Robotica: (2020) volume 38, pp. 819–830. © Cambridge University Press 2019
doi:10.1017/S0263574719001103

Motion Planning for Deformable Linear Objects Under Multiple Constraints

Jiangtao Ma, Jianhua Liu, Xiaoyu Ding* and Naijing Lv

School of Mechanical Engineering, Beijing Institute of Technology, Beijing 100081, China

(Accepted June 12, 2019. First published online: July 12, 2019)

SUMMARY

Deformable linear objects (DLOs) have a wide variety of applications in a range of fields. Their key characteristic is that they extend much further in one of their dimensions than in the other two. Accurate motion planning is particularly important in the case of DLOs used in robotics applications. In this paper, a new strategy for planning the motions of DLOs under multiple constraints is proposed. The DLO was modeled as Cosserat elastic rods so that the deformation is simulated accurately and efficiently. The control of the motion of the DLO was enhanced by supplementing one gripper installed at each end with additional supports. This allows DLOs to undergo complex deformations, and thus avoid collisions during motion. The appropriate number of supports and their positions were determined, and then a rapidly exploring random tree algorithm was used to search for the best path to guide the DLO toward its target destination. The motion of the simulated DLO is described as it is controlled using two grippers and specific numbers of supports. To prove that the proposed DLO motion planning strategy can successfully guide relatively long DLOs through complex environments without colliding with obstacles, a case study of the strategy was conducted when guiding a DLO through a puzzle.

KEYWORDS: Deformable objects; Elastic rod; Motion planning; Path planning; Virtual fixture.

1. Introduction

Deformable linear objects (DLOs), which are characterized by extending much further in one of their dimensions than in the other two, have a wide variety of applications in a range of fields. Examples of DLOs include electric cables, wiring harnesses, snake-like robots, surgical sutures, etc. DLO motion planning is important in industrial assembly and disassembly[1,2] and plays an essential role in robotics manipulation.[3–7] The key aim of DLO motion planning is to find a smooth and collision-free path that takes the DLO from its initial configuration to the goal configuration. DLO motion planning has been widely acknowledged as a challenging research area because it requires the deformation of DLOs to be simulated efficiently and realistically, which means it must be able to quickly predict upcoming collisions with itself and the environment.

Recently, it has become popular to base models of DLO motion on continuous mechanics theory, including Kirchhoff elastic rod theory[8,9] and Cosserat elastic rod theory.[10,11] Hermansson et al.[12] presented a path-planning method for assembling multibranch wire harnesses with the aid of a model based on Cosserat elastic rod theory. Their method can be applied to any tree-like DLO. Bretl and McCarthy[13] modeled DLOs as Kirchhoff elastic rods and specified manipulation constraints that are analogous to holding the DLO by a gripper at each end. They determined the functional relationship between the configuration of the DLO and the position and orientation of each gripper, which can be expressed as boundary conditions, and used it as the basis of their implementation of

* Corresponding author. E-mail: xiaoyu.ding@bit.edu.cn

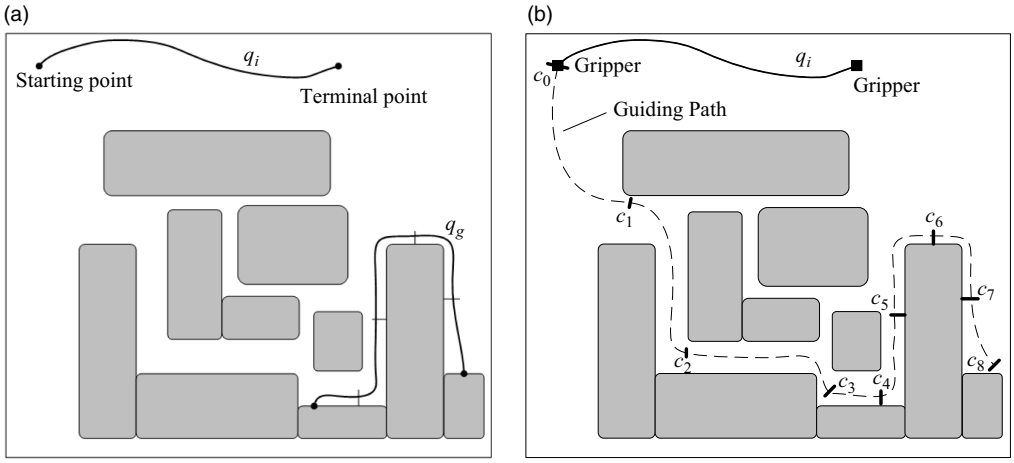

Fig. 1. The main concepts used in DLO motion planning. (a) Initial and goal configuration of the DLO. (b) Guiding path, grippers, and supports.

a sampling-based algorithm for DLO motion planning. Roussel et al.[14] presented a motion-planning method for DLOs based on the work in Bretl and McCarthy,[13] which detailed a fast method for checking collisions by approximating the geometry neighborhoods of the quasi-static elastic rods. Roussel et al.[15] extended their collision-free DLO motion planning strategy to handle objects that come into contact with DLOs. In these studies, only two grippers, one installed at each end, were used to manipulate the DLOs. If long DLOs are only controlled by a gripper at each end, they cannot be manipulated to generate the complex deformations that may be required to avoid collisions with obstacles. In this situation, complex deformation can be achieved by installing additional constraints. However, to the best of our knowledge, this issue has not been addressed in the literature.

