



HAL
open science

Motion Planning for an Elastic Rod using Contacts

Olivier Roussel, Pierre Fernbach, Michel Taïx

► **To cite this version:**

Olivier Roussel, Pierre Fernbach, Michel Taïx. Motion Planning for an Elastic Rod using Contacts. IEEE Transactions on Automation Science and Engineering, 2020, 17 (2), pp.670-683. 10.1109/TASE.2019.2941046 . hal-01954894

HAL Id: hal-01954894

<https://hal.science/hal-01954894>

Submitted on 14 Dec 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Motion Planning for an Elastic Rod using Contacts

Olivier Roussel, Pierre Fernbach and Michel Taïx

Abstract—This paper focuses on the motion planning problem of an extensible elastic rod in collision-free or contact space. The rod is assumed to be handled by grippers either at one or both extremities. Furthermore, during manipulation, the grasped end may change. We show that the use of both quasi-static and dynamic models can be coupled efficiently with sampling-based methods. Sampling directly in the submanifold of static equilibrium and contact-free configurations allows to take advantage of the dynamic model to improve the exploration of the state space. In this way, thanks to the contact information (point, forces, direction, number of contacts), the exploration of the RRT approach can be improved. We present a new RRT-SLIDE algorithm which guides the roadmap extension with a sliding contact mode based on some principles of human reasoning. We show that our approach is probabilistically complete. We also demonstrate the necessity of considering contacts on complex scenarios with several simulation experiments. Besides its performances, our algorithm does not require further tuning phase for a new scenario.

Note to Practitioners—This work was done under the industrial project Flecto (ANR-Digital Models). It aims at solving the assembly/disassembly task for a rod while satisfying the elasticity parameters of its material. For industrial applications, the resolution time is a critical point. On the one hand, probabilistic motion planning methods require to efficiently build a roadmap of valid rod configurations. On the other hand, accurate rod modeling implies the use of a simulator based on the finite element method. Nevertheless, the very large size of the roadmap, that leads to a high number of calls to the simulator, is conflicting with the high computational cost of finite-element based simulation. To overcome this problem, one solution is to reduce the number of simulator calls. This can be achieved by sampling the free space with an efficient parameterization and by limiting the use of the simulator to roadmap extension in the free space or in the contact space. We introduce heuristics based on contact information returned by the simulator to reduce significantly the computational time. One of the main advantages of our algorithm is that it does not require any tuning phase for each scenario. Although we do not solve the more general gripper manipulation planning problem, this approach could be used as a first step before computing the motion of the grippers. In the framework of our project, we did not consider disassembling operations implying undoing rod knots. Consequently, we do not take friction into account in our approach. In order to handle knots, it would be necessary to have a physics simulator that could handle friction for deformable rods.

Keywords—Sampling-based motion planning, elastic rod motion, sliding motion planning

I. INTRODUCTION

Manufacturing industries have taken benefit from numerical simulations for several decades. One of the most recent

applications is the digital mockup for assembly path determination. Motion planning plays an essential role in planning assembling/disassembling industrial tasks in order to validate mechanical designs. In robotics, a lot of work has been done in motion planning to compute free paths in digital models for mechanical systems. Random planning algorithms developed during the last decades, such as the Rapidly-exploring Random Tree (RRT) [1], enabled automatic problem resolution for rigid bodies having a large number of degrees of freedom.

In automotive or aeronautical industry, assembling deformable parts is a frequent requirement. In this context, most of these parts are circular Deformable Linear Objects (DLOs) (cable, hose, pipe,...) that are usually referred to as rods in the mechanical literature. A rod cannot be freely twisted, as its deformation is limited by the flexibility properties of the material in order to stay within its elasticity domain. As a consequence, assembly rod motion must satisfy the constraints imposed by elasticity properties.

In many motion planning problems, the goal is to compute a path or a trajectory in the collision-free space, i.e. without contact. The main reason for this is to reduce computation time. However, in realistic constrained environments, the contact condition is actually necessary. The idea of using contact or collision information to improve motion planning has been explored with rigid objects in previous works [2] [3] [4].

The goal of this paper is to solve the motion planning problem for a rod in admissible space (i.e. free or contact space) based on a RRT approach. A typical industrial example is shown in Figure 1. The solution trajectory in the admissible space must satisfy the elasticity constraints of the material as well as rod dynamics.

In probabilistic motion planning, sampling in the free configuration space is usually easy and can be achieved by rejection using a collision test. Nevertheless, sampling in the contact space is much more complex and cannot be achieved by rejection sampling in the configuration space, as the contact manifold is of lower dimension. To overcome this problem, we sample configurations in the free space only, using a quasi-static rod model [5]. Then, the roadmap extension is done using a dynamic model as local method, which can handle rod motions with contacts. Fitting between the two models is also presented.

In this paper, the rod is assumed to be manipulated by grippers at its extremities, but we do not consider the task of controlling the manipulator movement. Instead, we assume that the rod can be moved to follow our motion planning solution. This assumption is sufficient to prove the feasibility of the assembly task. It is a necessary first step for the general problem discussed in section VII which can be decoupled using the assumption presented above. To move the rod, we assume that only one of the grippers moves at a time and that

This work has been partially supported by the French National Research Agency under the industrial project Flecto (ANR- Digital Models)

All the authors are with LAAS-CNRS, Univ. de Toulouse, CNRS, UPS, Toulouse, France

Corresponding author: O. Roussel (olivier.roussel@laas.fr)

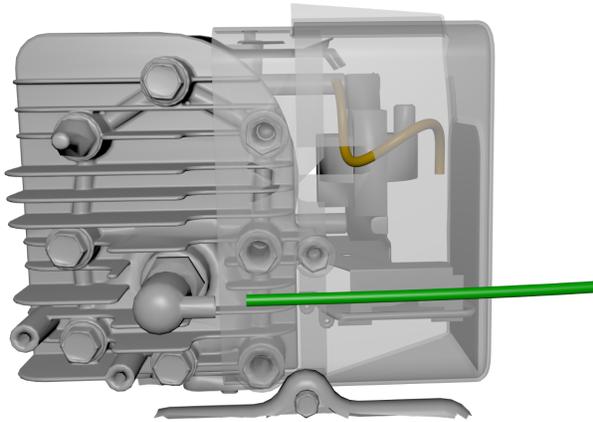


Figure 1: Industrial cases of disassembly studies for a free-flying cable. The cable has to get out from a highly constrained starting state (in yellow) to a low constrained state (in green).

the moving gripper can change during the motion.

An important requirement is to take into account the physical properties of the rod defined by its elasticity parameters. We therefore restricted our study to the case of a deformable rod amidst rigid obstacles. The case of rigid object manipulation in a deformable environment, as studied in [6], will not be addressed here. In this research project, we do not consider industrial cases where it is necessary to undo knots. Consequently, friction is not essential and can be neglected.

As the rod can be geometrically described by an infinite-dimensional continuous curve, planning in the discretization of this curve may lead to high finite-dimensional configuration spaces. Identifying the manifold of feasible configurations, which satisfy the mechanical constraints, is a challenging task. If one wants to consider both rod dynamics and contacts, this task becomes even more difficult.

In a previous work [7], we used the simulator XDE to generate sliding motions along the contact space without using contact information. The first contribution of the present paper is to take advantage of this contact information, i.e. normals, intensity and other characteristics of contact forces, returned by the dynamic simulator XDE to solve efficiently a new class of motion planning problems. We also provide an analysis of the simulation results to show the interest of each contact information.

The paper is organized as follows. A review of the related work is presented in next section. Section IV explains how to take advantage of both models (presented in section III) using the RRT method, in order to solve the motion planning problem for an extensible elastic rod in the admissible configuration space. Motion planning using explicit contact information is presented in section V. Finally, in section VI, the method is evaluated and discussed on several planning scenarios.

II. RELATED WORK

Some work on path planning for an elastic rod has been done considering only collision free space of quasi-static

configurations. These approaches are based on the numerical minimization of the total elastic energy for given gripper placements [8], [9], [10].

An approach to the motion planning problem considering rod dynamics and contacts is to plan in the finite space of controls for a given number of grippers. In this case, the state transition function is assumed to be known and can be delegated, for example, to a simulator. For instance, the motion planning problem for deformable objects is addressed in [11], where a deformable dynamics model is coupled with a kinodynamic motion planning algorithm. However, the use of fully deformable environments prevents the robot to be stuck in local minima and bypasses the local control problem.

More generally, the motion planning problem for deformable objects has already been investigated, especially in the case of simplified visually realistic deformation models. For example, [12] extends the Probabilistic Roadmap Methods (PRMs) [13] for deformable objects by reducing the deformation space to a one dimensional one. [14] uses the Constraint Based Motion Planning framework to simulate a deformable robot along a guide path computed for a point-like robot. This work has then been extended to the specific class of DLOs in [15]. Another interesting direction is investigated by [16] based on a two-step approach. A learning phase first collects high-dimensional samples and computes a new basis for the deformation set using linear dimensionality reduction. Then, this reduced deformation space is used with classical motion planning methods. Its main drawback lies in the limitation to linear reduction.

Assuming the rod is handled by both extremities, recent results based on the local solution of a geometric optimal control problem enabled to define this configuration space [5]. In this paper, it is shown that this configuration space defines a finite dimensional manifold that can be parameterized by a single chart. It is also shown that this parameterization is directly related to the wrenches applied on rod grippers. In previous work [17], we presented a motion planning algorithm for a quasi-static inextensible elastic rod and we showed that solution paths in the collision-free space can be computed efficiently. The elastic rod model being only valid without contact, we restricted the rod motion to the collision free space.

Based on this work, many tests in constrained environments show the bottleneck is due to the rod Direct Geometry Model (DGM) computation for each configuration, which is necessary to check collisions. The DGM computation time is at least in the same order of magnitude than collision checking.

In order to improve the algorithm performances, our approach uses simultaneously motion in the free space and in the contact space. To this end, any physics engine component can be used to simulate the rod motion, as long as this engine is able to simulate rod motion with contacts, provides contact information and allows to specify material elasticity parameters for the rod. We use eXtended Dynamic Engine (XDE) [18] as physical engine, which offers a realistic multi-body dynamics simulation with contacts for deformable bodies such as rods, modelled as geometrically exact 3D beams. As they are based on finite elements methods, these simulators are well suited for applications requiring physically accurate

simulations. But due to a very large number of calls of the simulator, the computational cost is too high to couple them with classical motion planning algorithms in an industrial context. However, it seems interesting to allow sliding motions along the contact space to improve significantly the exploration [2] and some contact information is already available thanks to contact models simulated by XDE.

