Motion Planning for a Deformable Linear Object

Olivier Roussel[1], Michel Taïx[1] and Timothy Bretl[2]

[1]: CNRS, LAAS, Univ de Toulouse, 7 avenue du colonel Roche, F-31400 Toulouse, Cedex 4, France,
[2]: Department of Aerospace Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA

***Abstract*— We present a motion planning algorithm for quasi-static Kirchhoff elastic rod in complex environments. By considering the set of quasi-static deformations which forms a finite dimensional manifold that can be parameterized by a single chart, the configuration space formulation extends nicely to this deformation space. As this parameterization is computationally expensive, our algorithm takes advantage of its smooth structure to approximate fast its neighborhood. Our approach enables motion planning for physically realistic deformable models for rods and our algorithm gives a significant performance improvement over classical motion planning approaches. We demonstrate the effectiveness of our approach on various academic and industrial scenarios, including difficulties such as narrow passages and high number of primitives.**

I. INTRODUCTION

Motion planning in Robotics has been extensively studied from last decades. But most of the work focused on rigid bodies and relatively little work has been given to extension of the problem to a deformable robot. Recent applications of robotics algorithmic in various fields such as virtual prototyping have brought new motivations in this direction. It is impossible to correctly modelize industrial case (used in automotive and aeronautics industry) whitout deformable parts. These parts are not handled yet due to the lack of efficient motion planning algorithms and this is becoming a major limitation.

In this paper we consider only the case of "Deformable Linear Objects", (DLOs), robot in a rigid environment. DLOs are characterized by having one dimension much more greater than the two others (cable, hose, pipe,...).

For a rigid body, the configuration is defined by the finite number of parameters (here 6) that defines the frame, whereas for a deformable robot the number of shapes given a fixed frame is infinite. As deformation can be seen as adding new degrees of freedom to the system, the extension to current configuration space based formulation might seem straightforward. However, working with an inefficient parameterization of a deformation would lead to a high number of dependent degrees of freedom. This involves a high-dimensional configuration space although the set of deformations describes a much lower-dimensional manifold. Also, the deformation model that would describe how to get from the configuration space to the workspace may be hard to compute.

Flexible rods mechanics have been extensively studied, especially in the case of elastic deformations [1] [16] [10] [3]. These works offer relatively accurate dynamic models of an elastic rod, but the computational cost induced by the use of numerical methods such as finite elements is too high to include them directly into a motion planning framework. For instance, in [15], Rodriguez et al. coupled a deformable dynamics simulator with a kynodynamic 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.

As in our context of assembling and disassembling studies the goal is to find a geometric path for the rod, the use of quasi-static models seems a reasonable assumption. In this direction, the work from Lamiraux and Kavraki [9] investigated for the manipulation planning problem of deformable objects based on the computation of their equilibrium configurations. This computation relies on the numerical minimization of the total elastic energy for given gripper placements. Some efforts have been done by Moll and Kavraki [12] in the special case of inextensible DLOs to reduce the cost of the minimization by the use of recursive subdivision. Similarly, the work of Wakamatsu et al. [18] defines another static model of elastic rods which also relies on optimization techniques. In all the cited approaches, the use of numerical methods make them too computationally expensive.

A different category of approaches, motivated by applications in computer graphics, relies on simplified deformation models that do not take into account mechanical properties. For instance, in [2] Bayazit et al. pre-compute some reduced models of the deformable object and use them to build a weighted roadmap through Probalistic Roadmap (PRM) methods. In [6], Gayle et al. use the Constraint Based Motion Planning framework to simulate a deformable robot using a mass-spring model along a rough estimate of the solution path. Kabul et al. [8] extended this work to DLOs using a kinematic chain as deformable model. In these cases, the quality of the solution mostly depend on the estimate and might be unable to solve some cases.

Another interesting direction has been investigated by Mahoney et al. [11] by the use of a two-step process. A learning phase first collects high-dimensional samples (e.g. using simulation) and computes a new basis for the deformation set using linear dimensionality reduction. Then, this reduced deformation space is used as parameterization to perform motion planning. Its main drawback lies in the limitation to linear reduction and its nonlinear extension requires additional information on the model.

In conclusion, to overcome the curse of dimensionality, we need a model that respects the physical properties of deformations, but also minimizes the number of model parameters.

