

Quotient-Space Motion Planning

Andreas Orthey and Adrien Escande and Eiichi Yoshida

Abstract—A motion planning algorithm computes the motion of a robot by computing a path through its configuration space. To improve the runtime of motion planning algorithms, we propose to nest robots in each other, creating a nested quotient-space decomposition of the configuration space. Based on this decomposition we define a new roadmap-based motion planning algorithm called the Quotient-space roadMap Planner (QMP). The algorithm starts growing a graph on the lowest dimensional quotient space, switches to the next quotient space once a valid path has been found, and keeps updating the graphs on each quotient space simultaneously until a valid path in the configuration space has been found. We show that this algorithm is probabilistically complete and outperforms a set of state-of-the-art algorithms implemented in the open motion planning library (OMPL).

I. INTRODUCTION

Motion planning algorithms are fundamental for robotic applications like product assembly, manufacturing, disaster response, elderly care or space exploration.

A motion planning algorithm takes as input a robot, its configuration space, an environment, a start and a goal configuration, and computes as output a path between start and goal if one exists [1]. This computation is NP-hard [2] scaling exponentially with the number of dimensions of the configuration space. Thus, the more degrees-of-freedom (dof) a robot has, the higher the runtime of the motion planning algorithm will be. This can become the limiting factor of any robotics application. It is therefore important to find suitable decompositions of the configuration space, such that a planning algorithm can quickly discover the relevant parts of the configuration space.

We developed a new decomposition of a given configuration space \mathcal{M} , which decomposes \mathcal{M} into a sequence of nested subspaces. We observe that any configuration space \mathcal{M} can be written as a product of subspaces

$$\mathcal{M} = \mathcal{M}_1 \times \cdots \times \mathcal{M}_K \quad (1)$$

This suggests we can decompose the configuration space in the following way: Start with the product of subspaces and successively remove one subspace after another. This leads to a sequence of nested subspaces as

$$\mathcal{M}_1 \subset \mathcal{M}_1 \times \mathcal{M}_2 \subset \cdots \subset \mathcal{M}_1 \times \cdots \times \mathcal{M}_K \quad (2)$$

Each subspace in this sequence is called a quotient-space, and the sequence itself is called a quotient-space decomposition [3].

The authors are with the CNRS-AIST Joint Robotics Laboratory, UMI3218/RL, Tsukuba-shi 305-8560, Japan (e-mail: {andreas.orthy, adrien.escande, e.yoshida}@aist.go.jp)

This research was supported by Grant-in-Aid for JSPS Research Fellow 16F116701

It turns out that each quotient-space can be represented by nesting a simpler robot inside the original robot. The prototypical example is a rigid body free-floating in space. The configuration space is $SE(3) = \mathbb{R}^3 \times SO(3)$, and we can decompose it as $\mathbb{R}^3 \subset \mathbb{R}^3 \times SO(3)$. The subspace \mathbb{R}^3 is called the quotient-space and represents a sphere nested inside the rigid body, abstracting the orientations of the rigid body.

Such a decomposition is advantageous: Imagine the sphere being infeasible at point $p \in \mathbb{R}^3$. Then the rigid body is infeasible at all configurations inside the subspace $p \times SO(3)$. We call this the necessary condition of nested robots.

We have developed a new algorithm, called the Quotient-space roadMap Planner (QMP), which is able to exploit such a quotient-space decomposition. We first show how to build a quotient-space decomposition for any robot in Sec. III. We then discuss the inner workings of QMP in Sec. IV, prove that QMP is probabilistically complete (Sec. V), and we develop three heuristics to improve its runtime (Sec. VI). Finally, we demonstrate that QMP (Sec. VII) can be applied to free-floating rigid bodies, free-floating articulated bodies, fixed-base serial chains and fixed-base tree chain robots.

II. RELATED WORK

We review two categories of papers. First, we review quotient-space decompositions and their application to motion planning. Second, we review alternative decomposition methods.

Quotient-space decompositions are ubiquitous in mathematics, appearing as quotient-groups in algebra, filtrations in linear algebra, or nests in functional analysis. The construction of simplicial complexes in algebraic topology is a prominent example of a quotient-space decomposition.

The application of quotient spaces to continuous spaces and decision making has been originally developed by [4] and [5]. In robotics, quotient-spaces have been used, albeit under different names. Bretl proposed a two-level decomposition: first a path on a stance graph is planned and then configurations along the path are sampled [6]. The stance graph can be seen as a quotient space of the configuration space by the stance subspaces. Grey et al. [7] use a two-level quotient space decompositions, embedding the torso inside a humanoid robot. A similar idea can be found in Tonneau et al.[8], approximating a robot by a simpler model. Both methods are similar to ours, but use only a two-level decomposition, and do not define the nesting procedure for general robots.

The closest approach to ours is the multi-level decomposition scheme by [9] whereby a sequence of nested robots is

created. Their planning algorithm starts with the lowest dof robot, computes a path, and uses this path as a constraint for the next bigger robot. This approach can neither deal with non-simple paths (see Fig. 4), nor with spurious paths (see Fig. 5, Sec. VI). It is thus not complete.

Closely related is the exploration/exploitation tree (EET) algorithm [10]. The authors compute a sequence of spheres in the workspace, called a tunnel, to approximate the space of collision-free paths. However, this method is not complete, and cannot handle spurious shortest tunnels/paths (see Sec. VI).

The motion planning problem can also be decomposed without using quotient-spaces.

Kunz et al. [11] use a hierarchical rejection sampling approach to improve the Informed-RRT* algorithm [12]. While the focus is different from ours, their method is methodically similar to our algorithm, where samples are discarded if they are not necessary feasible.

Our work is closely related to low-dimensional sampling techniques which guide configuration space sampling. The algorithm by van den Berg and Overmars [13] precomputes narrow passages of the workspace and uses those passages to sample the space more densely in those areas. A similar idea can be found in [14], where the authors discretize the workspace, compute a shortest path in the workspace between start and goal configuration, and then sample from a cell in workspace in proportion to the cell's distance to the shortest path. Closely related is also the dimensionality reduction method by [15], which considers ignoring paths not having a minimal swept volume.

The KPIECE algorithm [16] is another example of a hierarchical decomposition. The environment is divided into smaller and smaller boxes until a certain threshold is reached. A small box corresponds to workspace points near to a boundary. Those areas are sampled more frequently to effectively guide samples towards the configuration space boundary. This algorithm is orthogonal to ours: they decompose the environment, we decompose the robot.