In this context, we define a motion planning approach based on sampling-based methods in the admissible space. By sampling on the submanifold of quasi-static contact-free configurations, it is then possible to use dynamic simulation to extend the exploration of the state space. Using a parameterization to sample directly on the submanifold, we can limit the number of calls to the simulator to extension steps and take advantage of the provided information such as contact forces. Then, by allowing contacts between the deformable object and obstacles and sliding along the contact space, we show that we increase the deformation space and that we can efficiently handle constrained environments, i.e. having a very poor ϵ - *goodness* [19] as well as narrow passages. As our approach uses two different models for dynamic simulation and static equilibrium configurations, it is necessary to fit the model parameters order to ensure their convergence at quasi-static configurations.

III. ELASTIC RODS MODELING

This part essentially summarizes [7]. The aim is to present the two elastic rods models considered in our approach, each defining a different configuration space. On one side, we consider a quasi-static model in the collision-free space, living in a low dimensional space, and on the other side we make use of a rod dynamic model that handles contacts, living in a high dimensional space. Both of these models share material elasticity parameters through simple connections presented here.

A. Extensible elastic rods Static Equilibrium Model (SEM)

By formulating rod static equilibrium as an optimal control problem, Bretl and McCarthy [5] have shown that the manifold of equilibrium configurations of an inextensible and non-shearable rod can be described by a single global chart. In [7], we have proposed an extension of this work to an extensible and shearable rod which will be briefly summarized in this section.

Consider a thin, naturally straight elastic rod in static equilibrium (i.e. the rod is motionless) held fixed at each end. We may assume without loss of generality that the rod has unit length. With $t \in [0, 1]$ denoting 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 shape of the entire rod is described by a continuous map $q: [0, 1] \rightarrow SE(3)$. The rod is allowed to bend and twist, and the rod experiences axial extension and shear deformations. Therefore, the map $q: [0, 1] \rightarrow SE(3)$ must satisfy

$$\dot{q} = q \left(\sum_{i=1}^6 u_i X_i + X_4 \right) \quad (1)$$

for some $u: [0, 1] \rightarrow \mathbb{R}^6$, where X_i with $i \in \{1, \dots, 6\}$ is a basis for the Lie algebra $\mathfrak{se}(3)$.

The functions $u_i: [0, 1] \rightarrow \mathbb{R}$ with $i \in \{1, \dots, 6\}$ are strains along the rod, where u_1 is the twisting strain, u_2 and u_3 are the bending strains, u_4 is the axial strain, and u_5 and u_6 are the shear strains. We refer to q and u together as (q, u) .

The ends of the rod $q(0)$ and $q(1)$ are held by robotic grippers. Denote the set of all possible $q(0)$ by $\mathcal{B}_0 = SE(3)$ and the set of all possible $q(1)$ by $\mathcal{B} = SE(3)$, i.e. \mathcal{B}_0 and \mathcal{B} are the set of all possible placements of the robotic grippers holding the ends of the rod. Static equilibrium configurations of the rod are those which locally minimize the elastic potential energy stored in the rod. Assuming the elastic potential energy is quadratic in the strain functions and that twist, bending, stretching, and shear deformations are not coupled, the elastic potential energy is

$$\frac{1}{2} \int_0^1 \sum_{i=1}^6 c_i u_i^2 dt$$

where $c_1 > 0$ is the torsional stiffness, $c_2 > 0$ and $c_3 > 0$ are the bending stiffnesses, $c_4 > 0$ is the axial stiffness, and $c_5 > 0$ and $c_6 > 0$ are the shear stiffnesses of the rod. We say that (q, u) is a static equilibrium configuration of the rod if it is a local optimum of

$$\begin{aligned} & \underset{q, u}{\text{minimize}} && \frac{1}{2} \int_0^1 \sum_{i=1}^6 c_i u_i^2 dt \\ & \text{subject to} && \dot{q} = q \left(\sum_{i=1}^6 u_i X_i + X_4 \right) \\ & && q(0) = b_0, \quad q(1) = b \end{aligned} \quad (2)$$

for some $b_0 \in \mathcal{B}_0$ and $b \in \mathcal{B}$.

Theorem 1 in [7] provides conditions under which (q, u) is a normal extremal of (2). Let \mathcal{C} denote the set of all smooth maps $(q, u): [0, 1] \rightarrow SE(3) \times \mathbb{R}^6$ which satisfy the necessary conditions given in Theorem 1. Then any $(q, u) \in \mathcal{C}$ is completely defined by the choice of $\mathbf{a} \in \mathcal{A}$. Denote the resulting map by $\Psi: \mathcal{A} \rightarrow \mathcal{C}$. Theorem 2 in [7] provides a test to determine which of these extremals are local optima of (2), i.e which $\mathbf{a} \in \mathcal{A}$ produce local optima $\Psi(\mathbf{a}) \in \mathcal{C}$. Let $\mathcal{A}_{\text{stable}} \subset \mathcal{A}$ be the subset of all $\mathbf{a} \in \mathcal{A}$ which satisfy these conditions, and let $\mathcal{C}_{\text{stable}} = \Psi(\mathcal{A}_{\text{stable}}) \subset \mathcal{C}$.

These results enable us to use the coordinates $\mathcal{A}_{\text{stable}}$ to describe an equilibrium configuration of the rod. As coordinates in this chart are a subset of a low dimensional Euclidean space, it is especially well suited for sampling-based methods.

Furthermore, we will denote by $\mathcal{B}_{\text{stable}}$ the set of all $b \in \mathcal{B}$ such that there exists a static equilibrium configuration (q, u) for which $q(1) = b$. It is necessary to emphasize that for each configuration $(q, u) \in \mathcal{C}_{\text{stable}}$, there are a countable number of configurations in $\mathcal{C}_{\text{stable}}$ having the same grippers positions than (q, u) . Let $\Phi: \mathcal{C} \rightarrow \mathcal{B}$ be the map taking (q, u) to $q(1)$. By extension of Theorem 8 in [5], and considering $\Psi|_{\mathcal{A}_{\text{stable}}}$ the restriction of Ψ to $\mathcal{A}_{\text{stable}}$, the map $\Phi \circ \Psi|_{\mathcal{A}_{\text{stable}}}: \mathcal{A}_{\text{stable}} \rightarrow \mathcal{B}_{\text{stable}}$ is also a local diffeomorphism.

We now have a characterization of the set of all static equilibrium configurations of the rod. This allows us to use

a sampling-based planning approach in which we directly sample static equilibrium configurations.

B. The physical engine XDE

XDE offers a realistic multi-body dynamics simulation with various kinematics constraints (e.g. joints, kinematic loops) and with real-time performances [18]. In addition to rigid bodies and kinematic chains, it can also handle rods, modelled as geometrically exact 3D beams [20]. This model enables large displacements thanks to Reissner kinematics and uses geometrically exact finite elements. XDE can also simulate multi-body dynamics with non-smooth contacts between rigid bodies and rods and gives normal vector surface on contact points and intensity of contact forces. However, note that XDE cannot simulate friction between rods and other bodies as well as for rod self-contacts, which are currently unsupported. This limitation prevents from handling knots in the motion planning problem.

Furthermore, XDE provides smooth bodies interactions mechanisms based on reference frames control. For example, one must attach a reference frame to a rod extremity through a Proportional-Derivative controller and interact directly with this frame in order to simulate rod motion. In our work, this will be used for controlling rod grippers in simulation within the local planning method. Thus, XDE provides a dynamical model of extensible and shearable rods with contacts, based on numerical simulation.

C. Model fitting between the SEM and XDE

For the purposes of this paper, we consider rods that are made from an homogeneous isotropic linear elastic material. As is common in mechanical engineering, XDE defines rod material elasticity parameters through Young’s modulus E and shear modulus G . SEM stiffness coefficients c_i can then be deduced from elasticity parameters by $c_1 = GJ, c_2 = c_3 = EI, c_4 = EA$ and $c_5 = c_6 = GA$ where I represents the second moment of area, J the polar moment of inertia and A the cross-section area of the rod.

Using these parameters, simulated dynamic rods can be brought to quasi-static configurations described by the SEM by fitting rod grippers positions. As there exists a countable number of quasi-static configurations for a given grippers placement, this method does not give us the guarantee to fall into the desired local minimum of elastic energy. However, this will be sufficient to our needs and for keeping good exploration properties as explained in section VI-A.

IV. MOTION PLANNING IN ADMISSIBLE SPACE

In this section, we show how the use of both models can improve the efficiency of solving our rod motion planning problem. By taking advantage of the parameterization provided by the SEM, we can sample quasi-static contact-free configurations that guide the exploration in the space of all admissible cable states reachable by the simulator. The *RRT-GRIP* Algorithm 1 combines the accuracy of the XDE finite elements simulator with efficient exploration capabilities based on a kinodynamic RRT.

A. Problem formulation

As the configuration of the rod can be represented by the mapping $q : [0, 1] \rightarrow SE(3)$, the resulting configuration space is a subset of the infinite-dimensional function space $C^\infty([0, 1], SE(3))$. In our case, the rod is discretized by the simulator using finite elements into $N - 1$ elements (thus N nodes), thus the resulting configuration space is a subset of $SE(3)^N$, where each configuration \mathbf{q} is given by $\mathbf{q} = (q_1 \ q_2 \ \dots \ q_N)^\top$.

However, as we also need to consider rod dynamics provided by the simulator, the state space of the rod must be explored during motion planning. Let $\mathcal{X} \subset (TSE(3))^N$, i.e. the tangent bundle associated to each node configuration, be the state space of all rod states \mathbf{x} satisfying elasticity constraints defined by $\mathbf{x} = (\mathbf{q} \ \dot{\mathbf{q}})^\top$. The main drawback of considering the space of states is that we do not have a direct way to sample consistent states for the rod on \mathcal{X} .

