**IEEE Robotics and Automation Letters (RAL) paper presented at the
2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)**

Motion planning for dual-arm manipulation of elastic rods

Avishai Sintov[1], Steven Macenski[2], Andy Borum[3] and Timothy Bretl[2]

Abstract— **We present a novel motion planning strategy for the manipulation of elastic rods with two robotic arms. In previous work, it has been shown that the free configuration space of an elastic rod, i.e., the set of equilibrium shapes of the rod, is a smooth manifold of a finite dimension that can be parameterized by one chart. Thus, a sampling-based planning algorithm is straightforward to implement in the product space of the joint angles and the equilibrium configuration space of the elastic rod. Preliminary results show that planning directly in this product space is feasible. However, solving for the elastic rod's shape requires the numerical solution of differential equations, resulting in an excessive and impractical runtime. Hence, we propose to pre-compute a descriptor of the rod, i.e., a roadmap in the free configuration space of the rod that captures its main-connectivity. By doing so, we can plan the motion of any dual-arm robotic system over this roadmap with dramatically fewer solutions of the differential equations. Experiments using the Open Motion Planning Library (OMPL) show significant runtime reduction by an order of magnitude.**

I. Introduction

Industrial manipulation of rigid objects has been automated for quite a long time, while the handling of deformable objects is usually done manually due to the lack of feasible motion planning algorithms. Indeed, motion planning algorithms have mainly focused on the manipulation of rigid bodies by one or more robots. Much less attention has been given to motion planning for the manipulation of deformable objects in general, and elastic rods in particular. The term *elastic rod* refers to thin deformable wires or cables. Hence, the ability of robots to manipulate elastic rods would be practical in many applications. For instance, cable routing is one of the few operations done manually in automotive production lines [1]. Other applications include surgical suturing [2], knot tying [3], and automated handling of flexible material such as sheet metal parts [4].

In this paper, we focus on path planning for two robotic arms to manipulate an elastic rod (Figure 1). The rod is required to move between start and goal configurations while remaining in static equilibrium and avoiding self and obstacle collisions. We rely on previous work by Bretl and McCarthy [5] who showed that the configuration space of a Kirchhoff elastic rod [6], i.e., the set of all equilibrium configurations, is a six-dimensional smooth manifold parameterized by a set $\mathcal{A} \subset \mathbb{R}^6$. Given a point in $\mathcal{A}$, it can be used as the initial condition for solving a system of ordinary differential equations to find the corresponding shape of the rod. Bretl and McCarthy also provided a computational test to distinguish between stable and unstable equilibrium configurations.

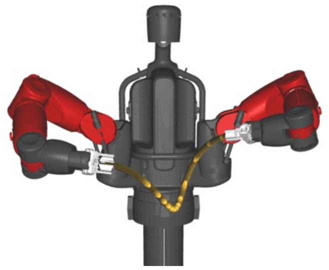

Fig. 1. Illustration of the Baxter robot manipulating an elastic rod.

A sampling-based algorithm for solely planning in $\mathcal{A}$ is now simple to implement. However, practical applications require planning of the robot's motion to manipulate the rod from one configuration to another. In other words, the configuration space of the system should incorporate the robot's joint space and the rod's configuration space $\mathcal{A}$. However, planning in this extended configuration space presents more challenges. A feasible path must satisfy a set of constraints, i.e., satisfy joint limits, ensure stability of the rod and be collision-free. In addition, grasping a rod in a desired configuration by its endpoints with two robotic arms requires that a closed kinematic chain constraint be satisfied. The configuration space of the closed chain system is a lower-dimensional subset of the ambient space satisfying the closure constraint [7]. The sampling-based strategy proposed in this work finds paths that lie strictly on the closure constraint while satisfying joint limits and are collision-free.

In this work, we discuss an approach to planing a feasible motion of the system using standard sampling-based planners. However, we will show that the use of standard planners results in a large amount of time spent in solving the differential equations describing the shape of rod and therefore have high computational costs. To tackle this problem, we propose a pre-computation strategy where a roadmap in $\mathcal{A}$ is stored in memory along with the solutions for the shape of the rod. This roadmap encodes the main connectivity in the stable and collision-free subspace of $\mathcal{A}$. Thus, we leave the expensive computations to an offline step, thereby saving a significant amount of time in the process of planning the motion of any dual-arm robotic system over the roadmap. To construct the roadmap and, more importantly, to rapidly connect to it during planning, we use the scale invariance property previously presented by Borum and Bretl [8]. Borum and Bretl have shown that the free configuration space of the rod is path connected. That is, a stable analytical path in $\mathcal{A}$ can easily be computed between any two stable configurations. We extend this notion and show that there are two-dimensional submanifolds in $\mathcal{A}$, termed *slices*, formed

[1] A. Sintov is with the School of Mechanical Engineering, Tel-Aviv University, Israel. e-mail: sintov1@tauex.tau.ac.il

[2] S. Macenski and T. Bretl are with the Department of Aerospace Engineering, University of Illinois at Urbana-Champaign, Urbana, IL, 61801, USA. e-mail: tbretl@illinois.edu.

[3] A. Borum is with the Department of Mathematics, Cornell University, NY, USA. e-mail: adb328@cornell.edu

by any two stable configurations and the origin, where any path in the slice is stable and free of self-collisions. With this notion, we can obtain an upper bound on the number of numerical solutions of differential equations that must be computed. Therefore, paths in these slices can rapidly connect the start and goal configurations to the roadmap.

There are several contributions in our work. Previous work on the manipulation of rods has focused solely on the path planning of the elastic object. To the best of our knowledge, our work is the first attempt to incorporate both the configuration space $\mathcal{A}$ of the rod and the configuration space of robotic arms in a path planning problem. Thus, we propose a paradigm for planning the manipulation of an elastic rod with two robots based on [5]. The paradigm is based on the usage of standard sampling-based motion planning algorithms over a pre-computed roadmap of the rod. The structure of the roadmap depends only on the properties of the rod, and is entirely independent of the robotic system used to manipulate it. Thus, the roadmap solely describes the rod and can be used to plan the motion of any dual-arm robotic system. While some planning algorithms pre-compute partial roadmaps [9], to the best of the authors' knowledge, the notion of planning for different robotic systems over a precomputed roadmap of another system has not been applied before. Finally, we propose a strategy to employ a single-query planning algorithm in the configuration space of the robotic arms over the roadmap.