III. QUOTIENT-SPACE DECOMPOSITION

First, we describe the idea of a quotient space and show two applications to the vector space \mathbb{R}^2 and the manifold $\text{SO}(2)^2$. Second, we show how robots can be nested in each other and thereby create a sequence of quotient spaces. Third, we show that being feasible in a quotient space is a necessary condition for being feasible in the configuration space.

A. Quotient Space

Let M be a vector space and C be a subspace of M . Then the quotient space of M by C , denoted by M/C is the space obtained by collapsing all equivalence classes of C in M to zero [3]. Collapsing a space is done by creating an equivalence relation \sim on M : for all $x, y \in M$ we have that $x \sim y$ if $x - y \in C$. This relation creates a partition of equivalence classes on the vector space. The set of equivalence classes is called the quotient space.

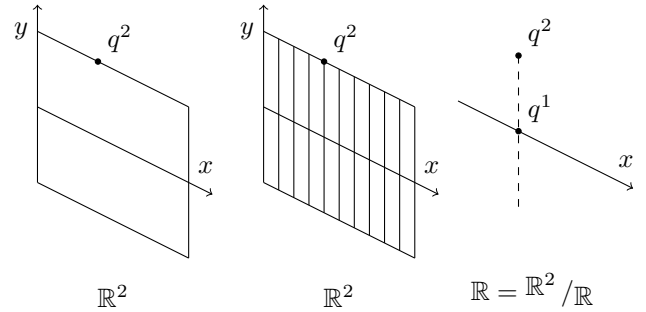
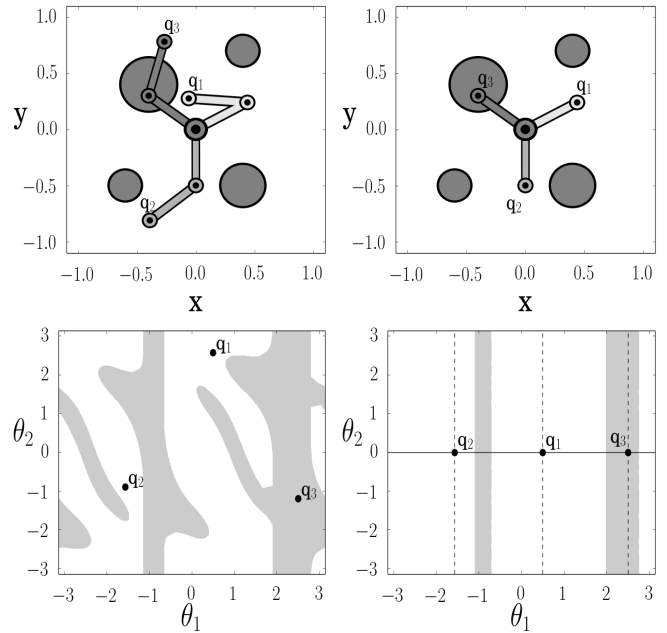


Fig. 1. Visualization of Quotient Space as collapsing of equivalence classes.



$$\mathcal{M}_1 = \text{SO}(2) \times \text{SO}(2) \quad \mathcal{M}_0 = \mathcal{M}_1 / \text{SO}(2)$$

Fig. 2. Quotient Space Decomposition for a 2-dof manipulator. Left Top: Environment with four obstacles. We have visualized three distinct configurations of the robot. Left Bottom: The configuration space with gray areas being infeasible configurations. Right Top: Environment with 1-dof nested manipulator at same configurations. Right Bottom: Configuration space is collapsed onto the quotient space. Being feasible in the quotient space is a necessary condition for feasibility in the configuration space. It can be seen that q_3 is infeasible, thus not fulfilling the necessary condition.

As an example, consider the vector space \mathbb{R}^2 and its subspace \mathbb{R} . We can partition \mathbb{R}^2 into equivalence classes of its subspace \mathbb{R} such that two points $x, y \in \mathbb{R}^2$ are equivalent if $x - y \in 0 \times \mathbb{R}$. We visualize this in Fig. 1, where \mathbb{R}^2 is first partitioned into the equivalence classes of \mathbb{R} (the vertical lines), and then all lines are collapsed to yield \mathbb{R} . The point $q^2 \in \mathbb{R}^2$ and all points on the dashed line are equivalent and therefore collapsed into the single point $q^1 \in \mathbb{R}$. We identify $\mathbb{R} = \mathbb{R}^2 / \mathbb{R}$ to denote the quotient space.

Quotient spaces generalize to manifolds [17]. Consider $\mathcal{M} = \text{SO}(2) \times \text{SO}(2)$, the configuration space of a 2-dof fixed-base manipulator in the plane. Two points $x, y \in \text{SO}(2)$ are equivalent if $x - y \in 0 \times \text{SO}(2)$. The manifold \mathcal{M} can be partitioned into equivalence classes of $\text{SO}(2)$, and then each equivalent class is collapsed to yield $\text{SO}(2) = \mathcal{M} / \text{SO}(2)$. This has been visualized in Fig. 2. On the left

three different configurations are shown and their position in the configuration space. On the right the quotient space is shown, corresponding to a 1-dof manipulator nested inside the 2-dof manipulator. Two interesting observations can be made in the quotient space: First, the configuration q_3 is infeasible, regardless of how the second joint is moved. Second, there does not exist a path between q_1 and q_2 . This can be inferred solely from the 1-dof manipulators configuration space.

This example shows the defining feature of quotient spaces: If a robot is nested inside another robot, then the feasibility of the nested robot is a necessary condition for the feasibility of the other robot¹. We will proceed to define how robots can be nested in each other, and we will show that this definition indeed leads to the necessary condition for feasibility.

B. Nesting of Robots

A robot \mathcal{R}_i is nested inside another robot \mathcal{R}_{i+1} if two conditions are fulfilled: First, the configuration space of \mathcal{R}_i is a subspace of \mathcal{R}_{i+1} , and second, the volume of the body of \mathcal{R}_i at each configuration must be a subset of the volume of the body of \mathcal{R}_{i+1} .

Let \mathcal{M}_i be the configuration space of robot \mathcal{R}_i , and let $\mathbf{V}_i(p) \subset \mathbb{R}^3$ be the volume of the body of robot \mathcal{R}_i at configuration $p \in \mathcal{M}_i$.

Definition 1 (Nested Robot). *Let $\mathcal{R}_i, \mathcal{R}_{i+1}$ be given. We say that \mathcal{R}_i is nested in \mathcal{R}_{i+1} , denoted as $\mathcal{R}_i \subseteq \mathcal{R}_{i+1}$, if*

- (1) $\mathcal{M}_{i+1} = \mathcal{M}_i \times \mathbf{C}_{i+1}$ such that $\mathcal{M}_i = \mathcal{M}_{i+1} / \mathbf{C}_{i+1}$
- (2) $\mathbf{V}_i(p) \subseteq \mathbf{V}_{i+1}(p \circ q)$ for any $p \in \mathcal{M}_i$ and $q \in \mathbf{C}_{i+1}$

whereby the operation \circ is the cartesian product defined as $p \circ q = (p, q) \in \mathcal{M}_{i+1}$.