Let now \mathcal{X}_{obs} be the closed set of all states in collision with an obstacle and the space of admissible states be defined by $\mathcal{X}_{adm} = \mathcal{X} \setminus int(\mathcal{X}_{obs})$. The simulator, which relies on a state transition function, must guarantee that from any initial state in \mathcal{X}_{adm} , all states resulting for any input will be also in \mathcal{X}_{adm} .

On the other hand, the result in section III-A has shown that all static equilibrium configurations of the rod can be parameterized using single chart mapping from a 6-dimensional Euclidean space \mathcal{A}_{stable} for a fixed rod base. Let now consider a free-flying quasi-static rod, its configuration space can be expressed using the same chart along with a parameterization of $SE(3)$. The 12-dimensional submanifold of \mathcal{X} parameterized by this mapping is the open set \mathcal{Y} that describes all free-flying quasi-static elastic rods held by its both extremities, i.e. without contacts along the rod. The set \mathcal{Y} is defined by

$$\mathcal{Y} = \mathcal{C}_{stable} \times SE(3)$$

Alternatively, this submanifold can be similarly defined by $\mathcal{Y} = \{\mathbf{x} \in int(\mathcal{X}_{adm}) \mid \dot{\mathbf{q}} = \mathbf{0}\}$.

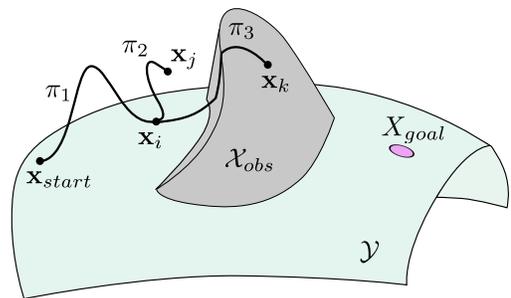


Figure 2: Illustration of the tree exploration in the state space \mathcal{X}_{adm} while sampling states on the submanifold of quasi-static contact-free states \mathcal{Y} . We can see that the resulting state of an extension step can lie on this submanifold as \mathbf{x}_i , be in contact as \mathbf{x}_k or be any state in \mathcal{X}_{adm} as \mathbf{x}_j .

We denote by $\mathbf{x}_{start} \in \mathcal{Y}$ the problem initial state and by $X_{goal} \subset \mathcal{Y}$ the problem goal region. Note that these are restricted to the set of quasi-static collision-free configurations.

As shown in Figure 2, the key idea is to explore the space of all reachable states \mathcal{X}_{adm} while sampling states on the submanifold \mathcal{Y} using coordinates in \mathcal{A}_{stable} along with a parameterization of $SE(3)$.

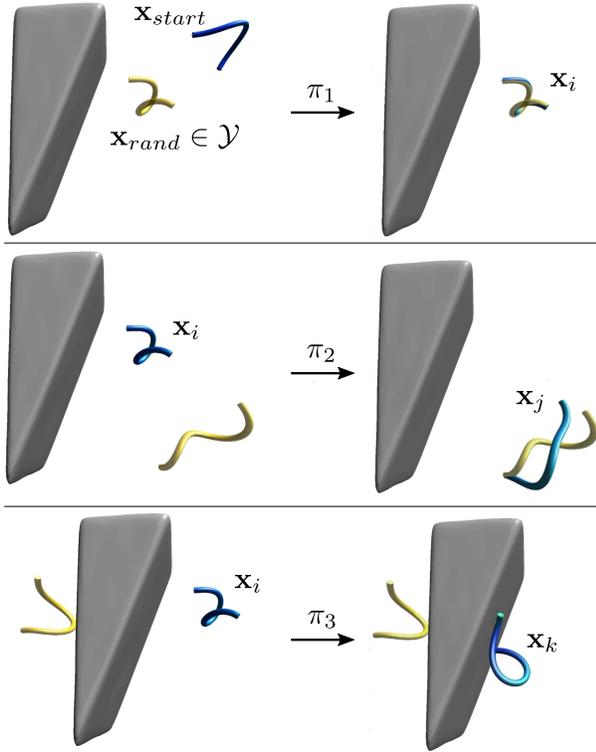


Figure 3: The three different roadmap extension cases in the workspace: π_1 (top), π_2 (middle) and π_3 (bottom). For each case, the randomly sampled state \mathbf{x}_{rand} is shown (in yellow).

During the roadmap extension in \mathcal{X}_{adm} , three cases are possible as illustrated in Figure 2 :

- π_1 : the extend method reach the $\mathbf{x}_{rand} = \mathbf{x}_i \in \mathcal{Y}$,
- π_2 : the extend method does not reach the \mathbf{x}_{rand} . The simulation timeout t_{max} stops the extend at $\mathbf{x}_j \notin \mathcal{Y}$,
- π_3 : the extend method does not reach the \mathbf{x}_{rand} , it slides and stops in contact at \mathbf{x}_k , $\mathbf{x}_k \in \partial X_{obs}$ and $\notin \mathcal{Y}$. Eventually, it can fall back to the previous cases if the motion in contact is over.

Figure 3 shows these different extend cases in the workspace corresponding to the following cases in \mathcal{X}_{adm} .

B. Local planning using dynamics and contacts

In our context, the local planning problem consists in finding the trajectory from a given state $\mathbf{x}_{from} \in \mathcal{X}_{adm}$ to a sampled state $\mathbf{x}_{to} \in \mathcal{Y}$. There are various possible approaches and we describe in this subsection our controller addressing this sub-problem.

We emphasize that taking advantage of contact information provided by the dynamic simulator by using contact sliding motions improves significantly the exploration of the search space for highly constrained cases (see section VI).

In this direction, we choose to exploit contact motions. Consequently, the local planner trajectory is not constrained

on the submanifold \mathcal{Y} . Our method consists in a two steps approach. First, the rod is manipulated by one of its grippers until the considered extremity reaches its goal position. Then, the other gripper is manipulated the same way while keeping the first one fixed as illustrated in Figure 4. The detailed algorithm is given in [7].

To move the gripper (i.e. one rod extremity), the XDE simulator provides a Proportional-Derivative controller on its position. We also consider a time limit t_{max} , chosen at random between given bounds T_{low} and T_{up} , for which the controllers are being applied for a given local planning instance.

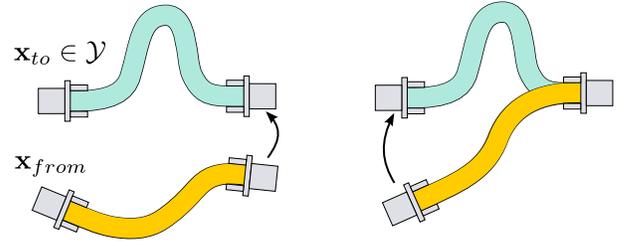


Figure 4: Illustration of the two steps approach for the local planner from a state \mathbf{x}_{from} to a quasi-static state \mathbf{x}_{to} . First, a randomly chosen gripper is manipulated to its respective goal (left). Once its goal is reached, the first gripper is fixed and the other one is manipulated the same way (right).

We can now define the control space by $\mathcal{U} = \mathcal{U}_\omega \times \{g_0, g_1\}$ where $\omega \in \mathcal{U}_\omega \subset \mathbb{R}^6$ is the wrench applied by the gripper on the rod and $\{g_0, g_1\}$ is a discrete degree of freedom indicating on which gripper the wrench is applied. To avoid confusion with existing notation, controls in \mathcal{U} will be referred to as ν .

As mentioned in section III-A, for a given rod configuration in $\mathbf{x}_{to} \in \mathcal{Y}$, there exists a finite number of configurations in \mathcal{Y} having the same end extremities positions. Then, when steering towards \mathbf{x}_{to} using our local method, there is no guarantee at this point that the local trajectory converges to \mathbf{x}_{to} . We will actually see in section IV-F that the probability it converges to \mathbf{x}_{to} tends to one as the number of planner iterations tends to infinity.

C. Kinodynamic planning with sampling on a submanifold

Our approach to the manipulation planning problem for elastics rods is based on classical kynodynamic sampling-based approaches [21] as detailed in Algorithm 1. Some work has been done in sampling-based planning while projecting configurations on a constraint manifold as sampling strategy [22]. In our case, the complexity of the constraints based on the elasticity of the rod material make the formulation of a projector to the sampling submanifold a difficult task. However, the single global chart derived from the SEM that parameterize the submanifold \mathcal{Y} provides us a direct way to sample on the manifold and in particular offers the guarantee that it can be covered entirely through random sampling.

The EXTEND_XDE method refers to the local planner presented previously. When a new state is added in the roadmap \mathcal{T} , it is labelled as free-space state or as contact state (as \mathbf{x}_k in figure 2).

Algorithm 1 RRT-GRIP: Kinodynamic RRT for an elastic rod with sampling on the quasi-static states submanifold

Input: Environment model, \mathbf{x}_{start} , X_{goal}

- 1: Initialize the tree \mathcal{T} with \mathbf{x}_{start}
- 2: **while** \neg solved **and** $iter < N_{max}$ **do**
- 3: $\mathbf{x}_{rand} \leftarrow$ random quasi-static state $\in \mathcal{Y}$ or state in X_{goal}
- 4: $\mathbf{x}_{near} \leftarrow$ NEAREST($\mathcal{T}, \mathbf{x}_{rand}$)
- 5: $\mathbf{x}_{new} \leftarrow$ EXTEND_XDE($\mathbf{x}_{near}, \mathbf{x}_{rand}$)
- 6: Add current state \mathbf{x}_{new} to \mathcal{T}
- 7: Add edge ($\mathbf{x}_{near}, \mathbf{x}_{new}$)
- 8: **end while**

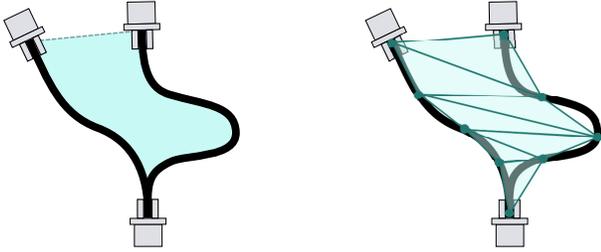


Figure 5: Swept area between two rod configurations and triangulation approximation used by the metric.