II. Related work

The manipulation of deformable objects by two or more robots has been extensively studied in the past few decades [10]–[12]. In this work, we address the manipulation of thin elastic rods, which have been of interest for centuries, beginning with the study of planar rods by Euler and Bernoulli and non-planar rods by Kirchhoff [13]. The configuration space directly describing the shape of an elastic rod has infinite dimension, where elements of this space are continuous maps. In addition, a number of possible configurations of the rod exist for a given placement of the grippers holding the ends of the rod. These challenges have made the motion planning of a rod a hard problem. The literature on path planning for elastic rods suggests exploring the set of equilibrium configurations indirectly, by sampling displacements of grippers and using numerical simulations to approximate their effect on the rod. This approach was developed in the work of Lamiraux and Kavraki [14] and was applied by Moll and Kavraki [15] to the manipulation of deformable linear objects. Hermansson et al. [16] relaxed gripping point constraints along an elastic harness while planning a collision-free path for a sphere around a predefined central grip point. These approaches use computationally expensive numerical methods that may limit their effectiveness in real-time motion planning. Another approach used in [17] simplifies the model of the deformed object by reducing it to a sequence of rigid masses and springs. In this approach, the solution is highly sensitive to the approximation, which in turn affects the quality of the planning.

In all the cited approaches, a feasible procedure to derive the free configuration space was not clear at that time. Bretl and McCarthy [5] later showed that the configuration space of the rod, i.e., the set of all equilibrium configurations, is a six-dimensional smooth manifold. They also provided a computational test to determine whether an equilibrium configuration on the manifold is stable or not, and a collision checking algorithm was used to find self-intersections. This allowed for a sampling-based planning algorithm to be used in which configurations of the rod could be sampled directly. Further research following the work of Bretl and McCarthy was published in [18] exploring additional planning properties in the free configuration space of the rod.

Bretl and McCarthy did not provide an insight into the topological properties of the free configuration space or guarantee that a feasible path from start to goal does exist. Moreover, the determination of whether an equilibrium configuration is a member of the free configuration space (i.e., is a stable configuration and free of self-collisions) is rather computationally expensive. Hence, the later work of Borum and Bretl [8] has provided the important insight that the free configuration space is path-connected, and a semi-analytical feasible path can easily be found. The checking of stability for a configuration in $\mathcal{A}$ burdens sampling-based planning algorithms, and the notion guaranteeing a feasible path in $\mathcal{A}$ can reduce its extensive computational complexity. However, the assured presence of a path is only viable while solely observing the configuration of the rod. Extending the problem for the planning of robotic arms manipulating the rod makes it more challenging.

Sampling-based planners are a commonly used in robotics for motion planning in high dimensional configuration spaces [19]. Seminal work by Kavraki et al. [20] proposed a sampling-based multi-query approach where a probabilistic roadmap is computed off-line. Then, the cost of online planning lies only in graph search. A practical application of pre-computed roadmaps was presented in [21] for motion planning of concentric tube robots navigating anatomical obstacles. The pre-computed roadmap loses its relevance when the environment changes and must be modified. Extensions were made to address these issues by maintaining and making efficient repairs based on recent changes in the environment [22]. In the work of Salzman et al. [23], which has similarities to ours, a self-collision-free roadmap of a multi-link system is pre-computed in a pre-processing step. Thus, during online planning, collisions are only checked with the extrinsic workspace obstacles. This approach makes the roadmap independent of the robot's environment. The same notion is the base of our work to employ an independent roadmap that describes the configuration space of the rod for any dual-arm system.

In our work, we pre-compute a roadmap for a lower-dimensional subset of the configuration space of the system, i.e., the roadmap only describes the configuration space of the rod, which can be coupled to any configuration space of two robotic arms. Some planning algorithms decouple the system to different components of motions [24], [25] naturally existing in many systems. An early implementation of this approach planed the motion of a manipulator while decoupling the motion of the first three links from the wrist [26]. Planning of multiple robots' motion can be done individually by computing a path for each robot and applying velocity tuning between paths for synchronized collision-free motion [27]. More recently, the work in [9] pre-computed dynamic roadmaps for the legs of a quadruped robot, which were used online to rapidly identify collision-free foothold positions.

III. Problem definition

In this section, we provide a formal definition of the path planning problem. We first describe a model of the elastic rod. While a complete derivation of this model is given in [5], we outline the key components of the model that are needed in later sections. We then describe the problem of path planning for both the rod and robots holding the rod.

A. The configuration space of an elastic rod

In this section, we present the necessary background from Bretl and McCarthy [5]. Their work showed that each equilibrium configuration of a Kirchhoff elastic rod [6] corresponds to a unique point in a subset of $\mathbb{R}^6$. Given the differential equations governing equilibrium configurations of the rod, they were able to map points in $\mathbb{R}^6$ to rod configurations and provide a computational criterion to determine their stability. We assume that the stiffness of the rod is high enough so that the effects of gravity can be neglected.

We assume that the rod is straight in the undeformed configuration and has length L. Using $t \in [0, L]$ to denote arc-length along the rod, the position and orientation of the rod at arc-length t are described by an element $q(t)$ of the special Euclidean group $SE(3)$. The rod's shape is described by a continuous map $q\colon [0, L] \to SE(3)$. In the Kirchhoff elastic rod model, the rod is allowed to twist and bend, but is unshearable and inextensible [6]. These constraints are enforced by requiring q to satisfy the differential equation

$$\dot{q} = q \left(\begin{smallmatrix} \widehat{u} & e_1 \\ 0 & 0 \end{smallmatrix} \right), \tag{1}$$

for some function $u\colon [0, L] \to \mathbb{R}^3$, where overdots denote differentiation with respect to t, the map $\widehat{\cdot}\colon \mathbb{R}^3 \to \mathfrak{so}(3)$ satisfies $a \times b = \widehat{a}b$ for all $a, b \in \mathbb{R}^3$, and $e_1 = [1\ 0\ 0]^T$. The function u_1 is the twisting strain and the functions u_2 and u_3 are the bending strains along the rod.

Each end of the rod is held by a robotic gripper. For now, we will assume, without loss of generality, that the base of the rod is held fixed at $q(0) = I$, where I is the identity element of $SE(3)$. This provides an initial condition for the differential equation (1). Since we neglect the effect of gravity, the resulting configuration of the rod can be translated and rotated to the pose of the gripper at $t = 0$. The gripper holding the rod at $t = L$ is free to move in $SE(3)$.