In this study, the concept of supports is defined to supply the additional constraints. Like grippers, the supports are abstractions of robot arms' behaviors. To control the motion of DLOs, supports supply two behaviors: the one is to restrict the tangent and position of DLOs, the other is to restrict the movement of the DLO by a specific length. Based on the concept of the support, a new strategy for planning the motions of DLOs under multiple constraints is presented. The deformation simulations are accurate and efficient because the model represents DLOs as Cosserat elastic rods. The motion-planning strategy was developed by extending the guiding path concept presented in ref. [16], and then by designing a method to determine the appropriate supports to apply to the motion of the DLO.

This paper is organized as follows: In Section 2, some basic concepts related to this topic are introduced. An overview of the framework of the proposed motion planning strategy is provided in Section 3, and the methods are described in detail in Sections 4–6. The proposed strategy is validated by conducting a case study, and the results of which are summarized in Section 7. The conclusions are presented in Section 8.

2. Conceptual Basis of the Proposed Strategy

In this section, the basic concepts underpinning the proposed motion planning strategy are introduced.

Configuration: As mentioned above, the shape and position of a DLO constitute its configuration, which is generally denoted as q_t. The key aim when planning the motion of a DLO is to transfer it from its initial configuration to its goal configuration by continually changing its configuration, without colliding it with itself or other objects. As shown in Fig. 1(a), q_i and q_g denote the initial and goal configurations of a DLO, respectively. The configuration of a DLO has a start, or front, point and a terminal, or back, point, as indicated in Fig. 1(a). The goal configuration of the DLO is shown in Fig. 2.

Guiding path: The guiding path is the track along which the DLO proceeds, as indicated by the dashed curve in Fig. 1(b). It should be noted that the guiding path is only a preliminary route that restricts the motions available to the DLO, so it does not completely coincide with the actual configurations of the DLO during motion.

https://doi.org/10.1017/S0263574719001103 Published online by Cambridge University Press

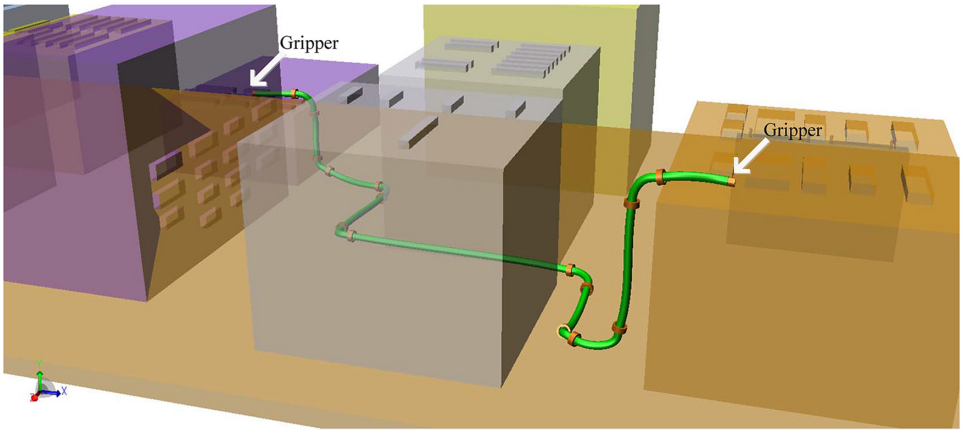

Fig. 2. The DLO and its grippers.

Gripper: One gripper installed at each end of the DLO is used to control the position and orientation of the DLO at both ends, as indicated by the square marks in Fig. 1(b). As such, they act as manipulation constraints. The grippers let go of the DLO when it enters a support and re-grasp the DLO as it emerges on the other side. The gripper in the CAD model is represented by a cylinder, shown in Fig. 2.

Support: The DLO can be prevented from colliding with obstacles by using another kind of constraint, which is imposed by adding supports. The supports are placed along the guiding path and are represented by c_i, such as $c_0, c_1, \ldots, c_8$ in Fig. 1(b). Supports supply the constraints to the motion of the DLO, that is, their position and orientation, thus allowing us to control its configuration. Section 5 describes the method used to determine how many supports should be installed, and where they should be located. Also, the performance of these supports is described in Section 6.

Cosserat elastic rod: In this study, the model used to calculate the configuration of the DLO combines Cosserat elastic rods with the boundary conditions imposed by the two types of constraints mentioned above. The model of Cosserat elastic rods is the same as that used in ref. [11]. The effects of the stretch, bending, torsion, and gravity of the DLO are described by this model. It is assumed that the DLO is in static equilibrium at every moment. Modeling the DLO as a Cosserat elastic rod enables us to determine its equilibrium configuration by solving its minimal potential energy.[10,17–19] The potential energy of a DLO with length L is:

$$\begin{cases} E = \int_0^L V_s ds + \int_0^L V_b ds + \int_0^L V_f ds \\ \varphi(0) = (x(0), v(0)) \\ \varphi(L) = (x(L), v(L)) \\ \varphi(c_i) = (x(c_i), v(c_i)) \end{cases} \tag{1}$$

where V_s is the stored stretch energy density and $\int_0^L V_s ds$ is the stretch energy. V_b is the density of the sum of bending energy and torsional energy and $\int_0^L V_b ds$ is the sum of the bending energy and torsional energy. V_f is the density of potential energy produced by the external force, such as gravity, and $\int_0^L V_f ds$ is the sum of potential energy produced by the external force. The other terms in Eq. (1) represent the boundary conditions. Specifically, $\varphi(0) = (x(0), v(0))$ and $\varphi(L) = (x(L), v(L))$ represent the position and tangent of the gripper at each end, and $\varphi(c_i) = (x(c_i), v(c_i))$ represents the position and tangent of the supports. The Cosserat elastic rod is solved using the numerical method described in refs. [17–19], that is, the synthesis of Newton's method, Gauss algorithm, and the conjugate gradient method,[17] the linear time dynamics algorithm,[18] and the finite element method.[19] The configuration of the DLO calculated every time by Eq. (1) is used for the next calculation to ensure that the boundary conditions correspond to a unique configuration of the DLO.