This work tries to tackle all the drawbacks of the mentioned approaches. We build on top of recent work from Bretl and McCarthy [5], which directly provides a parameterization of the finite-dimensional manifold of equilibrium configurations for a inextensible Kirchhoff elastic rod. This enables faster computation of deformation states and, coupled with neighborhood information, we show that it can be efficiently used for motion planning.

II. Approach

The work from [4] offers a single global chart to describe the manifold of equilibrium configurations of an elastic rod. As this chart is a low dimensional Euclidean space, it is especially well suited for sampling-based methods. However, computing the parameterization of this manifold which gives us the Direct Geometric Model (DGM) is computationally expensive with a realistic number of nodes to modelize the rod (between 100 to 300 nodes). In this case, computing time of DGM is the same order than collision checking and this constrain imply a new bottleneck for motion planning.

Our approach approximates the neighborhood of this parameterization and enables fast collision checking in this neighborhood, which is applied as an efficient approach to motion planning for free-flying quasi-static elastic rods. This approximation is based on the computation of the Jacobian at each node of the cable. Using approximate rod geometry is 10 times faster than the full DGM computation. Then, it can be efficiently used to validate a local path.

An important point is that our approximation is close to the exact solution. We show that the approximation error induced by Jacobian relative to the distance in the vicinity of the configuration space is small

This low error allows us to use the approximation of the model in the exploration of the configuration space efficiently. We present our motion planning algorithm based on this Jacobian approximation.

III. Motion Planning for Quasi-Static Elastic Rods

A. Problem Statement

Let R be a robot having n independent degrees of freedom. The configuration space $\mathcal{C}$ of R is an n-manifold consisting in the set of all configurations of R. Then the classic motion planning problem can be stated as finding a continuous path $\tau : [0,1] \rightarrow \mathcal{C}_{free}$, where $\mathcal{C}_{free} \subseteq \mathcal{C}$ is the set of *valid* configurations. In general, a configuration is *valid* if it is collision-free in the workspace.

Consider now that the robot R is deformable, and more specifically it is an elastic Kirchhoff rod in quasi-staticity. [5] showed that the set of deformations $\mathcal{D}$ is an m-manifold, then it is possible to apply the same formulation.

Sampling-based methods relies on a parameterization of the manifold. Unfortunatly, finding a global parameterization of a manifold is not always possible without going to higher dimensions (e.g. the Lie Group $SO(3)$). As $\mathcal{D}$ is a manifold, it can be described by an atlas $\{(\phi_\alpha, \mathcal{U}_\alpha)\}$, that is a collection of local charts $(\phi_\alpha, \mathcal{U}_\alpha)$ where $\mathcal{U}_\alpha \subseteq \mathcal{D}$.

Ideally, the model of deformation should provide a single chart $(\phi, \mathcal{D})$, which would induce a global parameterization ϕ^{-1} of $\mathcal{D}$, with $\phi^{-1} : \mathcal{U} \rightarrow \mathcal{D}$ and $\mathcal{U} \subset \mathbb{R}^m$. This parameterization would offer an elegant way to sample on the manifold and the Euclidean topology of the coordinates set gives a straightforward metric.

In this direction, the following section will introduce a global parameterization for Kirchhoff quasi-static elastic rods.

B. Kirchhoff Quasi-Static Elastic Rods

This section briefly presents the results of Bretl and McCarthy [5]. We encourage the interested reader to refer to the original paper for details. Consider a Kirchhoff elastic rod with fixed end positions. At equilibrium configurations, the rod locally minimizes its total elastic energy and its shape can be expressed as a local solution to a geometric optimal control problem. From Theorem 6 in [5], the set of solutions to this optimal control problem is a 6-manifold that can be parameterized by an atlas having a single chart, with coordinates given by the open subset $\mathcal{A} \subset \mathbb{R}^6$. Let $t \in [0,1]$ be the parameterization along the rod and $g : [0,1] \times \mathcal{A} \rightarrow SE(3)$ be the mapping that describes the rod spatial position at its parameter t for a given coordinate $a \in \mathcal{A}$ in previously described chart. Note these spatial positions are relative to the rod base $g(0,a)$. For a fixed, computing $g(t,a)$ for $t \in [0,1]$ requires the integration of several nonlinear differential systems (see Fig. 6 in [5]). More specifically, sufficient conditions for static equilibrium of the rod have to be checked. From Theorem 4 in [5], a rod configuration is in static equilibrium if and only if $\det(\mathbf{J}(t,a)) \neq 0$ for all $t \in (0,1]$ where the Jacobian matrix $\mathbf{J} \in \mathbb{R}^{6\times 6}$ is defined by:

$$\mathbf{J}(t,a) = T_{g(t,a)} L_{g(t,a)^{-1}} \left[\frac{\partial g(t,a)}{\partial a_1} \cdots \frac{\partial g(t,a)}{\partial a_6} \right] \quad (1)$$

At a given rod position t, this Jacobian matrix describes variations of the geometry $\partial g(t,a)$ with respect to variations ∂a in the chart expressed at the identity element of $SE(3)$. The set of coordinates a for which the rod is in static equilibrium will be noted $\mathcal{A}_{stable}$ with $\mathcal{A}_{stable} \subset \mathcal{A}$. These results enable us to use the coordinates $\mathcal{A}_{stable}$ to describe an equilibrium configuration of the rod.

C. Planning for Free-Flying Elastic Rods

In sampling-based methods for motion planning, a key predicate is determining if a configuration lies in $\mathcal{C}_{free}$, i.e. checking collision in $\mathcal{C}$. In most cases, this is performed by going back to the workspace $\mathcal{W}$ and carrying out two steps. First, we compute the DGM, which consists in a mapping from a robot configuration in $\mathcal{C}$ to its corresponding geometry in $\mathcal{W}$. Then, we check for collisions in $\mathcal{W}$ between robot geometry and obstacles. If the geometry is collision free, then q lies in $\mathcal{C}_{free}$.

Using the configuration space formulation, adding a 6 degree of freedoms free-flyer joint to the deformable robot changes the configuration space to $\mathcal{C} = \mathcal{A} \times SE(3)$ which is a 12-manifold. A configuration in this space will be noted $q^T = \begin{pmatrix} a & p \end{pmatrix}$ with $a \in \mathcal{A}$ and $p \in SE(3)$. Note that we consider here the free-flyer joint is attached to the base of the cable. Then the corresponding rod geometry is described by $w : [0,1] \times \mathcal{C} \rightarrow \mathcal{W}$ with $\mathcal{W} \subset SE(3)$. Let $\mathcal{C}_{col} \subseteq \mathcal{C}$ be the set of all configurations of the rod that lead to collisions with the environment in the workspace. Obviously, this set depends on both its deformation state a and its position p. Let now $\mathcal{C}_{self} = \{(a,p) \in \mathcal{C} \mid a \in \mathcal{A}_{self}\}$, with $\mathcal{A}_{self} \subseteq \mathcal{A}$, be the set of all deformation states which results in a self-colliding geometry. Then, the set of invalid configurations is given by:

$$\mathcal{C}_{inv} = \mathcal{C}_{col} \cup \mathcal{C}_{self} \cup \mathcal{C}_{unstable} \tag{2}$$

with $\mathcal{C}_{unstable} = \{(a,p) \in \mathcal{C} \mid a \notin \mathcal{A}_{stable}\}$. Finally, we have defined the set of valid configurations for a free-flying quasi-static elastic rod, i.e. $\mathcal{C}_{free} = \mathcal{C} \setminus \mathcal{C}_{inv}$.

D. Collision Checking With Fast DGM Approximation

It is now well known that the collision-checking step is a bottleneck for motion planning problems. This is especially true for virtual prototyping applications where the CAD models have generally more than 100'000 polygons. In common cases, computing the DGM is extremely fast (e.g. a kinematical chain). But in the deformable object context, computation of the DGM is considerably more costly. Using the rod model described in III-B, the DGM requires the numerical integration along the parameter t of several nonlinear differential systems. Even if these results are much more efficient than a numerical approach based on minimal energy optimization, this step becomes at least as costly as collision checking in the workspace for a sufficiently reasonable number of rod nodes. Note that the number of nodes is given by the integration resolution, and consequently the computational time to compute the DGM is in linear complexity with the number of rod nodes.