Given a robot \mathcal{R} and a sequence of nested robots $\mathcal{R}_1 \subseteq \dots \subseteq \mathcal{R}_K = \mathcal{R}$, the configuration spaces define a decomposition of \mathcal{M} as

$$\mathcal{M}_1 \subseteq \dots \subseteq \mathcal{M}_K \quad (3)$$

This decomposition will be called a quotient-space decomposition.

C. Necessary Conditions

From Definition 1 we can infer the key property of the quotient-space decomposition: if a nested robot is infeasible, then so is the original robot.

Let \mathbf{E} denote the environment, the subset of \mathbb{R}^3 containing obstacles. We say that robot \mathcal{R}_i is feasible at configuration $q \in \mathcal{M}_i$ if $\mathbf{V}_i(q) \cap \mathbf{E} = \emptyset$.

Theorem 1 (Necessary condition of nested robot). *If $\mathcal{R}_i \subseteq \mathcal{R}_{i+1}$, $p \in \mathcal{M}_i$, and $\mathbf{V}_i(p) \cap \mathbf{E} \neq \emptyset$ (robot \mathcal{R}_i is infeasible at p), then $\mathbf{V}_{i+1}(p \circ q) \cap \mathbf{E} \neq \emptyset$ for any $q \in \mathbf{C}_{i+1}$ (robot \mathcal{R}_{i+1} is infeasible everywhere).*

Proof. Since $\mathbf{V}_i(p) \cap \mathbf{E} \neq \emptyset$ and $\mathbf{V}_i(p) \subseteq \mathbf{V}_{i+1}(p \circ q)$ for any $q \in \mathbf{C}_{i+1}$, it must follow that $\mathbf{V}_{i+1}(p \circ q) \cap \mathbf{E} \neq \emptyset$. \square

¹Zhang and Zhang [4] call this the falseness-preserving property of quotient space decompositions

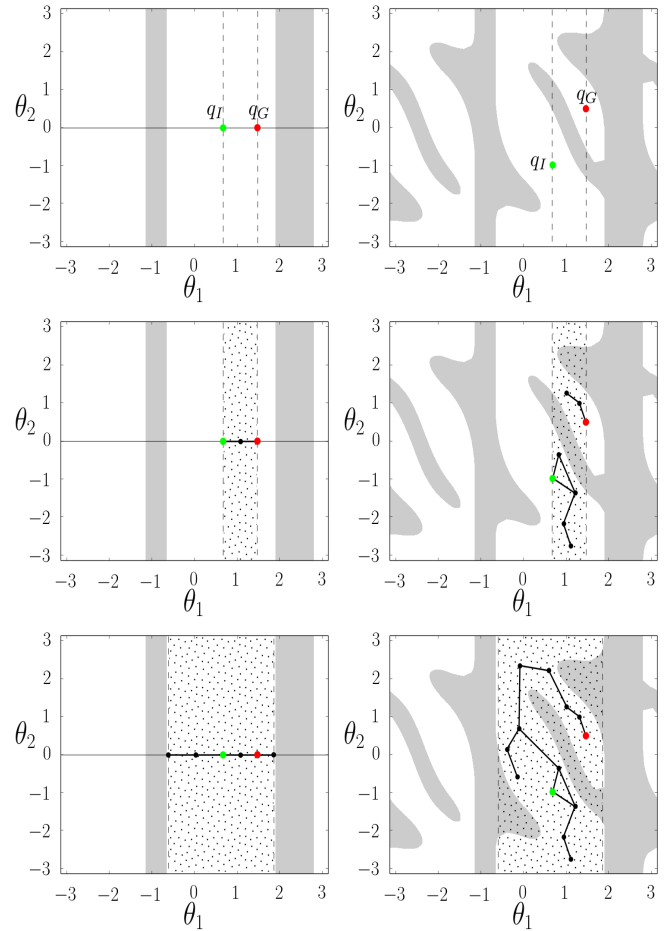


Fig. 3. Schematic of QMP for a two-level decomposition. See text for clarification.

Theorem 1 implies that if a configuration $p \in \mathcal{M}_i$ is infeasible, then so is the subspace $p \times \mathbf{C}_{i+1}$. We can exploit this fact to ignore subspaces of the configuration space, thereby developing a motion planning algorithm with lower runtime.

IV. QUOTIENT-SPACE ROADMAP PLANNER

The Quotient-Space roadMap Planner (QMP) works as depicted in Fig. 3. On the top right the configuration space of the 2-dof fixed-base manipulator is shown with a start configuration (green) and a goal configuration (red). The corresponding start and goal configurations on the quotient-space are shown in the top left figure. In the first step, a graph is grown on the quotient space (middle left). Once a valid path has been found between start and goal configuration, a second graph is grown on the configuration space (middle right), whereby the samples are constrained to lie above the quotient-space graph. Both graphs are simultaneously grown (bottom left), until a valid path has been found on the configuration space (bottom right), or until a time limit has been reached. For more than two quotient-spaces, this idea is iteratively continued.

A. Quotient-Space Roadmap

To simplify the algorithmic development, we group each quotient-space with its associated objects into a tuple called

the quotient-space roadmap.

Let \mathcal{M} be the N -dimensional configuration space of robot \mathcal{R} . We consider a nested sequence of K robots $\mathcal{R}_1 \subset \dots \subset \mathcal{R}_K = \mathcal{R}$, such that \mathcal{M} is decomposed into a sequence of quotient spaces as $\mathcal{M}_1 \subset \dots \subset \mathcal{M}_K = \mathcal{M}$.

To each quotient space \mathcal{M}_k we associate a start configuration q_k^I , a goal configuration q_k^G , a graph \mathbf{G}_k , a shortest path \mathbf{p}_k on \mathbf{G}_k between q_k^I and q_k^G , and a density V_k defined on $\mathbf{G}_{k-1} \times \mathbf{C}_k$ as

$$V_k = \frac{|\mathbf{G}_k|}{\mu(\mathbf{C}_k)L_{k-1}} \quad (4)$$

whereby $|\mathbf{G}_k|$ are the number of vertices of \mathbf{G}_k , L_{k-1} is the sum of all edge lengths of \mathbf{G}_{k-1} , and $\mu(\mathbf{C}_k)$ is the n^k -dimensional measure of \mathbf{C}_k . The density is used to decide which graph should be grown next.