3. Motion Planning Strategy Framework

Figure 3 presents the framework of the proposed motion planning strategy. The initial guiding path is generated after the initial and goal configurations and the workspace has been loaded. Next, the

https://doi.org/10.1017/S0263574719001103 Published online by Cambridge University Press

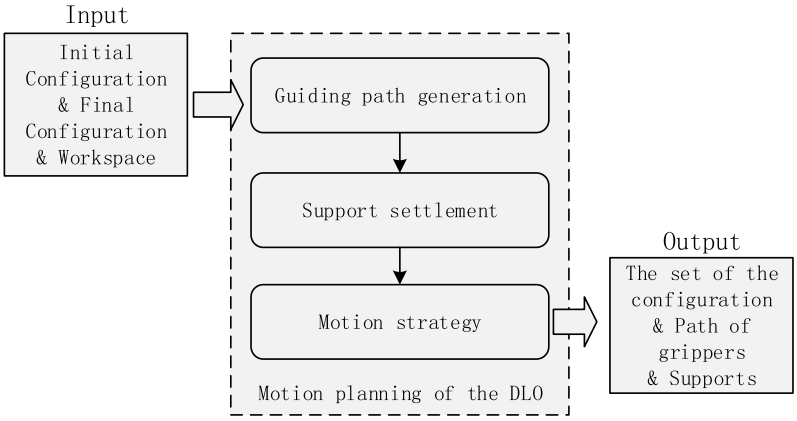

Fig. 3. The framework of the motion planning strategy.

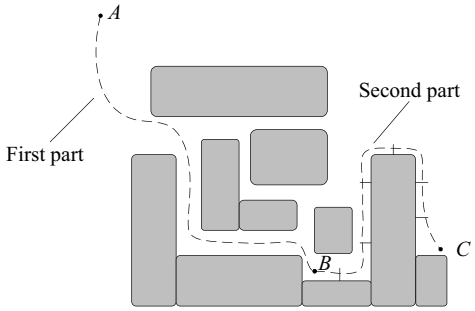

Fig. 4. Schematic diagram of the path that guided the motion of the DLO.

appropriate quantity of supports and their positions along the guiding path are calculated. Finally, changes in the configuration of the DLO cause it to move along the guiding path, from its initial configuration to its goal configuration. It avoids collisions with the aid of the supports. When the motion planning is finished, the output is a sequential set of the DLO configurations, the paths of the two grippers, and the locations of the supports placed along the guiding path.

4. Guiding Path Generation

The guiding path consists of two parts: the path from A to B and the path from B to C, as indicated in Fig. 4. A medial-axis-biased rapidly exploring random tree (MARRT) is used to generate the first part,[20] and the second part represents the final configuration of the DLO. The algorithm to find the first part of the guiding path evaluates the following three variables: (1) the medial axis of the workspace; (2) a MARRT that connects the initial and final configurations; and (3) the shortest path between the initial and final configurations on the MARRT. Rather than being a smooth curve, the first part of the guiding path is actually composed of several line segments. Figure 4 presents a schematic representation of a guiding path.

The medial axis of an object, which was first proposed by Blum,[21] is the set of all points that are equidistant to two or more obstacle boundaries. There are three categories of methods for finding the medial axis: thinning, tracing, and Voronoi-graph-based.[22] In this study, a Voronoi-graph-based method is used, and the method computes a subcomplex from the Voronoi diagram that lies close to the medial axis and converges to it as sampling density approaches infinity.[23] However, one of the other methods described in refs. [24, 25] could have been used. The details of the medial axis generation step are beyond the scope of the current study, but readers can refer to refs. [22–25] for further information. Figure 5(a) shows the medial axis of the workspace shown in Fig. 4.

The rapidly exploring random tree (RRT) is a category of algorithms designed to search spaces with obstacles efficiently by randomly constructing a space-filling tree. The tree is constructed incrementally from sample points drawn randomly from the search space and is inherently biased to grow

https://doi.org/10.1017/S0263574719001103 Published online by Cambridge University Press

Fig. 5. The process used to generate the guiding path. (a) An example medial axis. (b) The tree generated by the MARRT algorithm. (c) The guiding path between *A* and *B*. (d) The final guiding path.

toward the large unsearched areas of this space.[26] RRTs have been studied for many years, leading to the development of a wide range of RRT-based algorithms, such as RRT-Smart,[27] A*-RRT,[28] RRT-Fixed Number,[29] and MARRT,[20] etc.

In the MARRT algorithm, which is applied in this study, the random tree generation is biased toward the medial axis. The MARRT algorithm takes the start and goal nodes as input, which are, respectively, *A* and *B* in Fig. 4. MARRT is presented in Algorithm 1. In each iteration, it randomly samples a node n_{rand} in unsearched areas and finds the nearest node n_{near} in the tree. New nodes are expanded by Algorithm 2. These nodes consist of a polygonal chain *I*. Algorithm 1 ends when the tree grows at the node *B*.