Recalling that rod state is described into N nodes such that $\mathbf{x} = (q_1 \dots q_N \dot{q}_1 \dots \dot{q}_N)^\top$, the pseudo-metric $\rho : \mathcal{X} \times \mathcal{X} \rightarrow [0, \infty)$ considered in the NEAREST function is defined by

$$\rho(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{2} \sum_{k=1}^N \left(\|(\mathbf{p}_{i,k} - \mathbf{p}_{i,k-1}) \times (\mathbf{p}_{j,k-1} - \mathbf{p}_{i,k-1})\| + \|(\mathbf{p}_{j,k} - \mathbf{p}_{j,k-1}) \times (\mathbf{p}_{j,k} - \mathbf{p}_{i,k})\| \right) \quad (3)$$

where $\mathbf{p}_{m,k} \in \mathbb{R}^3$ is the translation part of the k -th node of the rod state \mathbf{x}_m .

Intuitively, this pseudo-metric approximates the swept surface area in the workspace between the two rod states as if all nodes could move in straight line, which gives us a lower bound to the real swept surface area (see Figure 5). As the number of rod nodes N would tend to infinity, the continuous equivalent would be the integral

$$\bar{\rho}(x_i, x_j) = \int_0^1 \|\mathbf{p}_i(t) - \mathbf{p}_j(t)\| dt \quad (4)$$

where $x_i, x_j \in \mathcal{Y}$ can be seen as the continuous function equivalent of rod state \mathbf{x} when N tends toward infinity.

Also, as rod velocities are not taken into account in this function, we have no guarantee that the reflexivity axiom is satisfied, i.e. $\rho(\mathbf{x}_i, \mathbf{x}_j) = 0$ does not imply that $\mathbf{x}_i = \mathbf{x}_j$. Thus, this function is only a pseudo-metric on \mathcal{X} . This design choice is due to the fact that we focus on probabilistic completeness of our planner on the submanifold \mathcal{Y} , not on the state space \mathcal{X}_{adm} . This point will be discussed in Section IV-F.

D. Motion Planning with XDE simulator

As illustrated in Figure 6, the output to the motion planning problem is the control trajectory $\pi_{sol} : [0, T] \rightarrow \mathcal{U}$ of duration T . The state of the rod along this trajectory can be fully retrieved from π_{sol} .

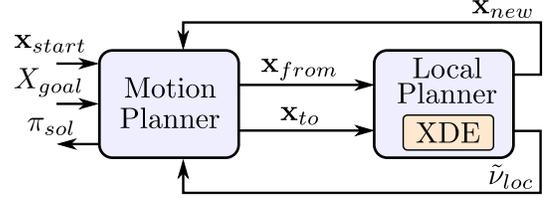


Figure 6: Overview of the motion planning framework.

At the local planning level, XDE provides a simulator and a PD controller which computes, for a given rod state \mathbf{x}_{from} and a sampled gripper state $b \in \mathcal{B}$, the sequence of controls $\tilde{\nu}_{loc} : [0, T_{loc}] \rightarrow \mathcal{U}$ to reach the resulting state \mathbf{x}_{to} , where T_{loc} is the duration of this local control sequence. At each simulation step, the PD controller computes the corresponding wrench and applies it on the gripper. As the gripper moves, the rod follows and may slides along obstacles (see Figure 7). The physics engine XDE can provide the state transition model $\mathbf{x}_{k+1} = f(\mathbf{x}_k, \nu_k)$ for a rod at step k . The given control ν_k represents external force and torque applied on a given gripper of the rod.

The PD controller for the gripper terminates if the goal b_{pos} is reached at null speed (i.e. if the error is below a threshold) or after a given amount of simulated time t_{max} , avoiding the system to be stuck in local minima.

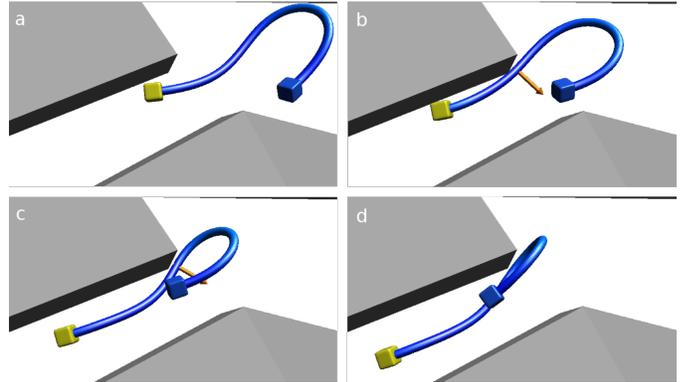


Figure 7: Local planning sliding motion of the rod (in blue) along an obstacle. Normal forces are shown as orange arrows. The yellow extremity represents the actuated gripper.

E. Benefits from extending to admissible space

As most of the motion planning problem formulation states, the configurations must lie in the open set of collision-free configurations \mathcal{C}_{free} . Also, it is well known that the exploration of classical sampling-based algorithms is very sensitive to the ϵ -goodness of the considered space. Although requiring only collision-free configurations can be an expectation from the application, we believe it is most commonly

used due to the challenges involved by including contacts. On the other side, taking advantage of the contact can also guide the exploration by allowing the tree to slide along the contact space.

Even if it may seem obvious, we emphasize that without contacts some motion planning scenarios for a rod cannot be solved. Indeed, as contacts add constraints to rod dynamics, it actually increases its deformation space by allowing new states that could not be reached without contacts. We will show in the results that allowing the contact space substantially changes the efficiency of our algorithm.

F. Discussion about completeness

Although our approach does not guarantee completeness on the state space \mathcal{X} , our main motivation is to use the admissible state space to improve the overall algorithm performances by allowing motions that can handle narrow passages much more efficiently. However, even if the generated roadmap will typically contain trajectories that does not entirely lie on the submanifold \mathcal{Y} , it can be shown that our approach is still probabilistically complete on \mathcal{Y} .

First, recall that desired start and goal states are quasi-static and collision free, i.e. $\mathbf{x}_{start} \in \mathcal{Y}$ and $X_{goal} \subset \mathcal{Y}$. Furthermore, states in $\mathcal{Y} = \mathcal{C}_{stable} \times SE(3)$ are sampled through coordinates in \mathcal{A}_{stable} , which is a global chart of \mathcal{C}_{stable} , along with a global parameterization of $SE(3)$. Then our sampling method enables a complete coverage of \mathcal{Y} .

The function ρ , defined in (3) and used as a pseudo-metric on \mathcal{X} , tends to its continuous equivalent $\bar{\rho}$ defined in (4) as the number of rod nodes N tends to infinity.

Proposition 1. *The function $\bar{\rho}$ is a metric on \mathcal{Y} .*

Proof: See Appendix A

We will then assume that the number of rod nodes N is sufficiently large so that ρ approximates properly $\bar{\rho}$.

From the dispersion of samples on \mathcal{Y} is given by

$$\delta(V, \rho) = \sup_{q \in \mathcal{Y}} \min_{v \in V} \rho(q, v)$$

where V are vertices of the exploration tree. Our planner restricted on \mathcal{Y} inherits the Voronoi bias and thus dispersion reduction of geometric RRTs, i.e.

$$\lim_{n \rightarrow \infty} \delta(V, \rho) = 0$$

where n is the number of planner iterations. It follows that the local planner tries to connect nodes that are closer (with respect to the metric ρ on \mathcal{Y}) as the number of iterations increases. Recall from IV-B that the local planner uses position control to connect two states. If the states to be connected are getting closer, they have a greater probability to be in the same attraction basin of elastic potential energy and then applying this control policy have greater probability to converge towards desired state.

Additionally, we have seen in section III-A that the mapping $\Phi \circ \Psi|_{\mathcal{A}_{stable}}$ between \mathcal{A}_{stable} and \mathcal{B}_{stable} , the set of corresponding rod extremity configurations $q(1)$, is a local diffeomorphism. As this result is not affected by the use of

free-floating rod base extremity, any infinitely small change in \mathcal{Y} leads to an infinitely small change in rod end extremities positions. Consequently, sampled end extremities positions also tend to be closer as the number of iterations grows.

Finally, each extension step $EXTEND_XDE(\mathbf{x}_{from}, \mathbf{x}_{to})$ will be performed by our local planner with a Proportional-Derivative controller on grippers position, i.e. the command is the sum of the linear interpolation of grippers position and velocity with fixed gains. Given \mathbf{x}_{to} and \mathbf{x}'_{to} sampled states in \mathcal{Y} and such that $\rho(\mathbf{x}_{from}, \mathbf{x}_{to}) < \rho(\mathbf{x}_{from}, \mathbf{x}'_{to})$, it is clear that as they correspond to null speed states, their contribution of the controller derivative part will be identical. However, the state \mathbf{x}_{to} which is closer to \mathbf{x}_{from} will lead to a lower control than \mathbf{x}'_{to} . The resulting local motion will have a lower speed and the corresponding trajectory in \mathcal{X}_{adm} will be closer to \mathcal{Y} . We can conclude that as number of iterations grows, samples will get closer and will be connected with trajectories that tends to be quasi-static, i.e. on the submanifold \mathcal{Y} .

Then, if a solution exists on \mathcal{Y} , the probability that our approach finds it tends to one as the number of planner iterations goes to infinity. This makes our algorithm probabilistically complete on \mathcal{Y} .

V. MOTION PLANNING WITH CONTACT INFORMATION

In the previous section, we have shown physical simulation can be used to slide along a contact, improving the extend method. However, we did not took into account any contact information to help the exploration. [3] use contact information to guide configuration sampling by tuning the growth method. In [23], the use of tangent contact plane enables to use a retraction-based extension and to improve the results. In these two works, the extension step is limited on the configuration space only, robot dynamics are ignored and only rigid articulated robot are considered. In our approach, the idea to exploit normal contact vector is also developed but contact nodes are created using a different method. Instead of retracting configurations to create contact nodes, we use dynamic simulation during the extension step. Such nodes are labelled differently in the roadmap and the extend method takes advantage of this as well as the number of contact points.

The physical engine XDE gives us several informations about the contact, i.e. the number of contacts, the contact points p_i and the contact normals \mathbf{n}_i . Given this contact information, we show in this section that this method offers a more efficient exploitation of the free space, leading to better performances for the planning algorithm. Previously introduced *RRT-GRIP* algorithm is replaced by *RRT-SLIDE* (Algorithm 2) which takes advantage of contact information.

The next subsections describe the different improvements of this algorithm. Firstly, we propose a new nearest neighbour search which takes into account the contact normal and the direction of the motion in order to reject invalid candidates. Secondly, we propose one method able to extend the roadmap from a configuration in contact.