We group all elements together into the quotient-space roadmap

$$\mathbf{Q}_k = \{q_k^I, q_k^G, \mathbf{C}_k, \mathcal{M}_k, \mathbf{G}_k, \mathbf{p}_k, V_k, \mathbf{Q}_{k-1}\} \quad (5)$$

with $\mathbf{C}_k = \mathcal{M}_k / \mathcal{M}_{k-1}$, $\mathcal{M}_0 = \emptyset$, $\mathbf{Q}_0 = \emptyset$.

B. Algorithmic Development

QMP is an adapted version of the probabilistic roadmap planner (PRM) for quotient space decompositions. We first summarize the workings of PRM, then show how QMP can be built from it.

A simplified version of PRM is depicted in Algorithm 1. While a planner terminate condition (PTC)² has not been reached (Line 2), the algorithm grows a graph \mathbf{G} on the configuration space \mathcal{M} of robot \mathcal{R} (Line 3). If there exists a path on the graph between start and goal configuration (Line 4), then this path is returned (Line 5). If the PTC is reached then PRM fails and returns an empty path.

The growing of the graph is depicted in Algorithm 2. A configuration q_{rand} is sampled from the configuration space \mathcal{M} (Line 1). If this configuration is valid (Line 2), then it is added to the graph (Line 3) and the nearest R configurations Q_R are searched (Line 4). The graph is extended in a straight line from each $q_{near} \in Q_R$ (Line 5) towards q_{rand} until it hits an obstacle (Line 6). The last configuration before the obstacle becomes q_{new} , and the edge between q_{near} and q_{new} is added to the graph (Line 7).

QMP is depicted in Algorithm 3. An empty priority queue is constructed (Line 1), and the k -th quotient roadmap is initialized (Line 3) and added to the queue (line 4). While no path between start and goal has been found (Line 5), we pop the quotient roadmap with the smallest density from the queue (line 6), grow its graph (line 7) and push it back onto the queue (line 8). Then we check if there exists a path on the current quotient space (line 9); if yes, we construct its solution (line 10), and we continue to the next quotient roadmap. For $k = 1$ the algorithm is equivalent to PRM. For $k > 1$, multiple quotient spaces are inside the queue,

²A planner terminate condition (PTC) has to be chosen by a user and can be a time limit, an iterations limit, or a desired cost of the resulting path.

and depending on the density function we pop one quotient space and grow its graph. The algorithm terminates if either the path \mathbf{p}_K has been found, or if the PTC is reached, in which case $\mathbf{p}_K = \emptyset$ is returned.

The growing of the quotient space graph is depicted in Algorithm 5. Instead of sampling \mathcal{M}_k as in the PRM, we sample instead \mathbf{C}_k uniformly, and we pick one configuration from the graph \mathbf{G}_{k-1} on \mathcal{M}_{k-1} (Line 1). The `SampleGraph` samples a uniform vertex from the graph \mathbf{G}_{k-1} . Then a random incoming edge is chosen, and a configuration uniformly on the edge is sampled. This is called Random-Vertex-Edge (RVE) sampling [18]. The cartesian product \circ merges the two configuration to yield a configuration on \mathcal{M}_k . The rest of the algorithm (line 2-7) operates as the PRM algorithm, with two exceptions. First, the R-Nearest-Neighbors method measures distance not by euclidean distance on \mathcal{M}_k , but by the graph distance on \mathbf{G}_{k-1} plus euclidean distance on \mathbf{C}_k (Line 4). Second, the `Connect` method does not interpolate along a straight line, but interpolates along the edges of the graph \mathbf{G}_{k-1} , while interpolating on \mathbf{C}^k using a straight line. For each vertex crossed on \mathbf{G}_{k-1} we add another configuration. The `Connect` method then returns a piece-wise linear (PL) path on \mathcal{M}_k . For each edge along this PL-path we add one edge to the graph \mathbf{G}_k (Line 7).

Interpolating along the graph instead of using a straight line should be seen as a change in the metric on \mathcal{M}_k . While a standard euclidean metric is agnostic about the graph on \mathcal{M}_{k-1} , our graph interpolation metric utilizes the knowledge about \mathbf{G}_{k-1} to improve the metric computation.

Algorithm 1 PRM(q_I, q_G, \mathcal{M})[19]

```

1: INIT( $\mathbf{G}, q_I$ )
2: while  $\neg$ PTC do
3:   GROWPRM( $\mathbf{G}, \mathcal{M}$ )
4:   if ISCONNECTED( $q_I, q_G, \mathbf{G}$ ) then
5:     return PATH( $q_I, q_G, \mathbf{G}$ )
6: return  $\emptyset$ 

```

Algorithm 2 GrowPRM(\mathbf{G}, \mathcal{M})

```

1:  $q_{rand} \leftarrow$  SAMPLE( $\mathcal{M}$ )
2: if  $\neg$ ISVALID( $q_{rand}$ ) then return
3: ADD_VERTEX( $q_{rand}, \mathbf{G}$ )
4:  $Q_R \leftarrow$  R-NEARESTNEIGHBORS( $q_{rand}, \mathbf{G}$ )
5: for  $q_{near} \in Q_R$  do
6:    $q_{new} \leftarrow$  CONNECT( $q_{near}, q_{rand}$ )
7:   ADD_EDGE( $q_{near}, q_{new}, \mathbf{G}$ )

```

C. Implementation Details

Our software uses the `Klamp't` [20] physical simulator, and the open motion planning library `OMPL` [21].

The nesting of robots has to be prespecified as a set of Unified Robot Description Format (URDF) files along with its subspaces. Each subspace is represented by an `OMPL` space, and our algorithm iterates through

Algorithm 3 QMP($q_{1,\dots,K}^I, q_{1,\dots,K}^G, M_1, \dots, M_K$)

```
1:  $Q \leftarrow \text{PRIORITY\_QUEUE}$ 
2: for  $k = 1$  to  $K$  do
3:    $\mathbf{Q}_k = \text{INIT}(q_k^I, q_k^G, M_k, \mathbf{Q}_{k-1})$ 
4:    $Q.\text{PUSH}(\mathbf{Q}_k)$ 
5:   while  $\mathbf{p}^k == \emptyset$  and  $\neg \text{PTC}$  do
6:      $\mathbf{Q}_{\text{least}} = Q.\text{POP}$ 
7:      $\text{GROWQMP}(\mathbf{Q}_{\text{least}})$ 
8:      $Q.\text{PUSH}(\mathbf{Q}_{\text{least}})$ 
9:     if  $\text{ISCONNECTED}(\mathbf{Q}_k)$  then
10:       $\mathbf{p}_k = \text{PATH}(q_k^I, q_k^G, \mathbf{Q}_k)$ 
11: return  $\mathbf{p}_K$ 
```