Algorithm 1: Medial Axis RRT

Input: The iteration number *num*, a start node *A*, an end node *B*.
Output: A Tree *T*
1. *T*.AddNode(*A*)
2. **for** 1 . . . *num* **do**
3. nrand ← RandomSampleNode()
4. nnear ← NearestNeighbor(*T*,n_{rand})
5. *I* ← MARRTExpand(n_{rand},n_{near})
6. *T*.Update(*I*,n_{near})
7. **return** *T*

The key of the MARRT is the expansion step. Algorithm 2 expands along the medial axis until the length of the expansion polygonal chain *I* reaches a maximum length l_{max}. During each iteration,

https://doi.org/10.1017/S0263574719001103 Published online by Cambridge University Press

a node n_{prev} on the medial axis is added to I, a new node n_{new} is extended from n_{prev} to n_{rand} at a distance $\triangle L$, and the n_{new} is pushed to the medial axis. The n_{new} is $\triangle L$ from n_{prev}. The algorithm ends until the length of I reaches the maximum length l_{max}.

Algorithm 2: MARRT Expansion

Input: Node n_{rand} and n_{near}, a step-distance $\triangle L$, a maximum expansion length l_{max}.
Output: Intermediate chain I

1. $I \leftarrow \Phi$
2. Node $n_{prev} \leftarrow n_{near}$, n_{new}
3. **repeat**
4. $n_{prev} \leftarrow n_{new}$
5. I.AddNode (n_{prev})
6. $n_{new} \leftarrow$ Extend(n_{prev}, n_{rand}, $\triangle L$)
7. PushToMedialAxis(n_{new}, $\triangle L$)
8. $dist \leftarrow I$.length()
9. **until** $dist \geq l_{max}$
10. **return** I

More details of the MARRT algorithm can be referred to ref. [20].

Figure 5(b) shows the tree obtained for the workspace shown in Fig. 4. It is easy to search for the shortest path between A and B on the tree in Fig. 5(b), as shown in Fig. 5(c). The path is composed of several nodes, with a short enough separation between adjacent nodes, distance $\triangle L$. Although it is not necessary, the coding work is much easier if the length of the DLO is an integer multiple of $\triangle L$. Finally, the second part of the guiding path is discretized by the step size $\triangle L$. The nodes on the guiding path can be numbered as n_k, so in this example, the node at A is numbered as n_0, the node at B is numbered as n_{32}, and the node at C is numbered as n_{54}.

5. Support Settlement

As shown in Fig. 5, the guiding path goes through an environment that contains many obstacles. Thus, the configuration of the DLO must be able to follow a relatively complex sequence of variations so that it can avoid collision as it proceeds along the guiding path. As discussed in the introduction, the DLO cannot avoid collisions if it is only held by two grippers. Hence, the supports are placed along the guiding path. These act as additional constraints that enable the control of the deformation of the DLO. As stated in Section 4, the second part of the guiding path corresponds to the goal configuration of the DLO. The grippers and supports for this part are provided as inputs to the motion planning process, as indicated in Fig. 3. Therefore, the supports need to be determined on the first part of the guiding path, for which the algorithm described below is used.

The search algorithm is presented by Algorithm 3. It starts at the goal node, that is, B, and numbers the supports to be determined, d_j. As shown in Fig. 6(a), the first support, d_0, is located at B and the second support, d_1, is initially located at the node proceeding B, that is, n_{31}. An assumed DLO is then placed between d_0 and d_1, as shown in Fig. 6(b). The length of the assumed DLO is equal to the length of the line segment between d_0 and d_1, and its tangent at d_0 is defined as the vector pointing from node n_{32} to n_{33}. The tangent of the assumed DLO at d_1 is defined as the vector pointing from node n_{31} to node n_{30}, as shown in Fig. 6(b). The Cosserat model, which is defined in Eq. (1), can be used to determine the configuration of the assumed DLO under these boundary conditions. The intersection is then verified between the assumed DLO and the obstacles using the collision detection algorithm described in ref. [30]. If the DLO is found to be on course for a collision, the search process will stop, leading the motion planning process to fail. However, this is unlikely to happen in practice. As shown in Fig. 6(b), the assumed DLO does not collide with obstacles, so d_1 moves to the previous node n_{30}, as shown in Fig. 6(c). At this point, the length of the assumed DLO is the sum of the lengths of the guiding path segments between d_0 and d_1. Then, as shown in Fig. 6(c), the tangent of the assumed DLO at d_1 becomes the vector pointing from node n_{30} to node n_{29}. Then, the configuration of the

https://doi.org/10.1017/S0263574719001103 Published online by Cambridge University Press