As shown in [5], the function u must satisfy certain conditions for the corresponding shape of the rod to be in equilibrium. To describe these conditions, we first define

$$\mathcal{A} = \{a \in \mathbb{R}^6 : (a_2, a_3, a_5, a_6) \neq (0, 0, 0, 0)\}. \tag{2}$$

$\mathcal{A}$ is simply $\mathbb{R}^6$ with a two-dimensional plane removed. Next, given $a \in \mathcal{A}$, solve the system of differential equations

$$\begin{aligned} \dot{\mu}_1 &= u_3\mu_2 - u_2\mu_3 & \dot{\mu}_4 &= u_3\mu_5 - u_2\mu_6 \\ \dot{\mu}_2 &= \mu_6 + u_1\mu_3 - u_3\mu_1 & \dot{\mu}_5 &= u_1\mu_6 - u_3\mu_4 \\ \dot{\mu}_3 &= -\mu_5 + u_2\mu_1 - u_1\mu_2 & \dot{\mu}_6 &= u_2\mu_4 - u_1\mu_5 \end{aligned} \tag{3}$$

for $t \in [0, L]$ with $u_i = c_i^{-1}\mu_i$, $i \in \{1, 2, 3\}$, and the initial condition $\mu(0) = a$. The constants $c_1 > 0$ and $c_2, c_3 > 0$ are, respectively, the torsional and bending stiffnesses of the rod. The function $\mu\colon [0, L] \to \mathbb{R}^6$ can be interpreted as the vector of internal forces and torques along the rod. Solving (1) with $u_i = c_i^{-1}\mu_i$ produces an equilibrium shape of the rod.

Based on the procedure described above (and as proved in Theorem 5 of Bretl and McCarthy [5]), each point in $\mathcal{A}$ corresponds to an equilibrium configuration of the rod. We denote such equilibrium configurations by the pair of functions (q, u). Each (q, u) and the corresponding μ are completely defined by the choice of $a \in \mathcal{A}$. Denote the resulting map by $(q, u) = \Psi(a)$ and define $\mathcal{C} = \Psi(\mathcal{A})$. In Theorem 5 of [5], it is shown that the map Ψ is injective, i.e. for each $(q, u) \in \mathcal{C}$ there exists a unique $a \in \mathcal{A}$.

We next describe a test to determine which equilibrium configurations of the rod are stable. A configuration (q, u) is a stable equilibrium configuration if $\det(\mathbf{J}(t)) \neq 0$ for all $t \in (0, L]$, where the matrix $\mathbf{J}(t)$ is acquired by solving the linear arc-length-varying matrix differential equations

$$\dot{\mathbf{M}} = \mathbf{F}(\mu(t))\mathbf{M} \qquad\qquad \dot{\mathbf{J}} = \mathbf{G}\mathbf{M} + \mathbf{H}(\mu(t))\mathbf{J} \tag{4}$$

with initial conditions $(\mathbf{M}(0), \mathbf{J}(0)) = (I, 0)$. The definitions of $\mathbf{G}$, $\mathbf{F}(\cdot)$ and $\mathbf{H}(\cdot)$ along with a full proof are given in [5]. Here also, the matrix functions $\mathbf{M}$ and $\mathbf{J}$ are completely determined by the choice of $a \in \mathcal{A}$. A point $t \in (0, L]$ at which $\det(\mathbf{J}(t)) = 0$ is called a *conjugate* point.

The above test allows us to determine which equilibrium configurations of the rod are stable. Denote the set of all $a \in \mathcal{A}$ that correspond to stable equilibrium configurations by $\mathcal{A}_{stable}$, and let $\mathcal{C}_{stable} = \Psi(\mathcal{A}_{stable})$. Also, define the map $\Phi : \mathcal{C} \to SE(3)$ such that a configuration (q, u) is mapped to $q(L)$. Given a path of the rod in $\mathcal{C}_{stable}$, the function Φ can be used to find the path of the robotic gripper that causes the rod to follow the path in $\mathcal{C}_{stable}$. In particular, letting $\mathcal{B}$ denote the space of boundary conditions, the map $\Phi \circ \Psi : \mathcal{A}_{stable} \to \mathcal{B}_{stable}$ takes $a \in \mathcal{A}_{stable}$ to the required pose of the gripper $b = q(L) \in \mathcal{B}_{stable}$. Finally, we define the free configuration space $\mathcal{A}_{free} \subset \mathcal{A}_{stable}$ to be the set of all $a \in \mathcal{A}$ that correspond to stable equilibrium configurations of the rod that do not contain self-intersections.

Planning in $\mathcal{A}$ is beneficial since a point in $\mathcal{A}$ uniquely defines an equilibrium configuration of the rod. On the other hand, many solutions exist for a given gripper pose $b \in \mathcal{B}_{\text{stable}}$. Hence, unlike planning in $\mathcal{A}$, planning in $\mathcal{B}$ requires knowledge of the initial rod pose and the entire path taken by the gripper. Another advantage of planning in $\mathcal{A}$ is that any two stable equilibrium configurations are path connected [8], a result that is used later in this work to improve the performance of the proposed planner.

B. The configuration space of the system

The elastic rod model derived in [5] and summarized in the previous section does not account for the configuration of the robots holding the rod's ends. Therefore, to generalize this model, we now consider a system comprised of two robotic arms holding an elastic rod. Let $\mathcal{Q}$ be the configuration space of the two arms formed by their joint space product, and $\mathcal{A}$ be the configuration space of the rod as defined in (2). Thus, the configuration space of the system is defined as $\mathcal{Z} = \mathcal{A} \times \mathcal{Q}$.

At any instant, a desired rod configuration can be treated as a rigid object. Hence, the two arms holding the rod by its endpoints impose a closed kinematic chain constraint $C(a, \phi) = 0$, where $\phi \in \mathcal{Q}$. Further, let $\mathcal{Z}_b \subset \mathcal{Z}$ be a restricted region due to obstacles, joint limits and the rod infeasible set $\mathcal{A} \setminus \mathcal{A}_{free}$. We define the free configuration set $\mathcal{Z}_o$ as follows:

Definition 1. *Let $\mathcal{Z}_o$ be the free configuration space of the system of the two arms and held rod such that*

$$\mathcal{Z}_o = \{(a, \phi) \in \mathcal{Z} : (a, \phi) \notin \mathcal{Z}_b,\ C(a, \phi) = 0\}\,. \quad (5)$$

That is, $\mathcal{Z}_o$ is the set of configurations that satisfy the closed kinematic chain, satisfy joint limits and are collision-free.

The motion planning problem is as follows. Given start and goal configurations $(a_s, \phi_s) \in \mathcal{Z}_o$ and $(a_g, \phi_g) \in \mathcal{Z}_o$ for the system, find a continues path $\tau : [0, 1] \to \mathcal{Z}_o$ such that $\tau(0) = (a_s, \phi_s)$ and $\tau(1) = (a_g, \phi_g)$.