Algorithm 4 Init($q_k^I, q_k^G, M_k, \mathbf{Q}_{k-1}$)

```
1: return  $\{q_k^I, q_k^G, M_k / M_{k-1}, M_k, \emptyset, \emptyset, 0, \mathbf{Q}_{k-1}\}$ 
```

them computing the quotient spaces. We currently support the following quotient space computations: $\mathbb{R}^3 = \text{SE}(3) / \text{SO}(3)$, $\mathbb{R}^2 = \text{SE}(2) / \text{SO}(2)$, $\mathbb{R}^{n-m} = \mathbb{R}^n / \mathbb{R}^m$, $\text{SE}(3) = \text{SE}(3) \times \mathbb{R}^n / \mathbb{R}^n$, and $\text{SE}(3) \times \mathbb{R}^{n-m} = \text{SE}(3) \times \mathbb{R}^n / \text{SE}(3) \times \mathbb{R}^m$, with $n, m \in \mathbb{N}$ and $n > m > 0$.

Our algorithm terminates after a path has been found on the configuration space, or a timelimit T has been reached.

V. PROBABILISTIC COMPLETENESS

A motion planning algorithm is probabilistically complete if the probability that the algorithm will find a path if one exists approaches one as the number of sampled points increases. We will show that QMP is probabilistically complete by alluring to the probabilistic completeness of PRM [22].

The main difference of QMP and the PRM on the configuration space is the choice of a sampling sequence. For PRM the sampling sequence is dense in the configuration space, while it is not true for the sampling sequence generated by QMP. However, we will show that the sampling sequence by QMP is dense in the space of feasible configurations, thus making QMP probabilistically complete.

Let \mathcal{M} be the configuration space and $U \subseteq \mathcal{M}$ be a subset. We say that U is dense in \mathcal{M} if $\text{cl}(U) = \mathcal{M}$ whereby cl is the closure of U . Let $\alpha : \mathbb{N} \rightarrow \mathcal{M}$ be a random sequence. We say that the sequence α is dense in \mathcal{M} if the set $\{\alpha_n, n \in \mathbb{N}\}$ is dense in \mathcal{M} [1].

Algorithm 5 GrowQMP(\mathbf{Q}_k)

```
1:  $q_{\text{rand}} \leftarrow \text{SAMPLEGRAPH}(\mathbf{G}_{k-1}) \circ \text{SAMPLE}(\mathbf{C}_k)$ 
2: if  $\neg \text{ISVALID}(q_{\text{rand}})$  then return
3:  $\text{ADD\_VERTEX}(q_{\text{rand}}, \mathbf{G})$ 
4:  $Q_R \leftarrow \text{R-NEARESTNEIGHBORS}(q_{\text{rand}}, \mathbf{G}_k, \mathbf{G}_{k-1})$ 
5: for  $q_{\text{near}} \in Q_R$  do
6:    $\{q_{n_1}, \dots, q_{n_J}\} \leftarrow \text{CONNECT}(q_{\text{near}}, q_{\text{rand}}, \mathbf{G}_{k-1})$ 
7:    $\text{ADD\_EDGES}(q_{n_1}, \dots, q_{n_J}, \mathbf{G}_k)$ 
```

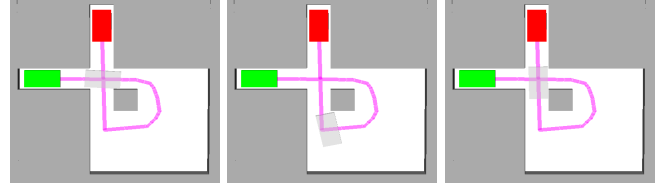


Fig. 4. Non-simple path on Quotient-Space: See text for clarification.

Each quotient space $\mathcal{M}_1, \dots, \mathcal{M}_K$ uses a sampling sequence as $\alpha^{(k)} : \mathbb{N} \rightarrow \mathcal{M}_k$. We will show that this sequence is dense in $\tilde{\mathcal{M}}_k = \{q \in \mathcal{M}_k | \mathbf{V}(q) \cap \mathbf{E} = \emptyset\}$, the feasible space of \mathcal{M}_k .

Theorem 2. α^K is dense in $\tilde{\mathcal{M}}_K$

Proof. By induction for $K = 1$ $\alpha^{(1)}$ is dense in \mathcal{M}_1 , and since $\tilde{\mathcal{M}}_1 \subseteq \mathcal{M}_1$, $\alpha^{(1)}$ is dense in $\tilde{\mathcal{M}}_1$. For $K > 1$, we assume $\alpha^{(K-1)}$ is dense in $\tilde{\mathcal{M}}_{K-1}$. Consider the sampling domain of $\alpha^{(K)}$, defined as $S = \{\alpha_n^{(K-1)} \times \mathbf{C}_K | \alpha_n^{(K-1)} \in \tilde{\mathcal{M}}_{K-1}, n \in \mathbb{N}\}$. Due to the necessary condition of nested robots (Theorem 1), we have that if $r \circ p \in \tilde{\mathcal{M}}_K$ then $r \in \tilde{\mathcal{M}}_{K-1}$. Thus $\tilde{\mathcal{M}}_K \subseteq S$. Since $\alpha^{(K)}$ is dense in S by definition, $\alpha^{(K)}$ is dense in $\tilde{\mathcal{M}}_K$. \square

Finally, we need to make sure that every $\alpha^{(k)}$ is an infinite sequence. This is only guaranteed if each quotient space from the priority queue in Algorithm 3 is chosen infinitely many times. A sufficient condition is to choose a density function strictly monotonically increasing, as we did in Eq. 4.

VI. QUOTIENT-SPACE HEURISTICS

We first discuss intricacies of quotient-spaces, and then use this knowledge to design heuristics for QMP.

A. Quotient-Space Intricacies

Solution paths on quotient-spaces do not behave as nicely as one would expect. We show two examples where the solution path is non-simple or spurious.

1) *Non-simple Paths on Quotient-Space:* A non-simple path is a path with self-intersections. We say that a path $\mathbf{p} : [0, 1] \rightarrow \mathcal{M}$ is simple, if it is injective, i.e. for any $t, s \in [0, 1]$: if $\mathbf{p}(t) = \mathbf{p}(s)$ then $t = s$ [23]. A path not being simple is called non-simple.

While non-simple paths do not occur in configuration spaces, they can occur in quotient spaces. Fig. 4 shows an example, where a rectangular shaped planar robot needs to move from a start configuration (green) to a goal configuration (red). The configuration space is $\text{SE}(2)$ and the quotient space is $\mathbb{R}^2 = \text{SE}(2) / \text{SO}(2)$, obtained by nesting a disk inside the rectangle. The solution path is simple on $\text{SE}(2)$, but non-simple on \mathbb{R}^2 .