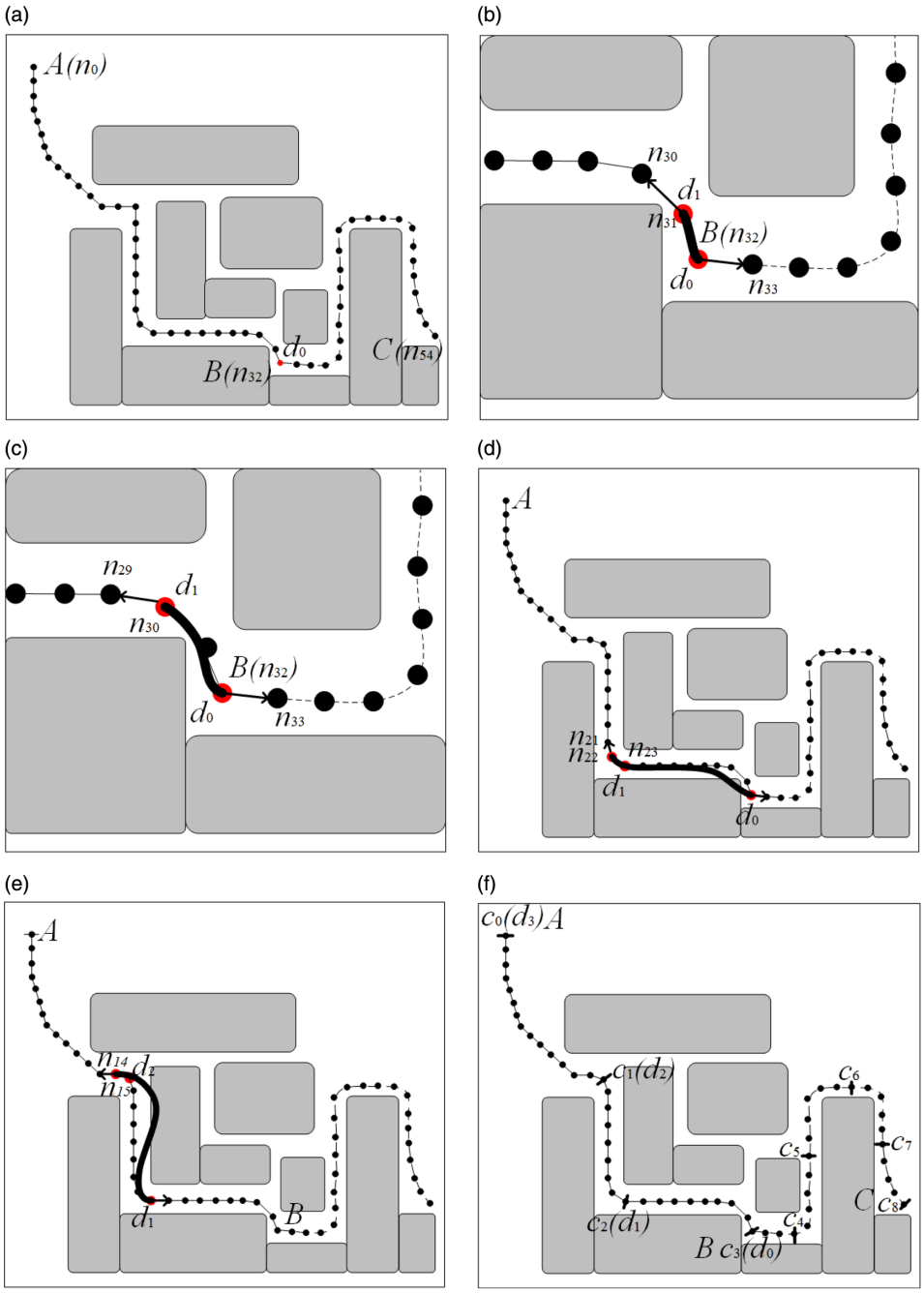

Fig. 6. The guiding path segmentation process. (a) Initially, support d_0 is located at point B (i.e., node n_{32}). (b) No collision is detected when support $d_1$is tried to be located at node n_{31}. (c) Again, no collision is detected when support $d_1$is tried to be located at n_{30}. (d) A collision is detected when support d_1 is tried to be located at node n_{22}. (e) Set the position of support d_2. (f) Renumber each support.

DLO is recalculated and checked that it does not intersect with any obstacles. If no collision will occur, the position of d_1 proceeds to the previous node n_{29} and the above calculation procedure is repeated, only stopping if the assumed DLO collides with any obstacles. In this example, the DLO collides with an obstacle when d_1 reaches node n_{22}, as shown in Fig. 6(d). In this case, d_1 moves back to what is now its final position, at n_{23}. Hence, according to this algorithm, support d_1 should be placed at node n_{23} and the direction of the support d_1 is the same as the tangent of the assumed DLO end at node n_{23}.

https://doi.org/10.1017/S0263574719001103 Published online by Cambridge University Press

After determining the location of d_1, which is also the initial position for the search for d_2, the algorithm begins its next search by repeating the same process used for d_1. This process finishes where the assumed DLO collides with obstacles, which in this case occurs when d_2 is located at node n_{14}, as shown in Fig. 6(e). Hence, the final location of d_2 is at node n_{15} and the direction of the support d_2 is the same as the tangent of the assumed DLO end at node n_{15}. The search process will stop when a support reaches the initial node A, which is the location of the final support. After the appropriate location of every support has been determined along the path, all of the supports are renumbered on the first and second part of the guiding path to c_k, as indicated in Fig. 6(f).

Algorithm 3: Support settlement

Input: Node A and B, and the guiding path.
Output: The supports $\{c_k\}$
1. d_0,$d_{\text{tmp1}} \leftarrow B$
2. nindex $\leftarrow B$.Index()
3. $i \leftarrow 1$
4. **While** nindex > 0
5. nindex $\leftarrow$ nindex - 1
6. $d_{\text{tmp2}} \leftarrow n_{\text{nindex}}$
7. *Assumed_DLO* $\leftarrow$ ConstructADLO(d_{tmp1}, d_{tmp2})
8. **While** *Assumed_DLO* is not collided with obstacles
9. nindex $\leftarrow$ nindex – 1
10. $d_{\text{tmp2}} \leftarrow n_{\text{nindex}}$
11. *Assumed_DLO* $\leftarrow$ ConstructADLO(d_{tmp1}, d_{tmp2})
12. **End While**
13. nindex $\leftarrow$ nindex + 1
14. d_i,$d_{\text{tmp1}} \leftarrow d_{\text{tmp2}}$
15. $i \leftarrow i$+1
16. **End While**
17. $\{c_k\} \leftarrow$ UpdateAllSupport($\{d_k\}$)
18. **Return** $\{c_k\}$

6. Motion of the DLO

By installing the appropriate configuration of supports, the motion of the DLO is constrained such that it can traverse the guiding path and reach the target configuration without colliding with any obstacles. This section describes the motion of a DLO that is controlled by two grippers with the aid of a specific number of supports.