1) Fast Neighborhood Approximation: To ensure a rod configuration q is in static equilibrium, we have to compute the Jacobian $\mathbf{J}(a)$ of the mapping g. Without additional computational time, we can take advantage of this computation which describes the behavior of the rod geometry in the neighborhood of a in the deformation space $\mathcal{A}$.

As illustrated in Fig. 1, Jacobian matrix enables us to approximate rod geometry up to 10 times faster than the full DGM computation.

2) Local path validation: In addition to check if a configuration lies in $\mathcal{C}_{free}$, sampling-based methods also requires a more general predicate. The local path validation consists in checking if a local path (i.e. the shortest path between two configurations with respect to the metric associated to $\mathcal{C}$), lies in $\mathcal{C}_{free}$. This is usually done by sampling along this local path with a given resolution Δq and checking collision in $\mathcal{C}$ for each sample. In our case, the geodesic between two configurations $q_1^T = \begin{pmatrix} a_1 & p_1 \end{pmatrix}$ and $q_2^T = \begin{pmatrix} a_2 & p_2 \end{pmatrix}$ of

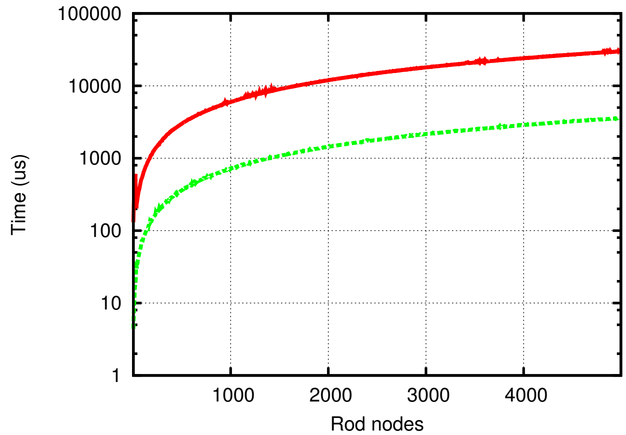

Fig. 1. Log scale computation time of the DGM (red) and its first order approximation (green).

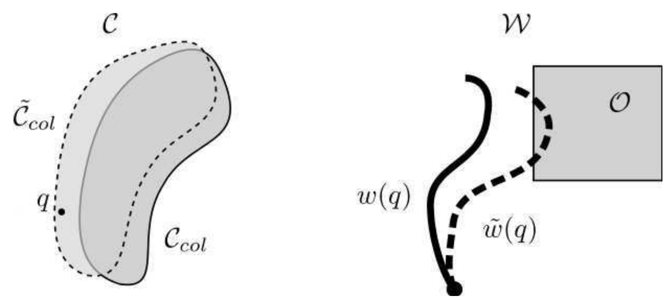

Fig. 2. Correspondance between approximation of the rod geometry $\tilde{w}(q)$ in the workspace $\mathcal{W}$ (right) and approximation of the environment colliding space $\tilde{\mathcal{C}}_{col}$ in the configuration space $\mathcal{C}$ (left). In this case, the approximation of the geometry induces a false invalid configuration detection.

$\mathcal{C} = \mathcal{A} \times SE(3)$ is given by $q(\lambda)^T = \begin{pmatrix} a(\lambda) & p(\lambda) \end{pmatrix}$ with

$$a(\lambda) = \lambda a_1 + (1-\lambda) a_2 \tag{3}$$

as $\mathcal{A}$ are coordinates on a chart and $p(\lambda)$ is the geodesic on $SE(3)$ as defined in [14].

The approximated geometry $\tilde{w}(q)$ given by (**??**) can be checked for self-collisions and collisions with the environment, i.e. determining if q lies in $\tilde{\mathcal{C}}_{col} \cup \tilde{\mathcal{C}}_{self}$, where $\tilde{\mathcal{C}}_{col}$ (resp. $\tilde{\mathcal{C}}_{self}$) are local approximations of $\mathcal{C}_{col}$ (resp. $\mathcal{C}_{self}$) due to the use of approximated geometry $\tilde{w}(q)$ (see Fig. 2). Note that the locally approximated invalid configurations set $\tilde{\mathcal{C}}_{col} \cup \tilde{\mathcal{C}}_{self}$ differs from $\mathcal{C}_{inv}$ defined in (2) by not taking into account unstable rod configurations $\mathcal{C}_{unstable}$ in addition to the error induced by the approximation. However, as it will be presented next, this difference is still much lower than $\mathcal{C}_{col} \cup \mathcal{C}_{self}$ and consequently, checking the approximated geometry $\tilde{w}(q)$ is a good guess of determining if q lies in $\mathcal{C}_{inv}$.