Experimental results (see Section VI) show the benefit of the sliding mode.

A. Nearest neighbour modification

The first modification concerns the nearest neighbor search. Originally, the search only consider the distance between two states, according to the metric. But thanks to the additional contact information, the search can be improved. Indeed, intuitively a motion in the opposite direction of the contact normal (assuming that the normals are directed towards the exterior of the obstacles) is infeasible, as the rod will be pushed against the obstacle.

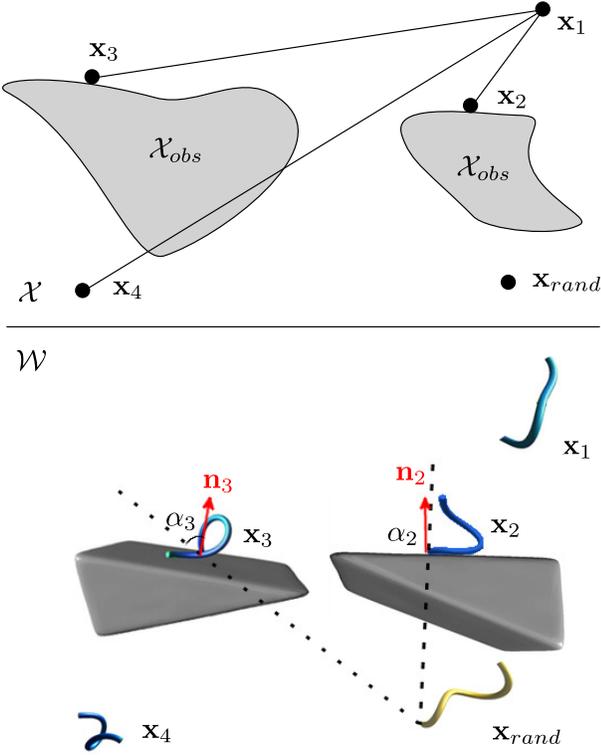


Figure 8: Example of situation during nearest neighbour search (top: configuration space, bottom : workspace). Each node \mathbf{x}_i in the roadmap represents a state \mathbf{x}_i of the rod. Red arrows are the contact normals and dashed lines are the direction of the desired motion of the gripper.

We introduce the value α which describes the angle between the direction of the contact normal \mathbf{n} and the direction of the desired motion for a given gripper $\mathbf{v} = (\mathbf{p}_{near} - \mathbf{p}_{rand})$ with \mathbf{p}_{near} (respectively \mathbf{p}_{rand}) the position of the rod for given gripper index in \mathbf{x}_{near} (respectively \mathbf{x}_{rand}). If α is close to zero, the force applied to the gripper would be orthogonal to the obstacle surface, and a call to the local planner from \mathbf{x}_{near} to \mathbf{x}_{rand} would most likely fail (i.e. return a state really close to \mathbf{x}_{near}). These cases should be avoided due to their high computational cost as they correspond to narrow phases of the collision detection. Consequently, the nearest node is ignored if the α is under a threshold and a farther node with higher chance to success is considered. This is illustrated in Figure 8 where the nearest configuration of \mathbf{x}_{rand} is \mathbf{x}_2 but the value of α_2 is close to zero. In this case, the nearest neighbour search will return the configuration \mathbf{x}_3 which is farther but have a higher probability to connect \mathbf{x}_{rand} .

This allows to reduce the number of useless and expensive calls to the local planner and to explore new trajectories between nodes that would have been ignored otherwise. This new nearest neighbour function, *NEAREST_ADMISSIBLE*, is called by line 8 in algorithm 2.

B. Contact sliding motions

The rod is already allowed to slide along the contact space by using XDE as local planner [7], but thanks to contact information, the exploration of the state space can be improved. At random intervals, a node of the tree among one of labeled as contact nodes is selected and an extend phase from this node of the tree is tried, as detailed in line 5 in Algorithm 2. The probability $P_{contact}$ is defined using the ratio of number of nodes in contact space and in free space in the roadmap. Given this probability, the function *SLIDE_CONTACT* is randomly called instead of the classic *EXTEND_XDE* function.

Algorithm 2 RRT-SLIDE($\mathbf{x}_{start}, X_{goal}$)

Output: Solution trajectory τ or fail

- 1: Initialize the tree \mathcal{T} with \mathbf{x}_{start}
 - 2: **while** \neg solved **and** $iter < N_{max}$ **do**
 - 3: $P \leftarrow \text{RANDOM}([0, 1])$
 - 4: **if** $P < P_{contact}$ **then**
 - 5: $\mathbf{x}_{new} \leftarrow \text{SLIDE_CONTACT}()$
 - 6: **else**
 - 7: $\mathbf{x}_{rand} \leftarrow$ random quasi-static state $\in \mathcal{Y}$ or state in X_{goal}
 - 8: $\mathbf{x}_{near} \leftarrow \text{NEAREST_ADMISSIBLE}(\mathcal{T}, \mathbf{x}_{rand})$
 - 9: $\mathbf{x}_{new} \leftarrow \text{EXTEND_XDE}(\mathbf{x}_{near}, \mathbf{x}_{rand})$
 - 10: Add vertex \mathbf{x}_{new} to \mathcal{T}
 - 11: Add edge $(\mathbf{x}_{near}, \mathbf{x}_{new})$ to \mathcal{T}
 - 12: **end if**
 - 13: **end while**
-

The method designed in *SLIDE_CONTACT* is based on common human behavior for this kind of manipulation task : when the rod must be moved to the other side of an obstacle or to guide it through a narrow passage, we don't try to avoid contact between the rod and the obstacle. Instead, we move the rod along the surface of the obstacle and make it slide on the environment until the obstacle edge is reached.

In line 2 of Algorithm 3, *CONTACT_INFORMATION* returns the contact with the highest contact force and also information about the different constraints applied by the other contacts.

Let \mathbf{n} be the contact normal and M the contact point. These informations are given by XDE and can be used to compute a local and linear approximation of the obstacle surface. In the current implementation, the contact with the highest contact force is used for this method. Other choices are possible, such as considering all the contacts to add constraints (see [2]) and sampling in a more constrained lower dimensionnal space.

The function *SAMPLE_ON_PLANE* line 3 of Algorithm 3 samples a 6D position in the plan which approximates locally the contact surface. The lines 6-8 of Algorithm 3 are similar to a call to the local planner. The input for the

Algorithm 3 SLIDE_CONTACT

Input: Roadmap \mathcal{T} , time limit t_{max}

Output: A new node \mathbf{x}_{new}

- 1: Select at random a node in contact $\{\mathbf{x}_c \in \mathcal{T} \mid \mathbf{x}_c \in \partial\mathcal{X}_{obs}\}$
 - 2: $(\mathbf{n}, M) \leftarrow \text{CONTACT_INFORMATION}(\mathbf{x}_c)$
 - 3: $p_{proj} \leftarrow \text{SAMPLE_ON_PLANE}(\mathbf{n}, M)$
 - 4: $g_i \leftarrow \text{SELECT_GRIPPER}(M)$
 - 5: $\mathbf{x}_{new} \leftarrow \mathbf{x}_c$
 - 6: **while** gripperDistance($\mathbf{x}_{new}, p_{proj}, g_i$) $> \epsilon$ **and** $t < t_{max}$ **and** isInContact(\mathbf{x}_{new}, g_i) **do**
 - 7: Apply position control on \mathbf{x}_{new} at g_i to p_{proj} for Δt
 - 8: **end while**
 - 9: Add current state \mathbf{x}_{new} to \mathcal{T}
 - 10: Add edge $(\mathbf{x}_c, \mathbf{x}_{new})$
 - 11: **return** \mathbf{x}_{new}
-

position controller is a single 6D position for the actuated gripper. The function *gripperDistance* returns the distance between the actuated gripper and desired 6D position, using a classical metric on $SE(3)$. Moreover, a specific terminal condition is added with the *isInContact* method. If the part of the rod near the actuated gripper leaves the surface of the obstacle, the local planner is interrupted and a node associated to the current state is created. Indeed, most of the time this mean that the rod have crossed the edge of the obstacle and a node with this state is strategically well placed, allowing a better exploration of the other side of the obstacle or to find a narrow passage. As is the local method, it is necessary to select a gripper to apply the local controller (*SELECT_GRIPPER*). In Algorithm 1, we select only the end of the rod, but in Algorithm 3 it is possible to take into account several gripper positions along the rod and to take the nearest to the contact point.

VI. RESULTS AND DISCUSSION

In this section, we present and discuss experimental results in simulation obtained on industrial and academic cases. A typical application to motion planning of elastic rods consists in assembly and disassembly studies of flexible parts such cables. Then, the following scenarios have been tested:

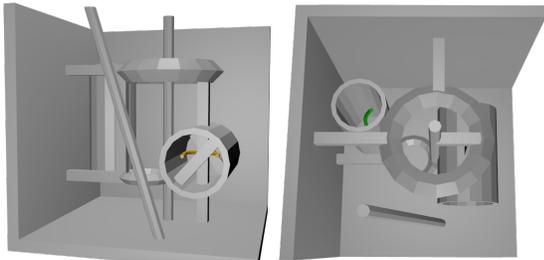


Figure 10: *Abstract* scenario with start (yellow) and goal (green) states.

- *Engine* (Figures 1, 9) represents an industrial disassembly scenario with an highly constrained initial state. This scenerio is one of the real use cases falling within

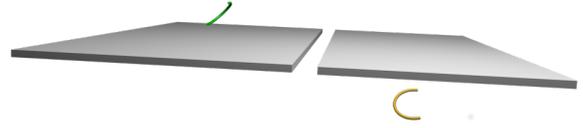


Figure 11: *Crack* scenario with start (yellow) and goal (green) states.

the scope of the industrial project with Siemens PLM Software.

- *Abstract* (Figure 10) shows several difficult passages and both goal and initial states are constrained (model taken from OMPL [24]).
- *Crack* (Figure 11) is a toy scenario with a crack-shaped narrow passage on a plane surface.
- *Hole* (Figure 12) is another toy scenario with a hole-shaped narrow passage on a plane surface. Here, goal and initial states are very close in the state space but not in the free space.



Figure 12: *Hole* scenario with start (yellow) and goal (green) states.