As shown in Fig. 7(a), the connection between the initial configuration and the guiding path is not smooth. That is because the tangents of the initial configuration and the searched guiding path at A are unlikely to be equal. Therefore, it is necessary to first ensure that the tangent of the initial configuration of the DLO is the same as the tangent of the guiding path at A by adjusting the grippers. The results of this step are shown in Fig. 7(b). The gripper at the front of the DLO then moves from c_0 to the next node, while the gripper at the back of the DLO (i.e. the terminal point) moves in the same distance, stopping when the gripper at the front arrives at C. It should be noted that the gripper at the back of the DLO first moves in the initial configuration and then moves along the guiding path.

As the motion proceeds along the path, the supports control the positions and tangents of the DLO at supports' positions, and each support restricts the movement of the DLO by the step size $\triangle L$ in each iteration step, as mentioned in Section 4. Meanwhile, the orientations of the grippers at the front and back of the DLO are always the vectors pointing from node n_l to node n_{l+1} and from node n_m to node n_{m-1}, respectively. This process is shown in Fig. 7(c) and (d). The Cosserat elastic rod model can be used to calculate the changes in the configuration of the DLO during motion. The steps above are presented by Algorithm 4.

https://doi.org/10.1017/S0263574719001103 Published online by Cambridge University Press

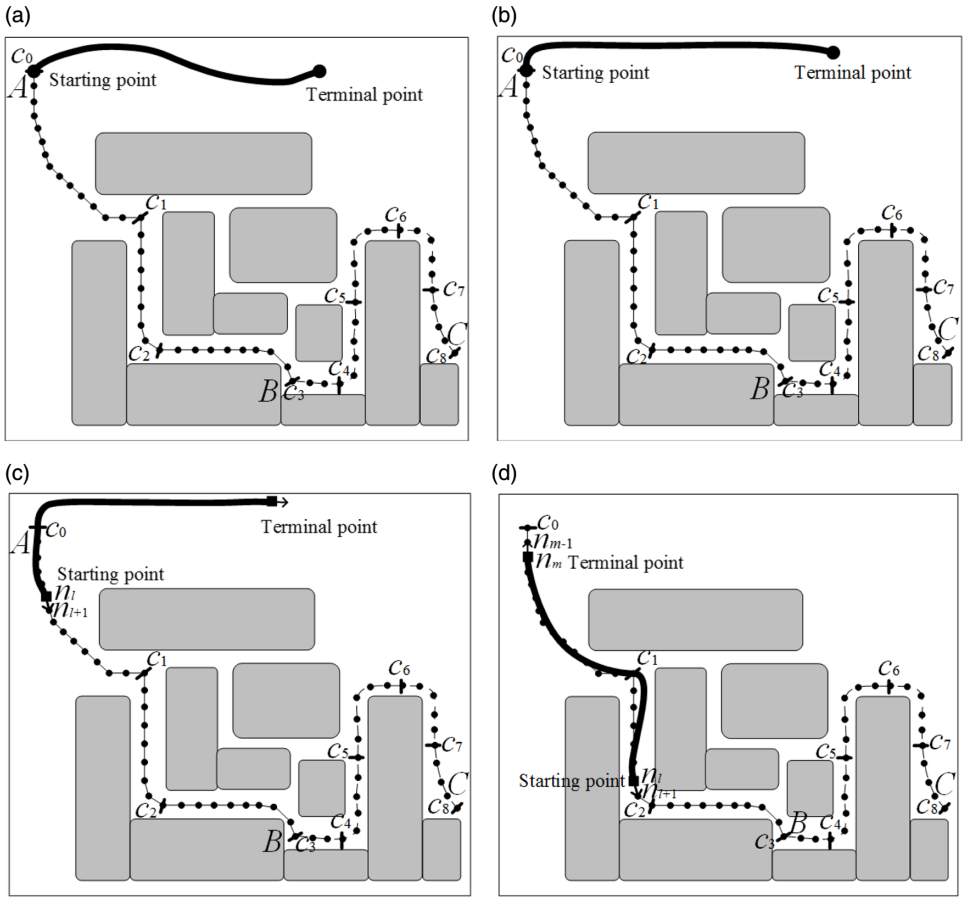

Fig. 7. The motion of the DLO. (a) The initial configuration of the DLO. (b) Adjusting the initial configuration of the DLO. (c) A snapshot of the DLO moving along the path when the terminal point is outside the guiding path. (d) A snapshot of the DLO moving along the path when the terminal point is on the guiding path.

Algorithm 4: Motion of the DLO

1. $q_{\text{adj}} \leftarrow$ AdjustConfigurationDLO(q_i)
2. The number of the guiding path nindex
3. **for** 1 . . . nindex **do**
4. if $i < B$.Index()
5. $\{c_j\} \leftarrow$ SearchSupport()
6. CalculateConfigurationDLO(q_{adj},$\{c_j\}$,n_i)
7. Else then
8. $\{c_j\} \leftarrow$ SearchSupport()
9. CalculateConfigurationDLO($\{c_j\}$,n_i,$n_{i-\text{nindex}}$)
10. End if
11. End for

7. Implementation

The proposed motion planning strategy was integrated into a virtual assembly process planning and information management system developed by ourselves. A case study is conducted to demonstrate that the proposed strategy can guide DLOs along paths while avoiding collisions. The DLO moved through the puzzle is shown in Fig. 8. The mechanical properties of the DLO include that bending Young's modulus is 80,000 Pa, stretch Young's modulus is 2,00,000 Pa, and shear modulus is 50,000 Pa. The case study was implemented on a 3.2 GHz Intel Core i5 PC with 4 GB RAM.