As sampled configuration along a local path are typically close, this approximation can be efficiently used to approximately validate a local path. There are different approaches to this end, and one is illustrated in Fig. 3. The basic idea states as follow: while sampling along the local path with a resolution Δq, we select the closest configuration where the DGM has been computed and we use its Jacobian to

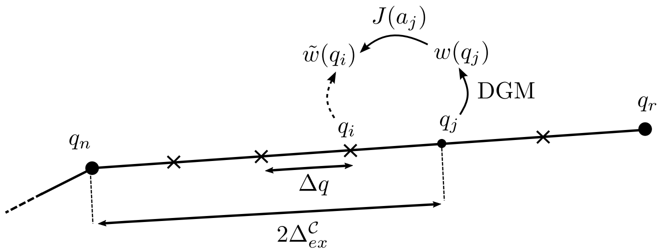

Fig. 3. Local path validation using DGM Jacobian approximation. The approximation of the geometry $\tilde{w}(q_i)$ is obtained using the geometry $w(q_j)$ of the closest configuration q_j having exact geometry. Crosses (resp. dots) represent configurations where the DGM will be approximated (resp. fully computed).

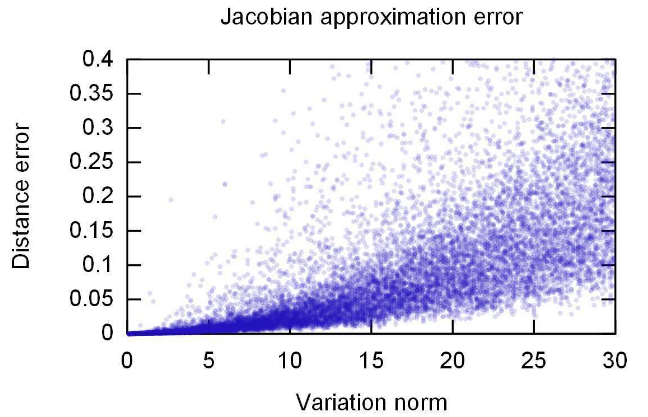

Fig. 4. Measurements of the distance error between an approximated and an exact geometry depending on the norm of the variation $\|\delta a\|$ in $\mathcal{A}$.

approximate the geometry of the current sample if the distance between the two configurations is less than a threshold Δ_{ex}^{C}. Otherwise, if there is no configuration with an evaluated DGM in the neighborhood bounded by Δ_{ex}^{C}, a new full computation of the DGM will be performed.

This validation scheme raises two sub-problems that can be handled efficiently. First, we must choose a good value for the sampling distance Δq. A too coarse resolution would lead to missed collisions and, in the opposite, considerable time would be wasted in unnecessary collision-checking for a too fine value. Although variations $\partial w(t,q)$ with respect to ∂p can be computed globally, variations $\partial w(t,q)$ with respect to ∂a require more attention as the mapping $g(t,a)$ is only a local diffeomorphism (see proof of Theorem 8 in [5]). To handle this problem, we chose to use Δq as an initial guess of the sampling resolution and we ensure the maximum node to node distance as defined in (4) is less than a given maximum penetration in the workspace.

The second problem is to characterize the neighborhood where we assume the validity of our approximation. This is detailed in the following paragraph.

3) Approximation error: Our work has been highly motivated by the low distance error between an approximated rod geometry using the Jacobian and the exact geometry given by the DGM (see Fig. 4). The distance function considered here is the maximum node to node distance given by:

$$\rho(w_1, w_2) = \max_{t\in[0,1]} d_T(w_1(t), w_2(t)) \tag{4}$$

where $d_T : SE(3) \times SE(3) \to \mathbb{R}$ is the Euclidean distance between the translation part of the nodes. Thanks to this results, we have a good estimation about how far from a configuration our approximation is assumed to be valid. For a given distance error, the corresponding norm of the variation in the deformation space $\mathcal{A}$ denoted Δ_{ex}^{A} is then used to compute its corresponding maximum variation Δ_{ex}^{C} in the full configuration space $\mathcal{C}$ for a given local path. This enables us to take advantage of the decoupling between the two subspaces $\mathcal{A}$ and $SE(3)$. This distance Δ_{ex}^{C} allows us to control the size of the neighborhood we want to approximate. As presented in the following algorithm, choosing a bad value for Δ_{ex}^{C} would not be critical for the validity of the result path and would only penalize performances.