Each scenario has been run 100 times with different random seeds and a limit in time of one hour, except for the *Engine* scenario (50 runs and 2 hours time limit). All benchmarks were run on a PC with 64GB of main memory and using one core of an Intel Xeon E5-1630 processor running at 3.7GHz. Implementation has been done in C++ using the Humanoid Path Planner framework (HPP) [25].

The results were recorded, analyzed and displayed with the framework Planner Arena [26]. Parameters used for this benchmark and results details can be found here ¹. The attached video shows planning results on these scenarios.

For collision checking, the geometry of the rod was approximated by a chain of capsules (i.e. Line Swept Spheres) whose volume is overestimated by a small tolerance threshold. Using different threshold values, a hierarchical chain of capsules can be obtained (see figure 13). The Flexible Collision Library (FCL) [27] was used to perform collision checking computations, i.e. collision between the rod and obstacles and self-collision. The number of capsules at the lowest level of the hierarchy was fixed to 50, which corresponds to the number N of rod finite elements in XDE.

In the following results, the planner *RRT_GRIP* is the one presented in [7], which is based on models presented in section III and uses XDE as local planner (as presented in section IV). This planner already solves efficiently scenarios showing difficult narrow passages and provides a solution trajectory

¹<https://laas-fernbach.shinyapps.io/plannerarena/>

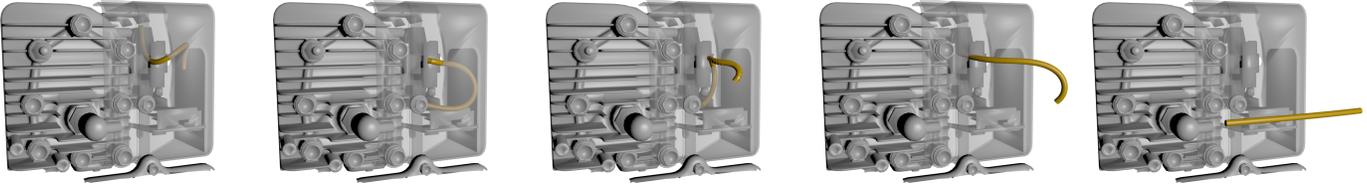


Figure 9: Extraction of a deformable elastic rod from an engine. The yellow rod shows the motion computed from the initial configuration (left picture). View has been clipped for clarity.

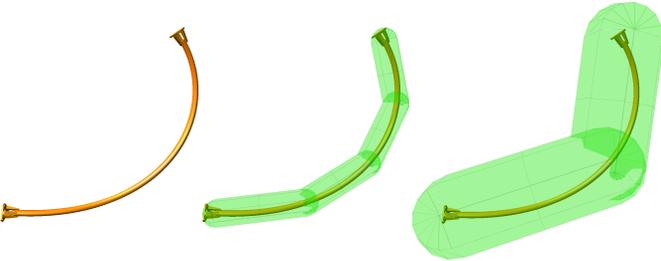


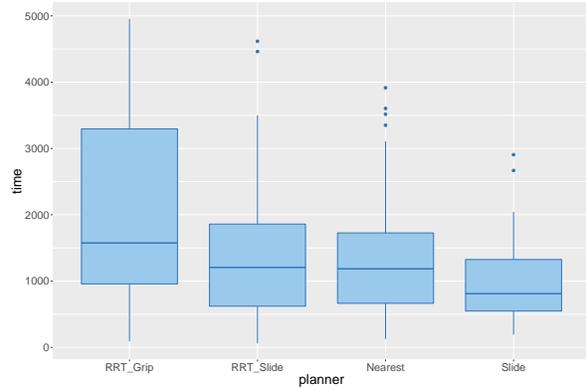
Figure 13: Hierarchical chain of capsules increasingly coarse.

with accurate rod dynamics, thanks to the simulation based on finite elements. The planner *RRT_SLIDE* is the one presented in this paper. It is based on *RRT_GRIP* and includes the modifications presented in section V. The planners *Nearest* and *Slide* show isolated impacts of each improvement proposed in sections V-A and V-B, respectively. Note that none of these scenarios can be solved in a reasonable amount of time with classical planners that only consider the free space as admissible.

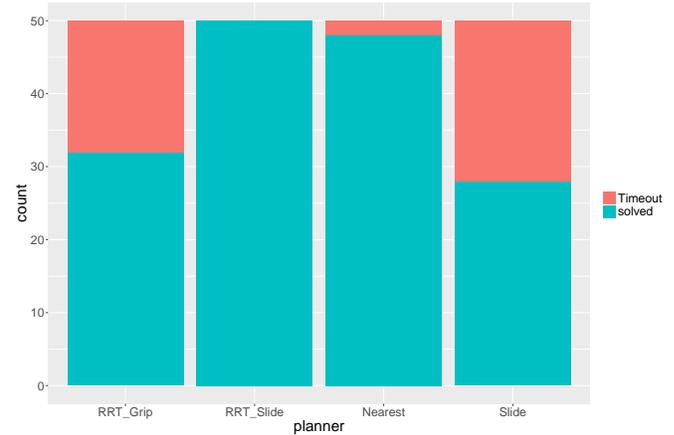
Figures 14a and 14b show the resolution time and the success rate for the *Engine* scenario. This scenario has a highly constrained initial state and a narrow passage towards the goal state. The modification of the nearest neighbour search increases the success rate of the algorithm, as it reduces the probability that the exploration tree remains stuck and enables a better exploration of the free space. By adding the sliding motions method, a success rate of 100% is reached and the resolution time is reduced by 27%. This method helps significantly the planner to find narrow passages by sliding in the contact space.

In the *Abstract* scenario (figures 15a and 15b), the reference algorithm always fails. Using the sliding motions method gives a faster resolution time but with a success rate of only 25%. The nearest neighbour modification increases significantly the success rate (only 10% of failures) at the cost of the resolution time. This is due to the fact that the sliding motion explores quickly the contact space while the nearest neighbour modification explores more free space and reduces the number of edges of low interest in the tree.

Crack (figures 16a and 16b) and *Hole* (17a and 17b) scenarios show similar features, i.e. one narrow passage and planar surfaces. These scenarios show that the sliding motion is really efficient in the case of contacts with planar surfaces, as they enable a fast exploration of the surface and thus can find the narrow passage quickly. However, with the only addition



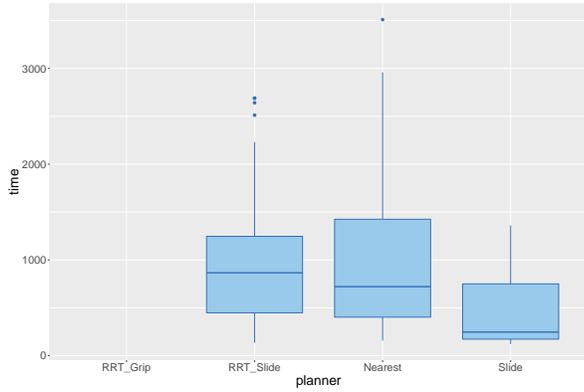
(a) Resolution time (s) for the *Engine* scenario.



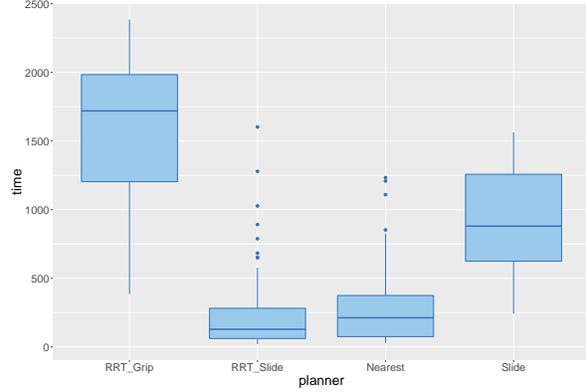
(b) Success rate for the *Engine* scenario.

Figure 14: Benchmark for *Engine*. *RRT_GRIP* : previous planner presented in [7], *RRT_SLIDE* : planner presented in this paper, *Nearest* : *RRT_GRIP* with contribution of section V-A, *Slide* : *RRT_GRIP* with contribution of section V-B.

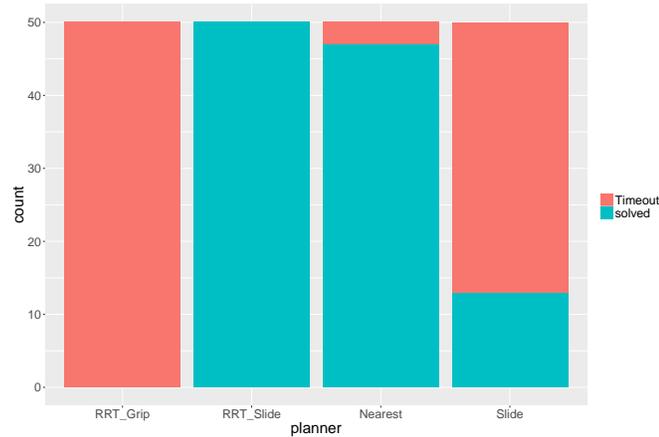
of sliding motion, the success rate is still low. This is because of the way the nearest neighbour is selected in the previous algorithm. Indeed, when considering only a distance parameter between two states, the probability to select a state on the wrong side of the narrow passage is high, especially in the *Hole* scenario where the initial and goal states are really close, but on opposite sides of the narrow passage. The roadmap created in this scenario shows an high concentration of nodes that are close to the goal state but on the wrong side of the narrow passage. This is inefficient as a lot of computational resources are wasted to compute these nodes and edges. The



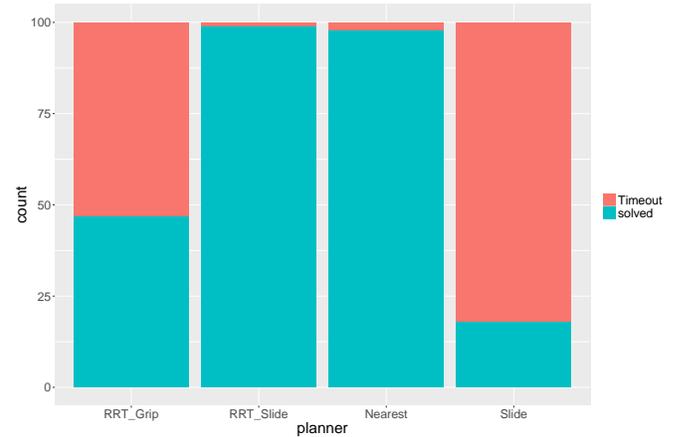
(a) Resolution time (s) for the *Abstract* scenario.