C. Preliminary results

The main practical outcome of the configuration space of an elastic rod described in Section III-A is the ease of implementation of any sampling-based motion planning algorithm. Planning solely in $\mathcal{A}$ is technically straightforward. However, preliminary planning results (to be shown in the experimental section) for the system seen in Figure 4.*I* show poor performance with an average runtime of at least 70 seconds in our experiments. The main reason for the excessive runtime is that around 98% of the computation is spent on solving the ODE's in (1)-(4). An average computation of a single ODE takes approximately 70 milliseconds. Hence, sampling a random configuration is relatively computational expensive and slows down the exploration. Such excessive runtime is not practical, for example, in a real-time motion planning framework. In the next section, we propose an alternative planning strategy in which we approximate the free configuration space of the rod in an off-line pre-processing phase.

IV. Method

We propose to approximate $\mathcal{A}_{free}$ with a graph structure, i.e., a roadmap, $\mathcal{A}_G \subset \mathcal{A}_{free}$ (Figure 2). Such a roadmap would capture the main connectivity of $\mathcal{A}_{free}$ while enabling immediate knowledge of the rod's configuration corresponding to each node. In other words, the roadmap is pre-computed off-line such that, for every node, the configuration a of the rod and its solution $\Psi(a)$ are stored. In this way, any path in $\mathcal{A}_G$ can almost immediately be mapped to a path of the rod in $\mathcal{Z}$ with no additional solutions of the ODE's (1)-(4). Thus, querying in $\mathcal{A}_G$ rather than in $\mathcal{A}_{free}$ is more conducive for rapid motion planning. We next describe the algorithm for generating the roadmap followed by the planning process.

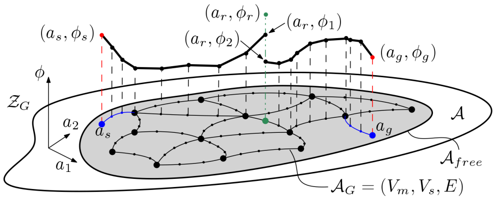

Fig. 2. Illustration of planning for the robot joints over the roadmap $\mathcal{A}_G$ of the rod. The roadmap $\mathcal{A}_G$ is an approximation of $\mathcal{A}_{free}$. Two branches of the two trees are expanded in $\mathcal{Z}_G$ toward a random configuration.

A. Generating a Roadmap

Let $\mathcal{A}_G = (V_m, V_s, E)$ be an undirected graph containing a set of m nodes or *milestones* V_m, a set of *edges* E connecting some pairs of milestones, and a set of *sub-milestones* V_s along the edges. Algorithm 1 describes a procedure to generate a roadmap $\mathcal{A}_G \subset \mathcal{A}_{free}$. We first generate the set V_m of random milestones in $\mathcal{A}_{free}$. Recall that membership in the set $\mathcal{A}_{free} \subset \mathbb{R}^6$ (which is checked in Line 5 of Algorithm 1) is evaluated by solving the differential equations (1)-(4) to determine if the corresponding configuration of the rod is stable and free of self-intersections. Edges are then added between each milestone and its k nearest neighbors. The addition of an edge is done using function get_edge(a_i, a_j), described later, and returns a set N_s of sub-milestones along a path connecting $a_i, a_j \in \mathcal{A}_{free}$. An edge $e = (a_i, a_j; N_s)$ is formed by connecting two nodes, a_i and a_j, and the sub-milestones N_s along the connection.

Algorithm 1 generate_roadmap$(m, k, L, \mathbf{c})$

```
 1: $V_m \leftarrow \emptyset,\ V_s \leftarrow \emptyset, E \leftarrow \emptyset$.
 2: for $i = 1 \to m$ do
 3:     repeat
 4:         $a \leftarrow rand()$
 5:     until rod_feasible$(a, L, \mathbf{c})$ {check if in $\mathcal{A}_{free}$}
 6:     $V_m \leftarrow V_m \cup \{(a, \Psi(a))\}$
 7: end for
 8: for $i = 1 \to m$ do
 9:     $N_k \leftarrow k$ nearest neighbors of $a_i \in V_m$.
10:     for all $a_j \in N_k$ do
11:         $N_s \leftarrow$ get_edge$(a_i, a_j, L, \mathbf{c})$.
12:         $E \leftarrow E \cup \{(a_i, a_j; N_s)\}$.
13:         $V_s \leftarrow V_s \cup N_s$.
14:     end for
15: end for
16: $H \leftarrow$ all_pairs_shortest_path$(\mathcal{A}_G)$.
17: return $\mathcal{A}_G = (V_m, V_s, E)$ and $H$.
```

For each milestone and sub-milestone a_i in the map $\mathcal{A}_G$, we store its solution $\Psi(a_i)$. Thus, V_m and V_s are sets of nodes in the form of $(a_i, \Psi(a_i))$. In practice, we require $\mathbf{b}_i$ (pose of the rod's end-tip when the other end-tip is in the origin) and a discrete set $P_i \subset \mathbb{R}^3$ of points along the rod for collision checks. Furthermore, fast on-line planning would require rapid querying of paths within $\mathcal{A}_G$. Therefore, we use the *Floyd-Warshall all-pairs shortest path* algorithm [28] to compute a graph reachability of the map. We construct and store a look-up table H enabling rapid querying of the shortest path between any two milestones in $\mathcal{A}_G$. Table H also enables easy knowledge of whether the roadmap is composed of one component or more.

It is important to note that the pre-computed roadmap is independent of any robotic system. That is, the roadmap can be used with any two robotic arms with the sole constraint of being able to grasp the rod by its end-tips for some configurations. Planning using the pre-computed roadmap for any dual-arm robotic system is presented next.

B. Planning on $\mathcal{A}_G$

Let $\mathcal{Z}_G \subset \mathcal{Z}$ be the product space of the joint space and the approximation roadmap, i.e., $\mathcal{Z}_G = \mathcal{A}_G \times \mathcal{Q}$. Rather than planning directly in $\mathcal{Z}$ and sampling computationally expensive rod configurations in $\mathcal{A}_{free}$, we propose to plan in $\mathcal{Z}_G$ subject to the constraints in (5). Furthermore, we address sampling-based motion planning where the standard Rapidly-exploring Random Tree [29] algorithm can be used. However, the nature of the problem provides better performance when using a Bi-Directional Rapidly-exploring Random Trees (Bi-RRT) [30] planning algorithm. Since our system is constrained by (5), the implemented Bi-RRT is, in fact, a slight modification of the Constrained Bi-RRT (CBi-RRT) proposed in [31]. Lazy planners have no advantage in planning over $\mathcal{A}_G$ since small steps are performed without local-connections to defer.

In any given planning query, the start and goal configurations of the rod will most likely not be included in the roadmap. Thus, the first step would be to connect them to the roadmap. The nearest milestones are identified and edges are added by using function get_edge. In this way, the start and goal rod configurations are fully included in the map. We note that this is the only step in which solving the ODE's (1)-(4) is required. Nevertheless, in the next section, we present a possible implementation of get_edge that can bound the computational runtime when adding an edge.