4) Fast Forward Geometry RRT: FFG-RRT: In this section, we present an extension of the classical RRT algorithm which encapsulates our fast neighborhood approximation. We will call it Fast Forward Geometry RRT (FFG-RRT). The global structure of the FFG-RRT algorithm detailed in Alg. 1 is very similar to a Lazy-RRT. In our context of motion planning for quasi-static Kirchhoff elastic rods, the APPROX_EXTEND function could be the local path validation as described in III-D.2.

The second part of the algorithm is given by the CHECK_SOLUTION function. This function tries to find a solution path in the current tree where most of the edges are only approximately valid and check for exact validity of this path. Invalidated edges are then removed from the tree.

Algorithm 1 FFG-RRT(q_{start}, q_{goal})

$\mathcal{T}$.init(q_{start})
for $i \leftarrow 1$ to k **do**
  $q_{rand} \leftarrow$ RANDOM_SAMPLE()
  $q_{near} \leftarrow$ NEAREST($\mathcal{T}, q_{rand}$)
  $q_{new} \leftarrow$ APPROX_EXTEND(q_{near}, q_{rand})
  if $q_{new} \neq q_{near}$ **then**
    $\mathcal{T}$.addVertex(q_{new})
    $\mathcal{T}$.addEdge(q_{near}, q_{new})
  end if
  ($solved$, τ) $\leftarrow$ CHECK_SOLUTION($q_{start}, q_{goal}, \mathcal{T}$)
  if *solved* **then return** τ
  end if
end for
return failure

Note the efficiency of this algorithm relies on the number of tree reconstructions in the function CHECK_SOLUTION. As we seen that our approximation is close to the exact DGM, this implies a low number of calls to this function and this will be verified in the experimental results.

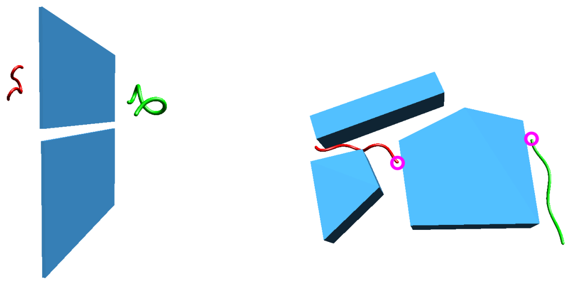

Fig. 5. The two toy scenarios used for benchmarks: crack (left) and backward (right). Start (resp. goal) configurations of the rod are represented in green (resp. red). Note the head of cable has been marked with a magenta circle on the Benchmark scenario to emphasize solution path could not be obtained with a point-like robot.

IV. EXPERIMENTAL RESULTS

In this section we will present and analyze results on solving the motion planning problem for free flying Kirchhoff elastic rods using the model presented in section III-B and our approach for fast approximation of the Direct Geometric Model detailed in section III-D.

A. Implementation Detail

We chose to implement our approach on top of the classical RRT and RRT-Connect [7] planners for the benchmarks using the C++ motion planning library OMPL [17]. The numerical integration of differential systems required to obtain the geometrical state of the rod was implemented using fourth order Runge-Kutta methods. For collision checking, the geometry of the rod was approximated to a hierarchical chain of capsules (i.e. Line Swept Spheres). The Flexible Collision Library (FCL) [13] was used to perform collision checking computations. All the benchmarks were run on a PC with 8GB of main memory and using one core of an Intel Core i7-2720QM processor running at 2.2Ghz.

B. Benchmark Scenarios

We selected four distinguished scenarios in order to test the effectiveness of our approach. Two firsts consist in toy scenarios (see Fig. 5) where each shows a specific difficulty. Last two scenarios are industrial cases with a disassembly study (see Fig. 6). On all of this cases, bounds have been set on the configuration space but, for clarity, environment bounding boxes are not shown in the illustrations.