https://doi.org/10.1017/S0263574719001103 Published online by Cambridge University Press

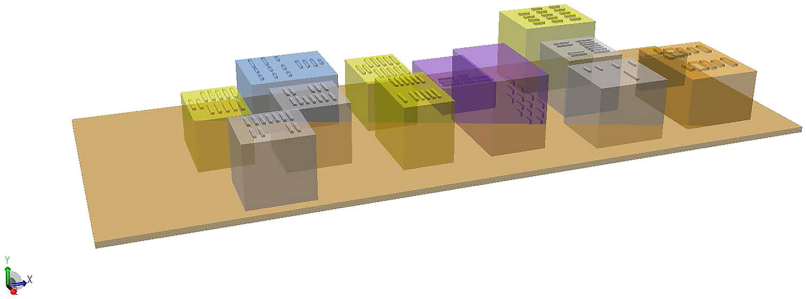

Fig. 8. The puzzle used in the case study.

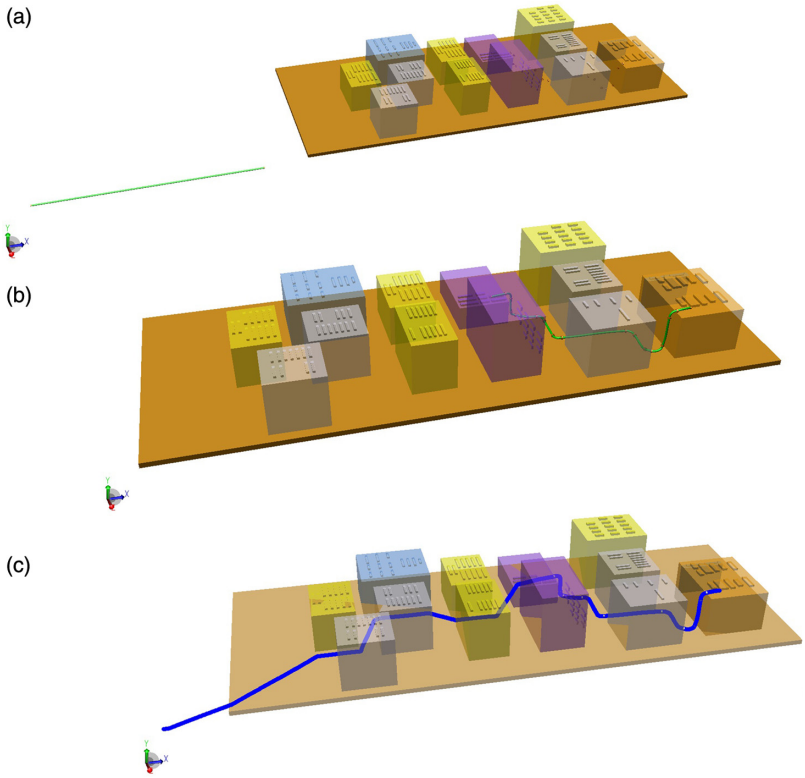

Fig. 9. The puzzle used in the case study of the motion of the DLO. (a) The initial configuration of the DLO. (b) The goal configuration of the DLO. (c) The guiding path through the puzzle. (d) The supports along the guiding path. (e) (f) Intermediate configurations of the DLO as it moved toward the goal configuration.

The parameters of the case study are as follows: the initial configuration of the DLO, from which it proceeded straight ahead, is shown in Fig. 9(a); the goal configuration of the DLO is shown in Fig. 9(b); and the algorithm selected the guiding path is shown in Fig. 9(c). The method discussed in Section 5 is used to determine the locations of the supports on the guiding path, which is shown in Fig. 9(d). Figure 9(e) and (f) shows the intermediate configurations of the DLO as it progressed through the puzzle. In this case, the length of the DLO is 1224 mm and ΔL is chosen to be 6 mm. The length of the guiding path is 2424 mm and the number of the supports is 28. The grippers draw the DLO by $\triangle L$ and the supports restrict the movement of the DLO by $\triangle L$ in each iteration step. During the motion of the DLO, the DLO is not stretched. The cost time of the guiding path generation is 10.26 s, the cost time of the support settlement is 91.95 s, and the cost time of the motion of the DLO is 60.8 s.

8. Conclusions

A novel strategy for DLO motion planning under multiple constraints was developed. DLOs were modeled as Cosserat elastic rods so that their deformation could be simulated accurately and efficiently. A RRT algorithm was used to search for the guiding path used to direct the DLOs, and then

https://doi.org/10.1017/S0263574719001103 Published online by Cambridge University Press