The basic principle of the planning algorithm is to randomly choose a node (milestone or sub-milestone) $a_r \in \mathcal{A}_G$ and generate a complementary random robot configuration $\phi_r \in \mathcal{Q}$. Then, two search trees anchored at the start and goal configurations, (a_s, ϕ_s) and (a_g, ϕ_g), are expanded toward the random configuration (a_r, ϕ_r) along the shortest path in $\mathcal{A}_G$ (found using H) with predefined joint steps. One tree is expanded, from the closest node in the tree, as much as possible toward (a_r, ϕ_r) (Figure 2). The second tree is then expanded toward the most advanced node of the first tree. The roles of the trees are swapped during iterations.

The probability for both trees to intersect with the same rod and joints configurations (a_r, ϕ_r) is low. In practice, the trees will most likely reach (if such path exists) the same rod configuration a_r but with different arms configurations. The arms hold the rod with the same shape but with different joint angles, i.e, the rod is at a different spatial pose in the workspace. Thus, the problem is now simplified to the manipulation of a rigid object from one robot configuration ϕ_1 to another ϕ_2 while maintaining constant rod configuration a_r and checking each for collisions. If the trees indeed reached the same rod configuration, we check for a local connection solely in $\mathcal{Q}$. If a rigid manipulation is possible then the trees are connected, resulting in a feasible path.

C. Edge addition

In this section, we address the strategy of connecting milestones in the roadmap and connecting the start and goal rod configurations to the roadmap using the function get_edge. We first consider the problem of connecting start and goal configurations to $\mathcal{A}_G$. One possibility is to search for an edge between two configurations in $\mathcal{A}_{free}$ by checking if configurations along a straight line or arbitrary curve, which connect them, are stable and free of self-collisions, which requires the solution of the ODEs (1)-(4). However, the number of ODE solutions in such a method is not bounded and may result in excessive computations. On-line connection of the start and goal configurations to the roadmap must be fast and therefore, require a minimal number of ODE computations.

In previous work [8], it was shown that the free configuration space $\mathcal{A}_{free}$ is path-connected. Based on a scale invariance property of the rod, for any two configurations in $\mathcal{A}_{free}$, a semi-analytical path can be constructed that is stable and free of self-intersections. The next theorem expands this notion and defines two-dimensional slices in $\mathcal{A}_{free}$.

Theorem 1. *Let $\sigma\colon [0,1] \to \mathcal{A}$. Define the function $t_1\colon [0,L] \to \mathbb{R}^+$ so that $t_1(s)$ is the first conjugate point along the rod configuration $\Psi(\sigma(s))$, which is found by solving* (1)-(4). *Similarly, define the function $t_2\colon [0,L] \to \mathbb{R}^+$ so that $t_2(s)$ is the first point of self-intersection along the rod configuration $\Psi(\sigma(s))$. Now, define the function $t_\sigma\colon [0,L] \to \mathbb{R}^+$ by*

$$t_\sigma(s) = \min\{t_1(s), t_2(s), L\} \qquad (6)$$

Next, define the function $\Theta_\sigma\colon [0,1] \times \mathbb{R}^+ \to \mathcal{A}$ by

$$\Theta_\sigma(s,l) = (l\sigma_1(s)\ l\sigma_2(s)\ l\sigma_3(s)\ l^2\sigma_4(s)\ l^2\sigma_5(s)\ l^2\sigma_6(s))^T$$

where $\sigma_i(s)$ is the i^{th} component of $\sigma(s)$. Now define $\alpha\colon [0,1] \to \mathcal{A}$ by $\alpha(s) = \Theta_\sigma(s, t_\sigma(s))$. We then have

$$\mathcal{A}_\alpha = \{a \in \mathcal{A}\colon a = \Theta_\alpha(s,l) \text{ for some } s \in [0,1], l \in (0,1)\} \subset \mathcal{A}_{free}. \qquad (7)$$

Proof: See the proof of Lemma 1 in [8].

Given two rod configurations $a_i, a_j \in \mathcal{A}_{free}$, any path σ can be chosen such that $\sigma(0) = a_i$ and $\sigma(1) = a_j$, e.g., a straight line as seen in Figure 3. Then, using Theorem 1 and σ, we can compute the path $\alpha(s)$ containing a_i and a_j that is the boundary of the defined slice $\mathcal{A}_\alpha \subset \mathcal{A}_{free}$. Note that except for the boundary points $\alpha(0) = a_i$ and $\alpha(1) = a_j$, points directly on $\alpha(s)$ may not be in $\mathcal{A}_{free}$. Next, let $q_s(t) \in$

$SE(3)$ for $t \in [0, L]$ be the rod configuration solution of $\Psi(\alpha(s))$. According to [8], the solution of $\Psi(\Theta_\alpha(s,l))$ for any $s \in [0,1]$ and $l \in (0,1]$ is given by

$$q_{s,l}(t) = \mathbf{D}_l q_s(lt)\mathbf{D}_l^{-1} \quad \text{where} \quad \mathbf{D}_l = \begin{bmatrix} \mathbf{I}_{3\times3} & \mathbf{0}_{3\times1} \\ \mathbf{0}_{1\times3} & l \end{bmatrix} \tag{8}$$

and for all $t \in [0, L]$. In practice, we have scaled the point $\alpha(s)$ by l and evaluated the rod's configuration at this scaled value l. Given $s \in [0,1]$ and $l \in (0,1]$, it is possible to solve $\Psi(\Theta_\alpha(s,l))$ through the ODE's (1)-(4) to acquire the rod shape $q_{s,l}(t)$. However, as discussed in Section III-C, these computations are quite expensive. On the other hand, the computation of (8) is simple and more efficient. This powerful notion states that once a path α is found, all points in $\mathcal{A}_\alpha$ can simply be evaluated using (8) while not requiring further ODE computations. Thus, we bound the number of ODE computations for an edge to be proportional to the distance between a_i and a_j or equal to $n_{ij} = \frac{1}{\delta_E}\|a_i - a_j\|$ where δ_E is the pre-defined resolution along $\sigma(s)$.

We can now construct paths $\bar{\alpha} : [0,1] \to \mathcal{A}_\alpha$ as follows. Define an n-degree polynomial $h\colon [0,1] \to (0,1]$ with constraints $h(0) = 1$, $h(1) = 1$, and $h(s_i) = f_i$ for some $s_i \in (0,1)$ and $f_i \in (0,1)$ where $i = 1, \ldots, n-1$. The value f_i is a factor scaling $\bar{\alpha}(s)$ away from $\alpha(s)$ (Figure 3). Then solutions along $\bar{\alpha}(s) = \Theta_\alpha(s, h(s))$ are given by (8).