2) *Spurious Shortest Path on Quotient-Space:* Spurious paths are solution paths on the quotient-space which are infeasible on the configuration space. Let \mathcal{M}_1 be a quotient space of the configuration space \mathcal{M}_2 and let $\mathbf{p}^1 : [0, 1] \rightarrow \mathcal{M}_1$ be the shortest feasible path on the quotient space between q_I^1 and q_G^1 . We say that \mathbf{p}^1 is feasible on the configuration space, if there exists a feasible path $\mathbf{p}^2 :$

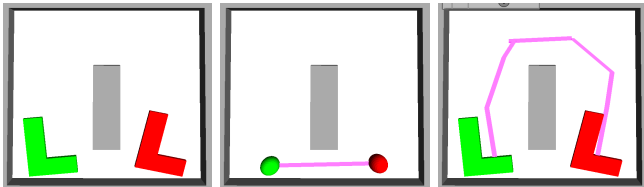


Fig. 5. Spurious shortest path in Quotient-Space: An environment for an L-shaped robot with configuration space $SE(2)$ and quotient space \mathbb{R}^2 . See text for clarification.

$[0, 1] \rightarrow \mathbf{p}^1 \times \mathbf{C}_2$. If \mathbf{p}^1 is infeasible on the configuration space, we call it spurious.

Consider the example in Fig. 5, where an L-shaped robot needs to move through an environment with two passages above and below a rectangular obstacle (Left). The quotient-space is represented by a disk nested in the L-shaped robot. The shortest path on the quotient space is spurious (Middle), since the only feasible path goes above the obstacle (Right).

B. Heuristics

Keeping the intricacies in mind, we design three heuristics which we found to be beneficial to reduce the runtime of QMP.

1) *Diminishing-Time Shortest Path Sampling*: If the shortest path is spurious this often can be established quickly. We add a diminishing time heuristic to sample first the shortest path on the quotient-space with probability $p = 0.8$ and decrease this probability over time with the amount of samples taken from the shortest path.

2) *Graph Thickening*: Often a solution path on the quotient-space is spurious, but a solution path nearby is not spurious and contains a solution on the configuration space. To alleviate this problem, we introduce an ϵ -graph thickening. Given a random sample q from the graph, we add an offset uniformly distributed from a ball around q with radius ϵ . This helps in finding nearby solutions. We have set $\epsilon = 0.01$ for free-floating robots and $\epsilon = 0.1$ for fixed-base robots.

3) *Increasing Clearance*: In some cases it is advantageous to inflate the shape of the nested robot such that the clearance of the solution path on the quotient space is increased. One has to be careful with this heuristic, since it trades-off completeness with efficiency. We apply this only for the free-floating robots in our experiments by increasing the size of the inscribed sphere by $\delta = 1.2$.

VII. EXPERIMENTS

We perform experiments on four different environments as shown in Fig. 8: A free-floating rigid body, a free-floating articulated body, a fixed-base serial chain and a fixed-base tree chain robot. The problem is to find a path for each robot from the initial configuration (green) to the goal configuration (red). All experiments have been performed on a quad core 2.6 Ghz laptop with 31 GB working memory. We perform collision detection using the proximity query package (PQP) [24].

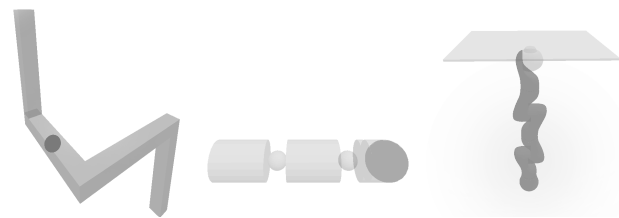


Fig. 6. Nesting of robots for the Double L-shape, Snake and KUKA LWR with windshield. The darker shade is the nested robot.



Fig. 7. Nesting of robots for Baxter Robot by Rethinking Robotics.

A. Free-floating rigid body

The double L-shape is a rigid body consisting of two L-shapes glued together. The environment is a wall with a small square hole in it, as depicted in Fig. 8. This problem was introduced by [13] as an example of a narrow passage planning problem. We decompose the double L-shape as $\mathbb{R}^3 \subset SE(3)$, as depicted in Fig. 6. We compared our algorithm QMP with three state of the art algorithms implemented in the OMPL software framework: PRM [25], bidirectional rapidly-exploring random tree (RRTConnect) [26] and the expansive space trees planner (EST) [27]. Fig. 9 shows that only QMP and PRM solved all runs, and that QMP has a median runtime of 2.5s, compared to 27s for PRM.

B. Free-floating articulated body

The mechanical snake is a 10-dof articulated body which has three links, connected each by two revolute joints with limits (See Fig. 6). The snake can freely translate and rotate in space. We found the most effective decomposition to be $\mathbb{R}^3 \subset SE(3) \times \mathbb{R}^4$. QMP has a median runtime of 30s (67s for EST). However, we can see that QMP has two outliers.

C. Fixed-base serial chain: KUKA LWR

The KUKA Lightweight Robot (LWR) is a 7-dof manipulator. We consider transporting a windshield through a factory simulating a car manufacturing task. Our decomposition is $\mathbb{R}^5 \subset \mathbb{R}^7$. In the experiments, only QMP was able to solve all runs, with a median time of 18s (164s for EST).

D. Fixed-base tree chain: Baxter

Baxter is a two-arm fixed-base robot with a tree kinematic chain having two serial chains of 7 dof for each arm, and having a total of 14 dof. We consider a maintenance task, where Baxter needs to move its arms into small openings of a defect water tank. We decompose Baxter as $\mathbb{R}^5 \subset \mathbb{R}^7 \subset \mathbb{R}^{12} \subset \mathbb{R}^{14}$ as depicted in Fig. 7. Only QMP was able to solve all runs, with a median runtime of 4.5s compared to 94s for RRTConnect.