(a) Resolution time (s) for the *Crack* scenario.



(b) Success rate for the *Abstract* scenario.



(b) Success rate for the *Crack* scenario.

Figure 15: Benchmark for *Abstract*. *RRT_GRIP* : previous planner presented in [7], *RRT_SLIDE* : planner presented in this paper, *Nearest* : RRT_GRIP with contribution of section V-A, *Slide* : RRT_GRIP with contribution of section V-B.

Figure 16: Benchmark for *Crack*. *RRT_GRIP* : previous planner presented in [7], *RRT_SLIDE* : planner presented in this paper, *Nearest* : RRT_GRIP with contribution of section V-A, *Slide* : RRT_GRIP with contribution of section V-B.

modification of the nearest neighbour search proposed here allows the algorithm to select states in the same side of the narrow passage with an higher probability. This explains the significant increase of the success rate for both of these scenarios when using the new nearest neighbour search.

A. Discussion

Overall, it can be observed that the modification of the nearest neighbour search always increases the planner success rate, and also sometimes increases the resolution time. This is explained by the fact that more possibilities are explored when linking two nodes together which reduces the probability that the tree remains stuck. The addition of contact sliding motions reduces the planner resolution time because the contact space is considerably more explored and this allows the planner to find narrow passages much faster. It can also be noted that the planner *RRT_SLIDE* which uses both of these improvements always performs better than the planners with only one of these modifications, underlying that these improvements are complementary.

These results, summarized in Table I, show that the improvement proposed in this article allows our planner to always perform better than the one proposed in [7].

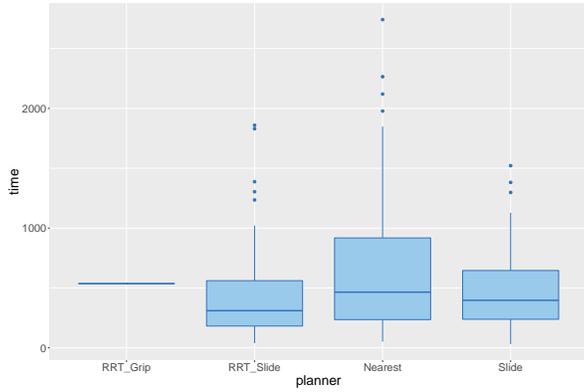
Furthermore, it should be noticed that in the scenarios where the rod is always in the free space (i.e. no need to make any contacts to reach the goal), performances will be the same than with the previous planner. Indeed, the improvements that we propose here are only triggered when the rod is in contact.

Table I: Kinodynamic planning for elastic rods with contacts results

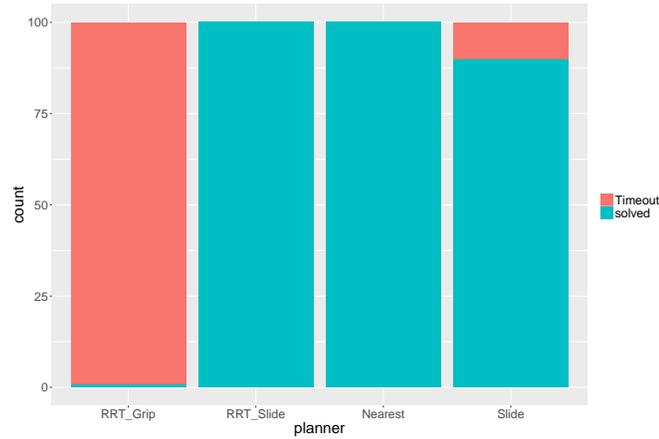
Scenario	Success rate		avg resolution time (s)	
	GRIP	SLIDE	GRIP	SLIDE
Crack	47%	99%	1592.8	233.6
Hole	1%	100%	536.2	411.3
Abstract	0%	100%	x	984.6
Engine	64%	100%	2073.3	1417.1

Figure 18 shows that taking into account sliding motion along surfaces drastically reduces the number of nodes.

One of the advantages of this algorithm is that its efficiency does not rely on a parameter tuning that is specific to a given



(a) Resolution time (s) for the *Hole* scenario.



(b) Success rate for the *Hole* scenario.

Figure 17: Benchmark for *Hole*. *RRT_GRIP* : previous planner presented in [7], *RRT_SLIDE* : planner presented in this paper, *Nearest* : *RRT_GRIP* with contribution of section V-A, *Slide* : *RRT_GRIP* with contribution of section V-B.

scenario. All benchmarks are done with the same motion planning parameters (see Planner Arena parameters). Evidently, better results could have been obtained if the parameters were adjusted individually for each scenario. As an example, in the *Engine* scenario, the resolution time can be divided at least by two using a tuned parameters set. Obviously, when considering a different rod, it is necessary to update its material elasticity parameters.

In order to improve overall performances, one possibility could be to use a more efficient simulator than XDE as long as sliding motion can be computed while taking into account rod elasticity parameters. If the simulator can model friction, then our approach can easily ensure that rod elasticity limits are also satisfied. This could be achieved, for example, by stopping the extension step if one of the rod element constraint gets out of its elasticity domain and adjusting the nearest neighbor search metric to avoid getting stuck. Another possibility to improve performances could be to take advantage of analytic solutions for the rod SEM in the 3-dimensionnal case. However, only the costate solution can be computed under the transverse isotropic material assumption. It should be noted that for the planar rod case (e.g. a metallic strip), we gave analytic forms of the DGM

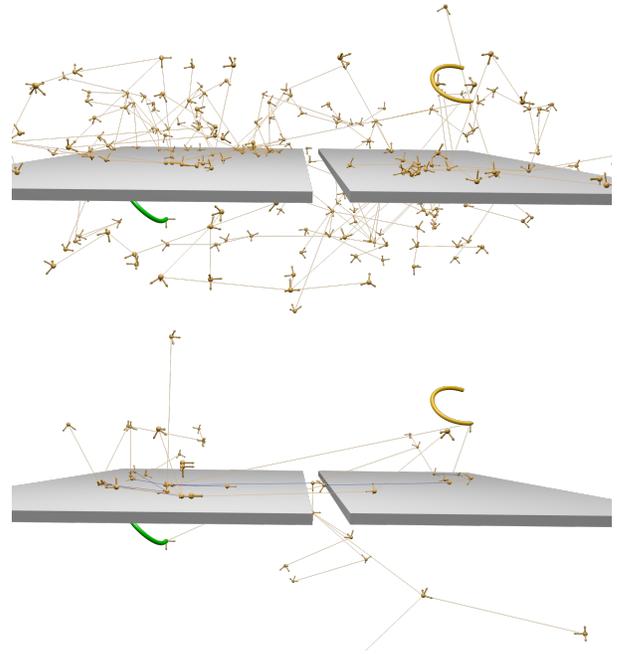


Figure 18: Difference between *RRT_GRIP* (top) and *RRT_SLIDE* (bottom) roadmaps. Blue edges are found with the *SLIDE_CONTACT* method.

[28], enabling an extremely fast computation (100 times faster) and avoiding numerical robustness issues.

VII. CONCLUSION

In this paper, we propose an approach to the motion planning problem for an extensible elastic rod in collision-free or contact space. It is shown that by using a static equilibrium model, we can efficiently sample on the submanifold of its quasi-static contact-free states whereas the kinodynamic planning algorithm uses dynamic simulation and explores the full state space of the rod. Thanks to dynamic simulation, the planning algorithm can take advantage of contacts by allowing sliding motions in order to solve difficult narrow passages, where typical sampling-based method working in collision-free space would fail. Furthermore, by taking into account contact information such as forces at contact points, this approach improves the roadmap exploration of the admissible space and consequently, increases the planning success rate.

One purpose of Product Lifecycle Management (PLM) software is the simulation of assembly/disassembly tasks to validate the final assembly process or the maintainability of a complex product, like a car or an aircraft. Assembly/disassembly scenarios showing difficult narrow passages are solved efficiently using our approach. Additionally, a solution trajectory with accurate rod dynamics is available thanks to the simulation based on finite element analysis. We emphasize that our algorithm does not require any parameter tuning specific to the scenario. This approach can be integrated into PLM tools in order to help the non-expert user to solve the motion planning problem. This solution includes the sequence of controls that should be applied to the rod and this information could be used in realistic conditions.

Despite the fact that manipulator paths are not considered here, our approach can be used as a first step to solve the general manipulation problem. Given the manipulators end-effector positions, velocities and wrenches provided by the solution trajectory, several methods are available to plan the motion of the manipulators. Also, note that our method provides a necessary condition for the existence of a solution to the general manipulation problem.

Handling more general cases, such as a different number of grippers and their position along the rod, could be done by a higher level manipulation planning algorithm which could be coupled with our approach to obtain a realistic application.

APPENDIX

Let's show that the function $\bar{\rho}$ defined in (4) is a metric on \mathcal{Y} . First, recall that $\mathbf{p} \in \mathbb{R}^3$ denotes the translation part of a configuration $q \in SE(3)$, and let us note $\mathbf{R} \in SO(3)$ the rotational part. Let $\mathcal{P}_{\text{stable}}$ be the selection of the translation part of all static equilibrium rod configurations, i.e.

$$\mathcal{P}_{\text{stable}} = \{\mathbf{p}(q) | (q, u) \in \mathcal{C}_{\text{stable}}\}$$

Let now $\varphi : \mathcal{A}_{\text{stable}} \rightarrow \mathcal{P}_{\text{stable}}$ be the map between coordinates of static equilibrium rod configurations and the translation part of their geometry.

Lemma 1. *If $\varphi(\mathbf{a}) = \varphi(\mathbf{a}') \forall \mathbf{a}, \mathbf{a}' \in \mathcal{A}_{\text{stable}}$, then $\mathbf{a} = \mathbf{a}'$.*

Proof. Equation (1) can be rewritten in terms of rotational and translational part by

$$\dot{\mathbf{p}} = \mathbf{R} \begin{pmatrix} 1 + u_4 \\ u_5 \\ u_6 \end{pmatrix} \quad \dot{\mathbf{R}} = \mathbf{R} \begin{pmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{pmatrix} \quad (5)$$

It is clear that \mathbf{p} is smooth, so it suffices to show that \mathbf{a} is uniquely defined by \mathbf{p} and its derivatives.