Given two rod configurations $a_i, a_j \in \mathcal{A}_{free}$, a path $\bar{\alpha}\colon [0,1] \to \mathcal{A}_{free}$ can be computed as discussed previously and the rod solution is easily acquired. This method is used in this work to rapidly connect the start and goal rod configurations to their nearest neighbors in $\mathcal{A}_G$. Once added, they are considered an integral part of the roadmap. Therefore, planning with the same start and goal configurations, i.e., in a real-time re-planning framework, may not require any ODE computations, yielding much faster planning. In addition but less significant, the same method can be used to connect milestones when generating the roadmap to expedite the off-line process. Rather than construct the roadmap in a conventional way by straightforward search of valid edges—where some will be rejected and waste time—this method can be used to guarantee that every two nodes in the roadmap are connected by an edge. Experiments suggest that this approach can reduce the off-line computation time by approximately 60%. Therefore, in the off-line and on-line computations, any two configurations can be connected using path-connected slices in $\mathcal{A}_{free}$ with the minimal required number of ODE solutions.

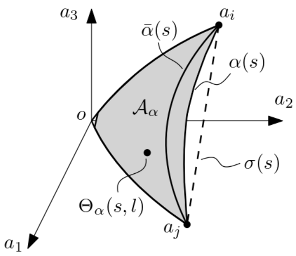

Fig. 3. A slice $\mathcal{A}_\alpha \subset \mathcal{A}_{free}$ formed by α.

D. Probabilistic Completeness

As discussed above, the roadmap $\mathcal{A}_G$ approximates $\mathcal{A}_{free}$ to a fixed subset. Therefore, sampling-based planning in $\mathcal{Z}_G$ is not probabilistic complete. Similar to [32], probabilistic completeness can be regained by continuing to expand $\mathcal{A}_G$ in a parallel process. In addition, the parallel process can continue to connect the start and goal rod configurations to more nodes in the roadmap. If a solution exists but not currently on the roadmap, more paths will be added until found, allowing the planner to be probabilistically complete.

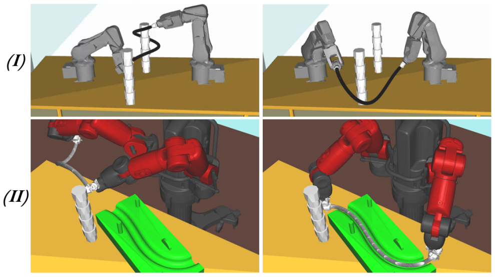

Fig. 4. The two tested environments: (*I*) two ABB robotic arms and (*II*) a Baxter dual-arm robot manipulating an elastic rod. Start and goal configurations are left and right, respectively.

V. Experiments and Analysis

This section presents experiments of planning in $\mathcal{Z}$ and $\mathcal{Z}_G$. The experiments were implemented in C++ with the Open Motion Planning Library (OMPL) [33] on an Intel-Core i7-8550U Ubuntu machine with 8GB of RAM. We used the Kinematics and Dynamics Library to sample configurations that satisfy the closed-kinematic chain constraint.

We tested the planning approach in two different environments seen in Figure 4. Environment *I* consists of two ABB IRB-120 industrial robotic arms. The robots must manipulate an elastic rod in a collision-free path between two poles. The Baxter robot in environment *II* manipulates the same rod into a channel of a specific shape through obstacles. The start and goal configurations were chosen such that they are sufficiently distant from each other while exercising the planner in the presence of obstacles. In both environments, the robots grip a rod of length $L = 1\ m$, and we have assumed the rod has stiffness $\mathbf{c} = (1\ Nm^2, 1\ Nm^2, 1\ Nm^2)$. We begin by showing results for planning in $\mathcal{Z}$ for the two environments. We then show statistics regarding the generation of the roadmap $\mathcal{A}_G$ with regards to different roadmap sizes. Finally, we show performance results of motion planning over $\mathcal{A}_G$ and analyze the effect of the roadmap size on the performance. Videos of the experiments can be seen in the supplementary material.

A. Planning in $\mathcal{Z}$

In this experiment, we show the results for planning in $\mathcal{Z}$, i.e., planning without approximating $\mathcal{A}_{free}$. Each environment was tested with three sampling-based planners: RRT, CBi-RRT and the Single-query Bi-directional probabilistic roadmap planner with Lazy collision checking (SBL) [34]. All planners are standard implementations in OMPL. Table II (left hand side) shows the performance of the three planners in $\mathcal{Z}$ for environments *I* and *II*. The values represent the average of 100 trials. The RRT for environment *II* did not produce any result in a runtime bound of 1,500 seconds. As discussed in Section III-C, planning directly in $\mathcal{Z}$ can indeed produce feasible solutions but with low performance.

A significant portion of the computation time is wasted on solving the ODE's (1)-(4). This motivates the approximation of $\mathcal{A}_{free}$. Results for the approximation are presented next.

B. *Planning in $\mathcal{Z}_G$*

Several approximation roadmaps were generated with $k = 4$ nearest neighbors (selected empirically) and a different number of milestones m. The generation time of the approximation roadmap as well as the size of the resulting data structure are shown at the bottom of Table I. Table I also presents the success rate and the resulting path quality for planning in Env. *I* over 10 randomly generated roadmaps for each size. Roadmaps where $m < 100$ have low success rate and, therefore, are not further used. Path quality is the average planned path length in $\mathcal{A}_{free}$ evidently improving as m increases. Table II summarizes the experimental results for an average of 100 runs for each of the generated roadmaps. Planning is performed for both environments with the same static roadmaps of the rod. The main observation from these results is the runtime reduction by at least an order of magnitude compared to planning directly in $\mathcal{Z}$ as seen in Table II. Moreover, we observe better performance as the density of the roadmap increases. Obviously, more milestones provide a better approximation of $\mathcal{A}_{free}$.

Figure 5 shows the failure distribution when planning in $\mathcal{Z}$ and $\mathcal{Z}_G$. Recall that a minimal number of ODE computations are required when planning in $\mathcal{Z}_G$ for connecting the start and goal rod configurations to $\mathcal{A}_G$. The average number and time of the ODE computations needed to connect the start and goal configurations to the roadmap are also reported in Table II. As we would expect, denser roadmaps yield shorter start and goal connections and therefore fewer ODE computations. We note that the net planning time without the connections is very low. That is, if the start and goal nodes already exist in the roadmap, the runtime is reduced by approximately 85%-92%. The important result is that planning in $\mathcal{Z}_G$ yields a reduction of more than 95% in the number of ODE computations compared to planning in $\mathcal{Z}$.