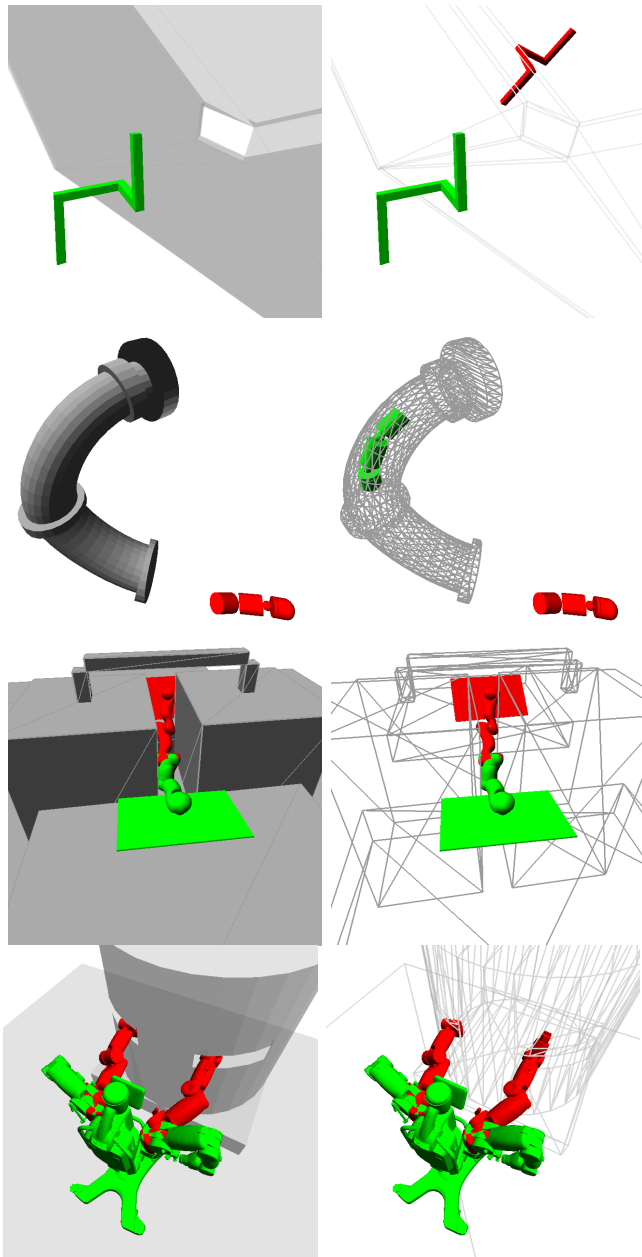


Fig. 8. The four experiments considered in this paper: For each experiment we show the start configuration of the robot (green) and the goal configuration (red). The left column shows the faces of the environment mesh, while the right column shows the edges of the environment mesh.

VIII. CONCLUSION

We have introduced the quotient-space decomposition, a configuration space decomposition based on nested robots. To exploit this decomposition we developed the Quotient-space roadMap Planner (QMP). We showed that QMP is probabilistically complete and that QMP has lower runtime on four environments compared to state-of-the-art motion planning algorithms implemented in the OMPL software.

We like to extend QMP in three directions. First, the quotient-space decomposition of a configuration space is not unique, and has to be specified by a human operator. We like to investigate which decomposition is optimal, and automate

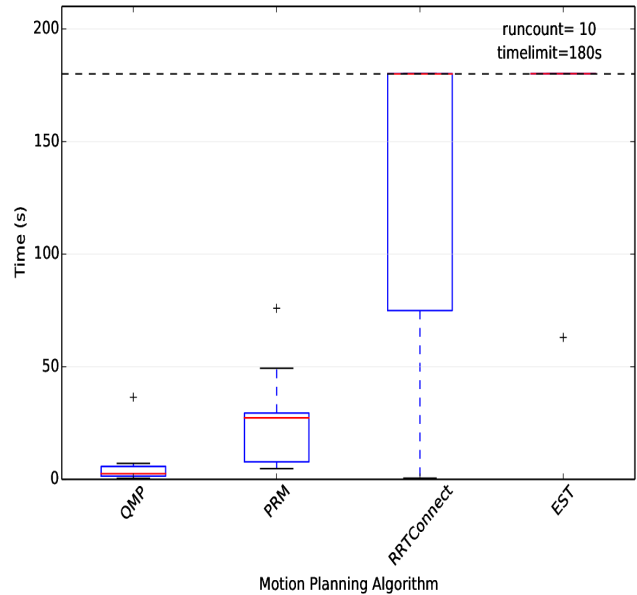


Fig. 9. Free-floating rigid body: Double L-Shape 6-dof benchmark.

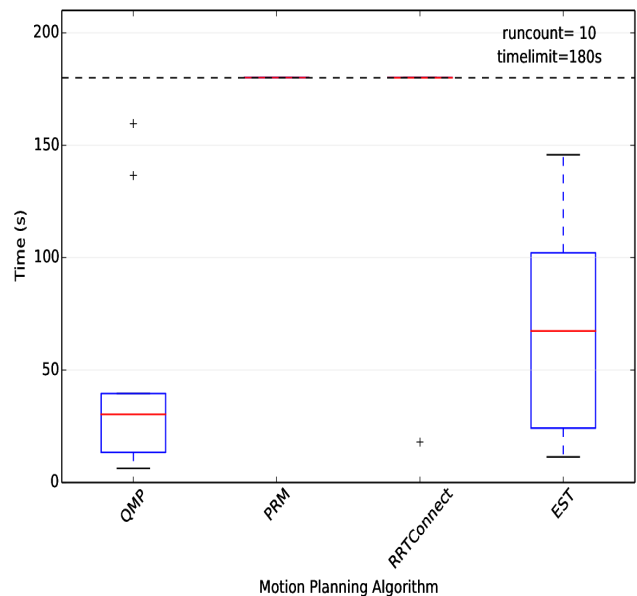


Fig. 10. Free-floating articulated body: Mechanical Snake 10-dof benchmark.

its specification. Second, we like to generalize our approach to closed kinematic chains, constraint manifolds, and dynamic constraints. Finally, we like to apply the quotient-space decomposition to environments changing over time, where a roadmap on the quotient space could improve fast decision making.

REFERENCES

- [1] S. M. LaValle. *Planning Algorithms*. Cambridge University Press, 2006.
- [2] John H Reif. Complexity of the mover's problem and generalizations. In *Conference on Foundations of Computer Science*, pages 421–427, 1979.
- [3] James Munkres. *Topology*. Pearson, 2000.
- [4] Ling Zhang and Bo Zhang. The quotient space theory of

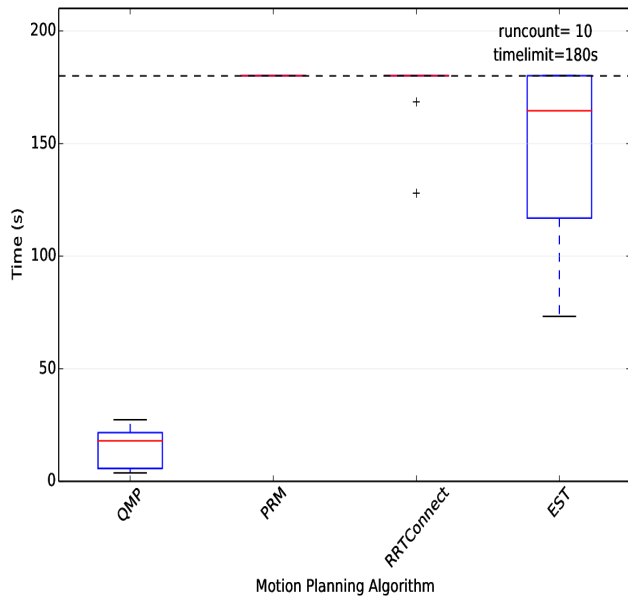


Fig. 11. Fixed-base serial chain: KUKA LWR 7-dof benchmark.