- **Crack.** Lightweight model where the rod must pass through a crack shaped narrow passage. The length (resp. diameter) of the rod is half the length (resp. width) of the crack free space.
- **Backward.** Lightweight model where a point-like robot path would give an invalid guess or approximate solution. Note that due to the bounding box, the DLO has to go trough the corridor between obstacles.
- **Free-Flying engine.** Industrial model with 132,000 polygons and where the rod models a cable in a typical disassembly study case.

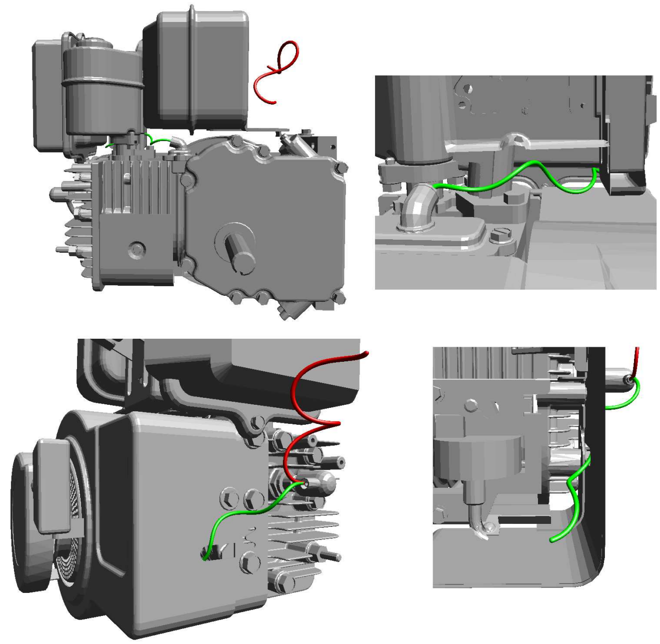

Fig. 6. Two industrial cases of disassembling for a free-flying cable (top) and a fixed base cable (bottom). The cable has to get out from a highly constrained start configuration (in green) to a low constrained configuration (in red). Closer views on the start configuration are represented on the right for each scenario.

- **Fixed engine.** The model is the same as in the previous scenario but here the cable has a fixed base.

Note that on all of these scenarios, the rod has to deform its shape to go from start to goal configurations. We ran the benchmark 30 times for each case. The number of rod nodes has been chosen to give sufficient realism of the physical deformation and depends on the level of deformation and the rod length.

C. Results

TABLE I
PLANNING PERFORMANCE COMPARISON

Scenario	RRT	FFG-RRT	RRT Connect	FFG-RRT Connect	
Crack	Failed 0	Failed 0	301,2 80	113,5 96,7	Time Success
Backward	Failed 0	Failed 0	450,6 93,3	172,9 86,7	Time Success
Free-flying engine	Failed 0	Failed 0	240,7 100	82,9 100	Time Success
Fixed engine	95,1 50	19,6 40	636,8 100	250,9 100	Time Success

Benchmark results are shown in Table I. Times are average planning time in seconds and success rate is in percentage. We see our approach solves about two to three times faster than its respective classical implementation, with a similar success rate. Also, as toy scenarios typically show two connected components with a narrow passage, bi-directional planners such RRT-Connect are highly more efficient in these cases. Thanks to the quality of the approximation, the function CHECK_SOLUTION is rarely called (0 to 2 calls) which ensures us a significant performance improvement. Theses cases where the approximated solution path is invalidated are mainly due to the non respect of sufficient conditions for static equilibrium as described in III-B (i.e. $q \in \mathcal{C}_{unstable}$).

V. CONCLUSION AND PERSPECTIVES

In this paper, we presented an new approach to the motion planning problem for a Kirchhoff elastic rod. Using a realistic physical model for the DLO, we are able to solve complex industrial scenarios in a reasonable time. We also introduced a new local planning scheme by taking advantage of the linearization of the Direct Geometrical Model which enables to solve the problem two to three times faster. This local method can be embedded in many different motion planning algorithms and extends to any model where the DGM cost becomes critical.

In the future, we would like to extend the approach by coupling with dynamic simulation to ensure mechanical constraints of the rod between two quasi-static rod configurations. Another interesting problem in manufacturing would be to take into account gripers of the rod during the planning process.