C. *Random start and goal configurations*

To evaluate the approach with varying boundary conditions, we tested the planning for random start and goal configurations. 100 queries were taken in Env *I* with start and goal configurations randomly chosen from $\mathcal{Z}_o$. All queries are known to have a solution. Planning in $\mathcal{Z}$ was tested with the CBi-RRT planner, and planning in $\mathcal{Z}_G$ was performed using a roadmap of size $m = 100$. This puts a lower bound on the performance. When limiting each planning iteration to 220 sec, the success rates for $\mathcal{Z}$ and $\mathcal{Z}_G$ were, respectively, 56% and 95%, and the average runtimes were 94 ± 56.3 sec and 6.39 ± 4.3 sec. The low success rate in $\mathcal{Z}$ is due to some queries that require excessive runtime. On the other hand, failure in $\mathcal{Z}_G$ is due to limited paths on the roadmap which is solved by increasing m. 100% success rate was achieved when $m = 1,000$.

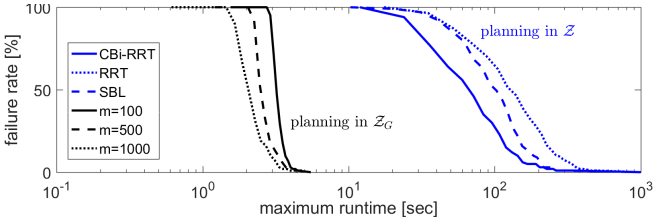

Fig. 5. Failure rate with regards to the average runtime when planning in (right blue curves) $\mathcal{Z}$ and in (left black curves) $\mathcal{Z}_G$.

VI. Hardware Experiments

To validate our approach and test model accuracy, we have built a setup comprised of two opposing ABB IRB-120 industrial robots with a distance of 1.1 m between their bases. We mounted two chuck grippers on the arms that grip a Nitinol rod (length 0.55 m and stiffness $\mathbf{c} = (0.77\ Nm^2, 1\ Nm^2, 1\ Nm^2)$, which corresponds to a circular cross-section) by its ends. For the start and goal configurations in Figure 6, planning in $\mathcal{A}_G$ produced a path in an average of 1.73 seconds, compared to 44.28 seconds when planning in $\mathcal{Z}$. This result shows that planning in $\mathcal{Z}$, even without obstacles, remains challenging in terms of computation time.

Planning either in $\mathcal{Z}$ or $\mathcal{Z}_G$, however, does not affect the accuracy of the rolled-out path. Hence, we here aim to compare planned paths of the rod, which are based on the theoretical model of eq. (1)-(4), and the actual rolled-out ones. The paths were executed by the arms in an open-loop fashion. Using a camera stationed in front of the robots, we reprojected the planned path onto the image plane, shown in Figure 6 by the dashed yellow curves. For the images in Figure 6, the maximum distances within the image plane between points on the physical rod and the closest point on the reprojected rod were 4.83, 7.31, 4.72, 7.79 and 8.51 mm, respectively. Measurement errors exist due to imperfect camera calibration and resolution accuracy. Thirteen additional experiments were conducted with different starting and goal configurations. Across all experiments, the average maximum image plane-distance between points on the physical rod and the closest point on the reprojected rod was 6.79mm.

VII. Conclusions

We have presented the problem of planning the motion of an elastic rod manipulated by two robotic arms. We propose the use of sampling-based algorithms to explore the composite configuration space of the joints space and the free configuration space of the rod. However, we have shown that straight-forward implementation of sampling-based planning is computationally expensive due to the ODE solutions for the rod shape. We therefore propose to approximate the free configuration space of the rod with a roadmap. This roadmap encapsulates the connectivity of the space and stores rod solutions in each node. In addition, the use of stable slices in the free configuration space of the rod lay a bound on the number of ODE computations for a local connection, which significantly reduce connection runtime. Planning for the robot joints over the roadmap along with

TABLE I

ROADMAP PROPERTIES WITH REGARDS TO m

	m				
	20	50	100	500	1,000
gen. time (min)	1.9	14.2	23.2	87.1	135.5
mem. size (MB)	74	163	259	869	1,500
success rate	2/10	6/10	10/10	10/10	10/10
path quality	147.3	128.6	91.7	59.1	48.3

TABLE II

PERFORMANCE DATA FOR PLANNING IN $\mathcal{Z}$ AND $\mathcal{Z}_G$

		planning in $\mathcal{Z}$			planning in $\mathcal{Z}_G$, $m =$		
		RRT	SBL	CBi-RRT	100	500	1,000
env. *I*	avg. time (sec)	125.3	107.5	77.7	3.3	2.56	2.16
	planning time (s)	1.93	1.58	1.25	0.375	0.377	0.472
	% ODE time	98.46	98.53	98.39	88.6	85.3	78.16
	num. ODE comp.	1,784	1,487	856	77	55	44
env. *II*	avg. time (sec)	-	685.5	160.4	4.1	2.7	2.03
	planning time (s)	-	12.82	2.67	0.360	0.316	0.257
	% ODE time	-	98.13	98.33	91.2	88.3	87.3
	num. ODE comp.	-	9,227	3,164	94	62	45

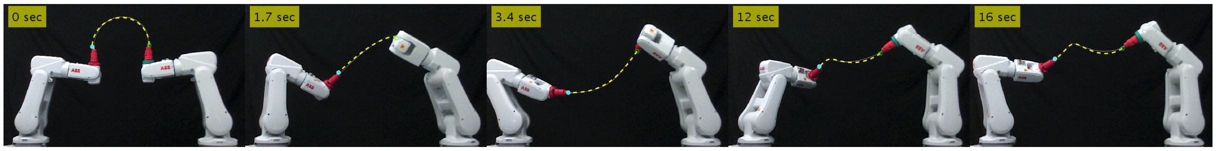

Fig. 6. Snapshots of an experiment where a Nitinol rod is manipulated using two ABB robots between start (left) and goal (right) configurations. The dashed yellow curve is the planned configuration projected onto the image plane.

connections through the slices have been shown, through a set of experiments, to reduce planning runtime by at least an order of magnitude. Future work could focus on including the effects of gravity by considering its potential when minimizing the total elastic energy of the rod to re-derive eq. (1)-(4), and further exploiting the scale invariance property of the rod to better understand the topology of the constrained configuration space $\mathcal{Z}_o$.

REFERENCES

[1] X. Jiang, K.-M. Koo, K. Kikuchi, A. Konno, and M. Uchiyama, "Robotized assembly of a wire harness in a car production line," *Advanced Robotics*, vol. 25, no. 3-4, pp. 473–489, 2011.