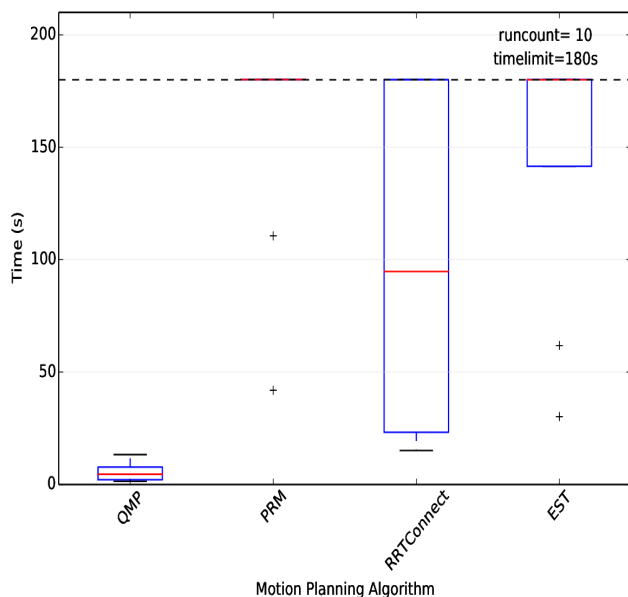


Fig. 12. Fixed-base tree chain: Baxter 14-dof benchmark.

problem solving. *Fundamenta Informaticae*, 59(2-3):287–298, 2004.

- [5] Jing Tao Yao, Athanasios V Vasilakos, and Witold Pedrycz. Granular computing: perspectives and challenges. *Transactions on Cybernetics*, 43(6):1977–1989, 2013.
- [6] Timothy Bretl. Motion planning of multi-limbed robots subject to equilibrium constraints: The free-climbing robot problem. *International Journal of Robotics Research*, 25(4):317–342, 2006.
- [7] Michael X Grey, Aaron D Ames, and C Karen Liu. Footstep and motion planning in semi-unstructured environments using randomized possibility graphs. In *IEEE International Conference on Robotics and Automation*, 2017.
- [8] S. Tonneau, A. D. Prete, J. Pettr, C. Park, D. Manocha, and N. Mansard. An Efficient Acyclic Contact Planner for Multipled Robots. *Transactions on Robotics*, 34(3):586–601, June 2018.
- [9] Liangjun Zhang, Jia Pan, and Dinesh Manocha. Motion

planning of human-like robots using constrained coordination. In *IEEE International Conference on Humanoid Robots*, 2009.

- [10] Markus Rickert, Arne Sieverling, and Oliver Brock. Balancing exploration and exploitation in sampling-based motion planning. *Transactions on Robotics*, 30(6):1305–1317, 2014.
- [11] Tobias Kunz, Andrea Thomaz, and Henrik Christensen. Hierarchical rejection sampling for informed kinodynamic planning in high-dimensional spaces. In *IEEE International Conference on Robotics and Automation*, pages 89–96. IEEE, 2016.
- [12] Jonathan D Gammell, Siddhartha S Srinivasa, and Timothy D Barfoot. Informed RRT*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic. In *IEEE International Conference on Intelligent Robots and Systems*, pages 2997–3004. IEEE, 2014.
- [13] Jur P Van den Berg and Mark H Overmars. Using workspace information as a guide to non-uniform sampling in probabilistic roadmap planners. *International Journal of Robotics Research*, 24(12):1055–1071, 2005.
- [14] Matt Zucker, James Kuffner, and J Andrew Bagnell. Adaptive workspace biasing for sampling-based planners. In *IEEE International Conference on Robotics and Automation*, pages 3757–3762. IEEE, 2008.
- [15] Andreas Orthey, Florent Lamiroux, and Olivier Stasse. Motion Planning and Irreducible Trajectories. In *IEEE International Conference on Robotics and Automation*, 2015.
- [16] Ioan A Şucan and Lydia E Kavraki. Kinodynamic motion planning by interior-exterior cell exploration. In *Algorithmic Foundation of Robotics VIII*. Springer, 2009.
- [17] John M. Lee. *Introduction to Smooth Manifolds*. Springer New York, New York, NY, 2003.
- [18] Jure Leskovec and Christos Faloutsos. Sampling from large graphs. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 631–636. ACM, 2006.
- [19] Steven M Lavalle and James J Kuffner Jr. Rapidly-Exploring Random Trees: Progress and Prospects. In *Algorithmic and Computational Robotics: New Directions*, 2000.
- [20] Kris Hauser. Robust contact generation for robot simulation with unstructured meshes. In *International Journal of Robotics Research*, pages 357–373. Springer, 2016.
- [21] Ioan A Şucan, Mark Moll, and Lydia Kavraki. The open motion planning library. *Robotics and Automation Magazine*, 2012.
- [22] Petr Svestka. *On probabilistic completeness and expected complexity for probabilistic path planning*, volume 1996. Utrecht University: Information and Computing Sciences, 1996.
- [23] Manfredo Perdigao Do Carmo. *Differential geometry of curves and surfaces*, volume 2. Prentice-hall Englewood Cliffs, 1976.
- [24] Eric Larsen, Stefan Gottschalk, Ming C Lin, and Dinesh Manocha. Fast proximity queries with swept sphere volumes. Technical report, Technical Report TR99-018, Department of Computer Science, University of North Carolina, 1999.
- [25] Lydia E Kavraki, Petr Svestka, J-C Latombe, and Mark H Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *Transactions on Robotics*, 12(4):566–580, 1996.
- [26] James J Kuffner and Steven M LaValle. RRT-connect: An efficient approach to single-query path planning. In *IEEE International Conference on Robotics and Automation*, volume 2, pages 995–1001. IEEE, 2000.
- [27] David Hsu, J-C Latombe, and Rajeev Motwani. Path planning in expansive configuration spaces. In *IEEE International Conference on Robotics and Automation*, volume 3, pages 2719–2726. IEEE, 1997.