ACKNOWLEDGMENT

This work was supported by the French National Research Agency under the project Flecto (ANR- Digital Models). Industrial models are courtesy of Siemens-KineoCAM.

REFERENCES

[1] Stuart S Antman. Kirchhoff's problem for nonlinearly elastic rods(for axial extension and cross section shear). *Quarterly of applied mathematics*, 32:221–240, 1974.

[2] O. Burçhan Bayazit, Jyh-Ming Lien, and Nancy M. Amato. Probabilistic roadmap motion planning for deformable objects. In *ICRA*, pages 2126–2133. IEEE, 2002.

[3] Miklós Bergou, Max Wardetzky, Stephen Robinson, Basile Audoly, and Eitan Grinspun. Discrete Elastic Rods. *ACM Transactions on Graphics (SIGGRAPH)*, 27(3):63:1–63:12, aug 2008.

[4] Timothy Bretl and Zoe McCarthy. Equilibrium configurations of a kirchhoff elastic rod under quasi-static manipulation. In Emilio Frazzoli, Tomás Lozano-Pérez, Nicholas Roy, and Daniela Rus, editors, *WAFR*, volume 86 of *Springer Tracts in Advanced Robotics*, pages 71–87. Springer, 2012.

[5] Timothy Bretl and Zoe McCarthy. Quasi-static manipulation of a kirchhoff elastic rod based on a geometric analysis of equilibrium configurations. *I. J. Robotic Res.*, 33(1):48–68, 2014.

[6] Russell Gayle, Paul Segars, Ming C. Lin, and Dinesh Manocha. Path planning for deformable robots in complex environments. In Sebastian Thrun, Gaurav S. Sukhatme, and Stefan Schaal, editors, *Robotics: Science and Systems*, pages 225–232. The MIT Press, 2005.

[7] James J. Kuffner Jr. and Steven M. Lavalle. Rrt-connect: An efficient approach to single-query path planning. In *Proc. IEEE Intl Conf. on Robotics and Automation*, pages 995–1001, 2000.

[8] Ilknur Kabul, Russell Gayle, and Ming C. Lin. Cable route planning in complex environments using constrained sampling. In Bruno Lévy and Dinesh Manocha, editors, *Symposium on Solid and Physical Modeling*, pages 395–402. ACM, 2007.

[9] Florent Lamiraux and Lydia E. Kavraki. Planning paths for elastic objects under manipulation constraints. *I. J. Robotic Res.*, 20(3):188–208, 2001.

[10] Joel Langer and David A. Singer. Lagrangian aspects of the kirchhoff elastic rod. *SIAM Rev.*, 38(4):605–618, December 1996.

[11] Arthur Mahoney, Joshua Bross, and David Johnson. Deformable robot motion planning in a reduced-dimension configuration space. In *ICRA*, pages 5133–5138. IEEE, 2010.

[12] Mark Moll and Lydia E. Kavraki. Path planning for deformable linear objects. *IEEE Transactions on Robotics*, 22:625–636, 2006.

[13] Jia Pan, Sachin Chitta, and Dinesh Manocha. Fcl: A general purpose library for collision and proximity queries. In *ICRA*, pages 3859–3866. IEEE, 2012.

[14] F. C. Park. Distance metrics on the rigid-body motions with applications to mechanism design. *ASME Trans., Journal of Mechanical Design*, 117(1):48–54, 1995.

[15] Samuel Rodríguez, Jyh-Ming Lien, and Nancy M. Amato. Planning motion in completely deformable environments. In *ICRA*, pages 2466–2471. IEEE, 2006.

[16] JC Simo. A finite strain beam formulation. the three-dimensional dynamic problem. part i. *Computer methods in applied mechanics and engineering*, 49(1):55–70, 1985.

[17] Ioan A. Şucan, Mark Moll, and Lydia E. Kavraki. The Open Motion Planning Library. *IEEE Robotics & Automation Magazine*, 19(4):72–82, December 2012. http://ompl.kavrakilab.org.

[18] Hidefumi Wakamatsu and Shinichi Hirai. Static modeling of linear object deformation based on differential geometry. *I. J. Robotic Res.*, 23(3):293–311, 2004.