[2] R. C. Jackson and M. C. Çavuşoğlu, "Needle path planning for autonomous robotic surgical suturing," in *Proc. IEEE Int. Conf. on Rob. and Aut.*, 2013, pp. 1669–1675.

[3] J. Schulman, J. Ho, C. Lee, and P. Abbeel, *Learning from Demonstrations Through the Use of Non-rigid Registration*, 2016, pp. 339–354.

[4] M. Saadat and P. Nan, "Industrial applications of automatic manipulation of flexible materials," *Ind. Rob.*, vol. 29:5, pp. 434–442, 2002.

[5] T. Bretl and Z. Mccarthy, "Quasi-static manipulation of a kirchhoff elastic rod based on a geometric analysis of equilibrium configurations," *Int. Jou. of Robotics Research*, vol. 33, no. 1, pp. 48–68, 2014.

[6] S. Antman, *Nonlinear Problems of Elasticity*. Springer, 2005.

[7] A. Sintov, A. Borum, and T. Bretl, "Motion planning of fully actuated closed kinematic chains with revolute joints: A comparative analysis," *IEEE Rob. & Aut. Letters*, vol. 3, no. 4, pp. 2886–2893, 2018.

[8] A. Borum and T. Bretl, "The free configuration space of a kirchhoff elastic rod is path-connected," in *Proc. IEEE Int. Conf. on Rob. and Aut.*, 2015, pp. 2958–2964.

[9] A. Short and T. Bandyopadhyay, "Legged motion planning in complex three-dimensional environments," *IEEE Rob. & Aut. Let.*, 2017.

[10] K. Kosuge, H. Yoshida, T. Fukuda, M. Sakai, and K. Kanitani, "Manipulation of a flexible object by dual manipulators," in *Proc. IEEE Int. Conf. on Rob. and Aut.*, vol. 1, 1995.

[11] H. G. Tanner, "Mobile manipulation of flexible objects under deformation constraints," *IEEE Trans. on Rob.*, vol. 22, pp. 179–184, 2006.

[12] D. Navarro-Alarcon, H. M. Yip, Z. Wang, Y. H. Liu, F. Zhong, T. Zhang, and P. Li, "Automatic 3-d manipulation of soft objects by robotic arms with an adaptive deformation model," *IEEE Transactions on Robotics*, vol. 32, no. 2, pp. 429–441, April 2016.

[13] R. Levien, "The elastica: a mathematical history," EECS Department, University of California, Berkeley, Tech. Rep., Aug 2008.

[14] F. Lamiraux and L. E. Kavraki, "Planning paths for elastic objects under manipulation constraints," *International Journal of Robotics Research*, vol. 20, pp. 188–208, 2001.

[15] M. Moll and L. E. Kavraki, "Path planning for deformable linear objects," *IEEE Trans. on Robotics*, vol. 22, no. 4, pp. 625–636, 2006.

[16] T. Hermansson, R. Bohlin, J. S. Carlson, and R. Söderberg, "Automatic assembly path planning for wiring harness installations," *Journal of Manufacturing Systems*, vol. 32, no. 3, pp. 417 – 422, 2013.

[17] I. Kabul, R. Gayle, and M. C. Lin, "Cable route planning in complex environments using constrained sampling," in *Proc. of the ACM Symposium on Solid and Physical Modeling*, NY, 2007, pp. 395–402.

[18] O. Roussel, A. Borum, M. Taïx, and T. Bretl, "Manipulation planning with contacts for an extensible elastic rod by sampling on the submanifold of static equilibrium configurations," in *Proc. IEEE Int. Conf. on Rob. and Aut.*, 2015, pp. 3116–3121.

[19] M. Elbanhawi and M. Simic, "Sampling-based robot motion planning: A review," *IEEE Access*, vol. 2, pp. 56–77, 2014.

[20] L. E. Kavraki, P. Svestka, J. C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Tran. Rob. and Aut.*, vol. 12, no. 4, pp. 566–580, 1996.

[21] L. G. Torres, C. Baykal, and R. Alterovitz, "Interactive-rate motion planning for concentric tube robots," in *Proc. IEEE Int. Conf. on Rob. and Aut.*, 2014, pp. 1915–1921.

[22] J. van den Berg, D. Ferguson, and J. Kuffner, "Anytime path planning and replanning in dynamic environments," in *Proc. IEEE Int. Conf. on Rob. and Aut. (ICRA)*, Pittsburgh, PA, May 2006, pp. 2366 – 2371.

[23] O. Salzman, K. Solovey, and D. Halperin, "Motion planning for multilink robots by implicit configuration-space tiling," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 760–767, July 2016.

[24] T.-Y. Li and H.-C. Chou, "Motion planning for a crowd of robots," in *Proc. IEEE Int. Conf. on Rob. and Aut.*, vol. 3, 2003, pp. 4215–4221.

[25] P. Isto and M. Saha, "A slicing connection strategy for constructing prms in high-dimensional cspaces," in *IEEE ICRA*, 2006.

[26] T. Lozano-Perez, "A simple motion-planning algorithm for general robot manipulators," *IEEE J. Rob. & Aut.*, vol. 3, pp. 224–238, 1987.

[27] M. Saha and P. Isto, "Multi-robot motion planning by incremental coordination," in *IEEE/RSJ IROS*, 2006, pp. 5960–5963.

[28] T. H. Cormen, C. Stein, R. L. Rivest, and C. E. Leiserson, *Introduction to Algorithms*, 2nd ed. McGraw-Hill Higher Education, 2001.

[29] S. LaValle and J. Kuffner, J.J., "Randomized kinodynamic planning," in *Proc. IEEE Int. Conf. on Rob. and Aut.*, vol. 1, 1999, pp. 473–479.

[30] J. J. Kuffner and S. M. LaValle, "Rrt-connect: An efficient approach to single-query path planning," in *Proc. IEEE Int. Conf. on Rob. and Aut.*, vol. 2, 2000, pp. 995–1001.

[31] D. Berenson, S. Srinivasa, D. Ferguson , and J. Kuffner, "Manipulation planning on constraint manifolds," in *Proc. IEEE Int. Conf. on Rob. and Aut.*, 2009.

[32] I. A. Şucan and S. Chitta, "Motion planning with constraints using configuration space approximations," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 1904–1910.

[33] I. A. Sucan, M. Moll, and L. E. Kavraki, "The open motion planning library," *IEEE Rob. & Aut. Mag.*, vol. 19, no. 4, pp. 72–82, 2012.

[34] G. Sánchez and J.-C. Latombe, *A Single-Query Bi-Directional Probabilistic Roadmap Planner with Lazy Collision Checking*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 403–417.