Recall that $u_i = c_i^{-1} \mu_i$ for $i \in \{1, \dots, 6\}$ and $a = \mu(0)$. Also, without loss of generality, we can take here $q(0) = e$ where e is the identity element of $SE(3)$. Now taking $t = 0$, (5) becomes

$$\dot{\mathbf{p}}(0) = \begin{pmatrix} 1 + c_4^{-1} a_4 \\ c_5^{-1} a_5 \\ c_6^{-1} a_6 \end{pmatrix} \quad (6)$$

Differentiating and using (5), we have

$$\ddot{\mathbf{p}} = \mathbf{R} \begin{pmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{pmatrix} \begin{pmatrix} 1 + u_4 \\ u_5 \\ u_6 \end{pmatrix} + \mathbf{R} \begin{pmatrix} \dot{u}_4 \\ \dot{u}_5 \\ \dot{u}_6 \end{pmatrix} \quad (7)$$

From Theorem 1 in [7], we have at $t = 0$

$$\begin{aligned} \dot{\mu}_4(0) &= c_3^{-1} a_3 a_5 - c_2^{-1} a_2 a_6 \\ \dot{\mu}_5(0) &= c_1^{-1} a_1 a_6 - c_3^{-1} a_3 a_4 \\ \dot{\mu}_6(0) &= c_2^{-1} a_2 a_4 - c_1^{-1} a_1 a_5 \end{aligned} \quad (8)$$

Define the constants β_i for $i \in \{1, 2, 3\}$ by

$$\begin{aligned} \beta_1 &= c_1^{-1} (c_5^{-1} - c_6^{-1}) \\ \beta_2 &= c_2^{-1} (c_6^{-1} - c_4^{-1}) \\ \beta_3 &= c_3^{-1} (c_4^{-1} - c_5^{-1}) \end{aligned}$$

Using (8) and taking $t = 0$, (7) becomes

$$\ddot{\mathbf{p}}(0) = \begin{pmatrix} \beta_3 a_3 a_5 + \beta_2 a_2 a_6 \\ c_3^{-1} a_3 + \beta_3 a_3 a_4 + \beta_1 a_1 a_6 \\ -c_2^{-1} a_2 + \beta_2 a_2 a_4 + \beta_1 a_1 a_5 \end{pmatrix} \quad (9)$$

The system of equations given by (6) and (9), which relates $\dot{\mathbf{p}}(0)$ and $\ddot{\mathbf{p}}(0)$ to coordinates \mathbf{a} has clearly a unique solution. Then, φ is injective. \square

Lemma 2. *The map $\varphi : \mathcal{A}_{\text{stable}} \rightarrow \mathcal{P}_{\text{stable}}$ is bijective.*

Proof. As the map $\phi : \mathcal{A}_{\text{stable}} \rightarrow \mathcal{C}_{\text{stable}}$ is a homeomorphism, by construction, φ is clearly surjective. From Lemma 1, φ is also injective. Then, φ is bijective. \square

Proposition 1. *The function $\bar{\rho}$ being defined by*

$$\bar{\rho}(x_i, x_j) = \int_0^1 \|\mathbf{p}_i(t) - \mathbf{p}_j(t)\| dt \text{ is a metric on } \mathcal{Y}.$$

Proof.

- 1) *Non-negativity and symmetry.* From its definition, it is clear that $\bar{\rho}$ is non-negative and symmetric.
- 2) *Triangle inequality.* Proving triangle-inequality is also straightforward. Given $x_i, x_j, x_k \in \mathcal{Y}$, we have

$$\|\mathbf{p}_i(t) - \mathbf{p}_j(t)\| + \|\mathbf{p}_j(t) - \mathbf{p}_k(t)\| \geq \|\mathbf{p}_i(t) - \mathbf{p}_k(t)\|, \forall t \in [0, 1]$$

Integrating along the rod:

$$\int_0^1 \|\mathbf{p}_i(t) - \mathbf{p}_j(t)\| dt + \int_0^1 \|\mathbf{p}_j(t) - \mathbf{p}_k(t)\| dt \geq \int_0^1 \|\mathbf{p}_i(t) - \mathbf{p}_k(t)\| dt$$

Then we have $\bar{\rho}(x_i, x_j) + \bar{\rho}(x_j, x_k) \geq \bar{\rho}(x_i, x_k)$

- 3) *Identities of indiscernibles.* From Lemma 2, we know the map $\varphi : \mathcal{A}_{\text{stable}} \rightarrow \mathcal{P}_{\text{stable}}$ is bijective. This means the translation part of the rod geometry at static equilibrium (i.e. in \mathcal{Y}) suffices to uniquely define its configuration. It follows that $\bar{\rho}(x_i, x_j) = 0$ if and only if $x_i = x_j$, $\forall x_i, x_j \in \mathcal{Y}$. \square

ACKNOWLEDGMENT

The authors would like to thank Siemens PLM Software for engine industrial model.

REFERENCES

- [1] J. J. Kuffner and S. M. LaValle, "Rrt-connect: An efficient approach to single-query path planning," in *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, vol. 2, 2000, pp. 995–1001 vol.2.
- [2] S. Redon and M. C. Lin, "Practical local planning in the contact space," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 2005.
- [3] Rodriguez, X. Tang, J.-M. Lien, and N. M. Amato, "An obstacle-based rapidly-exploring random tree," in *Proc. IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, May 2006, pp. 895–900.
- [4] Z. Sun, D. Hsu, T. Jiang, H. Kurniawati, and J. Reif, "Narrow passage sampling for probabilistic roadmap planners," *IEEE Trans. on Robotics*, vol. 21, no. 6, pp. 1105–1115, 2005.
- [5] T. Bretl and Z. McCarthy, "Quasi-static manipulation of a kirchhoff elastic rod based on a geometric analysis of equilibrium configurations," *Int. Jour. Robotic Research*, vol. 33, no. 1, 2014.

- [6] B. Frank, C. Stachniss, N. Abdo, and W. Burgard, "Efficient motion planning for manipulation robots in environments with deformable objects," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2011.
- [7] O. Roussel, A. Borum, M. Taix, 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 Robotics and Automation*, 2015.
- [8] F. Lamiroux and L. E. Kavraki, "Planning paths for elastic objects under manipulation constraints," *Int. Jour. Robotic Research*, vol. 20, no. 3, 2001.
- [9] M. Moll and L. E. Kavraki, "Path planning for deformable linear objects," *IEEE Trans. on Robotics*, vol. 22, no. 4, 2006.
- [10] H. Wakamatsu and S. Hirai, "Static modeling of linear object deformation based on differential geometry," *Int. Jour. Robotic Research*, vol. 23, no. 3, 2004.
- [11] S. Rodríguez, J.-M. Lien, and N. M. Amato, "Planning motion in completely deformable environments," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 2006.
- [12] O. B. Bayazit, J.-M. Lien, and N. M. Amato, "Probabilistic roadmap motion planning for deformable objects," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 2002.
- [13] N. Amato and Y. Wu, "A randomized roadmap method for path and manipulation planning," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 1996.
- [14] R. Gayle, P. Segars, M. C. Lin, and D. Manocha, "Path planning for deformable robots in complex environments," in *Proc. Robotics: Science and Systems*. The MIT Press, 2005.
- [15] I. Kabul, R. Gayle, and M. C. Lin, "Cable route planning in complex environments using constrained sampling," in *Proc. ACM Symposium on Solid and Physical Modeling*, 2007.
- [16] A. Mahoney, J. Bross, and D. Johnson, "Deformable robot motion planning in a reduced-dimension configuration space," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 2010.
- [17] O. Roussel, M. Taix, and T. Bretl, "Efficient motion planning for quasi-static elastic rods using geometry neighborhood approximation," in *Proc. IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics*, 2014.
- [18] X. Merlhiot, J. L. Garrec, G. Saupin, and C. Andriot, "The xde mechanical kernel: Efficient and robust simulation of multibody dynamics with intermittent nonsmooth contacts," in *Proc. Second Joint Int. Conf. on Multibody System Dynamics*, 2012.
- [19] L. E. Kavraki, J.-C. Latombe, R. Motwani, and P. Raghavan, "Randomized query processing in robot path planning," *Jour. of Computer and System Sciences*, vol. 57, no. 1, 1998.
- [20] J. Simo, "A finite strain beam formulation. the three-dimensional dynamic problem. part i," *Computer methods in applied mechanics and engineering*, vol. 49, no. 1, 1985.
- [21] S. M. LaValle and J. J. Kuffner, "Randomized kinodynamic planning," *Int. Jour. of Robotics Research*, vol. 20, no. 5, 2001.
- [22] D. Berenson, S. Srinivasa, D. Ferguson, and J. J. Kuffner, "Manipulation planning on constraint manifolds," in *Proc. IEEE Inter. Conf. on Robotics and Automation*, 2009.
- [23] J. Pan, L. Zhang, and D. Manocha, "Retraction-based rrt planner for articulated models," in *IEEE Int. Conf. on Robotics and Automation*, May 2010, pp. 2529–2536.
- [24] I. A. Sucan, M. Moll, and L. E. Kavraki, "The open motion planning library," *IEEE Robotics & Automation Magazine*, vol. 19, no. 4, 2012.
- [25] J. Mirabel, S. Tonneau, P. Fernbach, A.-K. Seppälä, M. Campana, N. Mansard, and F. Lamiroux, "HPP: a new software for constrained motion planning," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2016.
- [26] M. Moll, I. A. Sucan, and L. E. Kavraki, "Benchmarking motion planning algorithms: An extensible infrastructure for analysis and visualization," *IEEE Robotics & Automation Magazine*, vol. 22, no. 3, 2015.
- [27] J. Pan, S. Chitta, and D. Manocha, "Fcl: A general purpose library for collision and proximity queries," in *Proc. IEEE Inter. Conf. on Robotics and Automation*, 2012.
- [28] O. Roussel, M. Renaud, and M. Taix, "Closed-forms of planar kirchhoff elastic rods: application to inverse geometry," in *Proc. IMA Conference on Mathematics of Robotics*, 2015.