 that reaches (x2, y2, z2) to the end of a configuration of C1 that reaches (x1, y1, z1) to obtain a configuration of C that reaches (x1 + x2, y1 + y2, z1 + z2). Consequently, if (x1, y1, z1) is in the reachable set of C1 and (x2, y2, z2) is in the reachable set of C2, then (x1 + x2, y1 + y2, z1 + z2) 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 C1 and C2 in which the points that C1 and C2 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 (x1, y1, z1) in the reachable set of C1 and a point (x2, y2, z2) in the reachable set C2 such that x1 + x2 = x, y1 + y2 = y, and z1 + z2 = z.

The reachable set of C is therefore the following:

Reachable(C) = {(x1 + x2, y1 + y2, z1 + z2) | (x1, y1, z1) ∈ Reachable(C1) and (x2, y2, z2) ∈ Reachable(C2)}

This is equivalent to the Minkowski sum of the reachable volumes of C1 and C2:

Reachable(C) = Reachable(C1) ⊕ Reachable(C2)

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

Reachable(C) = Reachable(l1) ⊕ Reachable(l2) ⊕ ... ⊕ Reachable(lN)

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 [13]). 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 Lc = {l1, l2, ..., ln}, joints Jc = {j0, j1, ..., jm}, and RV-space constraints Sc = {S0, S1, ..., Sn} placed on its joint positions. As a base case, the constrained reachable volume of the chain l1 is Reachable(l1) ∩ S1. Note that if S1 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 {l1, ..., li} is CRV i (Figure 4(a)). In the linkage {l1, ..., li+1}, the base of link li+1 coincides with the end of link li, and CRV i is the set of possible locations of this endpoint. The set of points the endpoint of li+1 can reach is therefore CRV i ⊕ Reachable(li+1) (Figure 4(b)), and the set of points that this endpoint can reach while satisfying the constraint Si+1 is (CRV i ⊕ Reachable(li+1)) ∩ Si+1 (Figure 4(c)). By induction, the constrained reachable volume of the chain C must be RV (l1) ∩ S1 if |Jc| = 1 and (RV (Lc − li−1) ∩ Sc = S1 ∩ Si+1) 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 4.4.4.3 for computing the reachable volume of a single joint could be applied to every joint, but that would require O(|J|2) time, where |J| is the number of joints. Here we use a method described in [13] that runs in 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 [29] 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.
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 5.3.1). 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 [13] 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 $O(|L|)$ 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 $O(1)$ in problems without constraints. The cost of this step is therefore $O(|L|)$.

Algorithm 1 Generating configurations for chains

Input: A chain C
Output: A randomly sampled configuration of C by setting values for its dof
1: Compute the reachable volume $RV_C$ of C
2: Set end effector of C to be a random point from $RV_C$
3: SampleInternal(C)
4: Convert Sample to C-Space Sample
5: Randomly sample translational + rotational coordinates

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
1: if C only has 1 link then
2: return
3: Let $j$ be the joint at the midpoint of C
4: Let $C_l$ be the portion of C to the left of $j$
5: Let $C_r$ be the portion of C to the right of $j$
6: $RV_l =$ reachable volume of $C_l$
7: $RV_r =$ reachable volume of $C_r$
8: The position of $j = \text{random point from } RV_l \cap (RV_r + \text{base,} -\text{base}) + \text{position of the base of } C_i \text{ in RV-space of the robot}$
9: SampleInternal($C_l$)
10: 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 $j_1$ 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 $j_2$ through $j_r$ (which we will denote as $RV_{j_2,r}$) and $j_1$ through $j_r$ ($RV_{j_1,r}$). We next recall that reachable volumes are symmetric which means that the reachable volume of the chain $j_2$ through $j_r$ ($RV_{j_2,r}$) is equal to the reachable volume of the chain $j_1$ through $j_r$ ($RV_{j_1,r}$). Algorithm 2 samples $j$ by placing $j$ in the intersection of $RV_{j_1,r}$ and $RV_{j_2,r}$. Because $RV_{j_1,r}$ and $RV_{j_2,r}$ were computed while computing the reachable volume of the chain, we don’t need to compute them during sampling. We only need to translate $RV_{j_1,r}$ by $j_1$ and translate $RV_{j_2,r}$ by $j_r$, which is done by defining the bases of $RV_{j_1,r}$ and $RV_{j_2,r}$ to be at $j_1$ and $j_r$ (which can be done in constant time). $j$ is then sampled by selecting a position from the intersection of $RV_{j_1,r}$ and $RV_{j_2,r}$ as described in Section VI.A.

Samples are generated by computing a bounding box or patch around the intersection of $RV_{j_1,r}$ and $RV_{j_2,r}$ (which can be done in constant time), and then repeatedly generating samples and testing if they are in $RV_{j_1,r}$ and $RV_{j_2,r}$. 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., $j_1$ or $j_r$) is between the minimum and maximum values for that linkage (as described in [13]) 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 $O(1)$. The total time to sample all the internal joints in the linkage is therefore $O(|L|)$. In the final step, we convert the sample to a joint angle sample which can be done in $O(|L|)$ 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). 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
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 VI-V). 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 3) 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 [13].

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 $RV_j$ and $RV_{ij}$. Unfortunately, reachable volumes for problems with constraints are not symmetric and cannot be reused.

Algorithm 4 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 3) 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 $P_{prev}$. $RV_{prev}$ is the volume of space that can occupy given the placement of its parent in the traversal. (See [13] 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 this intersection. The complexity of these operations is proportional to the complexity of the reachable volumes [13] so the running time of this method is $O(|L|)$ 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 [13], the running time of this method is $O(|L|)$ 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 $j_{prev}$, a set of partial positions $P$, and a reachable volume $RV$

Output: An updated set of partial positions $P$ including $j$

1: if $j \in P$ then
2:    return $P$
3: else
4:    if $RV_j \cap (P_{j_{prev}} \oplus \text{ReachableVolume}(edge(j, j_{prev}))) = \emptyset$ then
5:        return Fail
6:    $P_j = \text{random point from } RV_j \cap (P_{j_{prev}} \oplus \text{ReachableVolume}(edge(j, j_{prev})))$
7:    for all $j' \in \text{Neighbors}(j) \setminus j_{prev}$ do
8:        ComputeSampleHelper($j', j, P, RV$)
9:    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 $S$

1: $P_{root} = (0, 0, 0)$
2: Let $j$ be an arbitrary joint from Neighbors($root$)
3: $S_j = S_j \cap \text{ReachableVolume}(edge(root, j))$
4: Remove edge($root, j$) from $R$
5: for all $j' \in \text{Neighbors}(root) \setminus j$ do
6:    Let $j$ be an end effector from the branch composed of $j'$ and its descendants
7:    $RV_j = \text{ComputePartialRV}(j, root, \text{array}([J]))$
8:    let $P_j$ be a random point from $RV_j$
9:    $RV_{Sample} = \text{ComputeSampleHelper}(j, \emptyset, P, RV)$
10:   $c = \text{CSpaceSample(RandomPosition, RandomOrientation, JointAngles(RV_{Sample}))}$
11: 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 possible 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].

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

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.
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 [13] we show that our method can solve a wide variety of motion planning problems.

REFERENCES