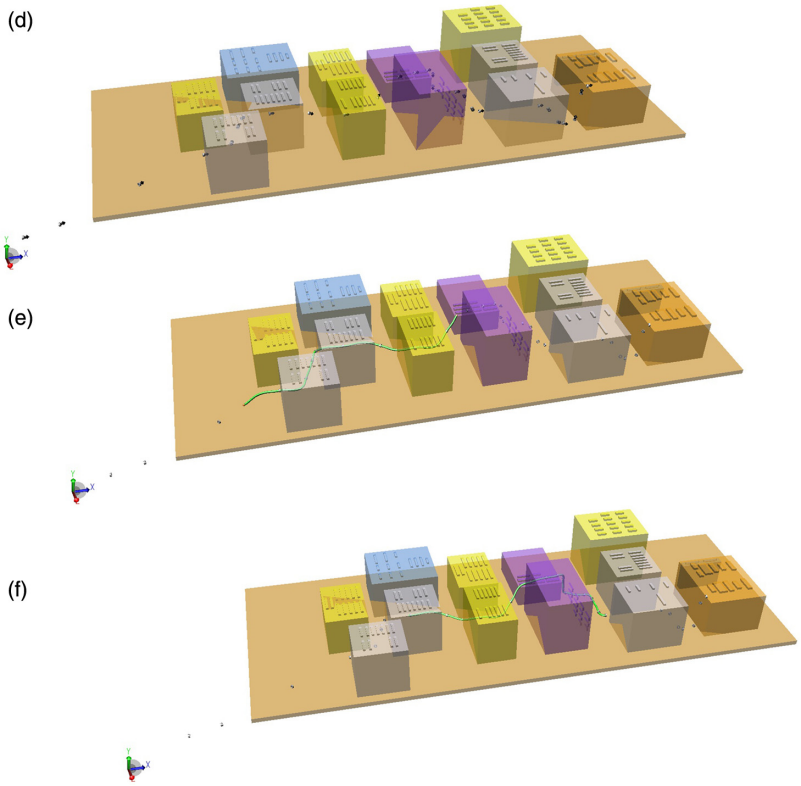

Fig. 9. (Continued)

a method was developed to determine how many supports are required and where to locate them on the guiding path. Then, the DLO was controlled by the grippers and supports to move from the initial configuration to the goal configuration. The case study demonstrated that the proposed strategy guides relatively long DLOs through complex environments without causing them to collide with obstacles. In future work, this strategy will be extended to a case where surface contact occurs between the DLO and obstacles.

Acknowledgments

This study was funded by the National Natural Science Foundation of China (Grant No. 51605030), the National Defense Fundamental Research Foundation of China (Grant No. JCKY2016204B201), and the National Defense Fundamental Research Foundation of China (Grant No. JCKY2017204B502).

Supplementary material

To view supplementary material for this article, please visit https://pan.baidu.com/s/1jyDTVzArWb58R-7QAhiO5w Extraction code:i5hx

References

1. C. Wienss, J. Scharping, S. Müller, I. Nikitin, G. Goebbels, M. Göbel and N. Hornung, "Complex Cable Bundle Simulation and Validation in VR," *Second Uksim European Symposium on Computer Modeling and Simulation*, Liverpool, UK (2008) pp. 412–417.
2. J. Xin, K. M. Koo, K. Kikuchi, A. Konno and M. Uchiyama, "Robotized assembly of a wire harness in a car production line," *Adv. Robot.* **25**(3–4), 473–489 (2011).
3. M. Moll and L. E. Kavraki, "Path Planning for Minimal Energy Curves of Constant Length," *IEEE International Conference on Robotics and Automation*, New Orleans, LA, USA (2004) pp. 2826–2831.
4. M. Moll and L. E. Kavraki, "Path planning for deformable linear objects," *IEEE Trans. Robot.* **22**(4), 625–636 (2006).
5. R. Gayle, S. Redon, A. Sud, M. C. Lin and D. Manocha, "Efficient Motion Planning of Highly Articulated Chains Using Physics-Based Sampling," *IEEE International Conference on Robotics and Automation*, Roma, Italy (2007) pp. 3319–3326.

https://doi.org/10.1017/S0263574719001103 Published online by Cambridge University Press

6. P. Jimenez, "Survey on model-based manipulation planning of deformable objects," *Robot. Comput. Integr. Manuf.* **28**(2), 154–163 (2012).
7. T. Zhang, W. Zhang and M. M. Gupta, "An underactuated self-reconfigurable robot and the reconfiguration evolution," *Mech. & Mach. Theo.* **124**, 248–258 (2018).
8. L. Greco, M. Cuomo, L. Contrafatto and S. Gazzo, "An efficient blended mixed B-spline formulation for removing membrane locking in plane curved Kirchhoff rods," *Com. Met. in App. Mech. & Eng.* **324**, 476–511 (2017).
9. H. Du, W. Xiong, H. Wang and Z. Wang, "Physical deformation configuration of a spatial clamped cable based on Kirchhoff rods," *Assem. Auto.* **38**(1), 26–33 (2018).
10. H. Xiong, Z. L. Li, J. Chang, L. You, J. J. Zhang and M. Wang, "Modelling dynamics of transmission conductors with Cosserat rod," *Com. Assist. Mech. & Eng. Sci.* **20**(1), 73–79 (2013).
11. N. Lv, J. Liu, X. Ding and H. Lin, "Assembly simulation of multi-branch cables," *J. Manu. Sys.* **45**, 201–211 (2017).
12. T. Hermansson, R. Bohlin, J. S. Carlson and R. Söderberg, "Automatic assembly path planning for wiring harness installations," *J. Manu. Sys.* **32**(3), 417–422 (2013).
13. T. Bretl and Z. McCarthy, "Quasi-static manipulation of a Kirchhoff elastic rod based on a geometric analysis of equilibrium configurations," *Int. J. Robot. Res.* **33**(1), 48–68 (2014).
14. O. Roussel, M. Taix and T. Bretl, "Efficient Motion Planning for Quasi-Static Elastic Rods Using Geometry Neighborhood Approximation," *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Besacaon, France (2014) pp. 1024–1029.
15. 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," *IEEE International Conference on Robotics and Automation*, Seattle, WA, USA (2015) pp. 3116–3121.
16. R. Gayle, M. C. Lin and D. Manocha, "Constraint-Based Motion Planning of Deformable Robots," *IEEE International Conference on Robotics and Automation*, Barcelona, Spain (2005) pp. 1046–1053.
17. M. Grégoire and E. Schömer, "Interactive simulation of one-dimensional flexible parts," *Comput. Aid. Des.* **39**(8), 694–707 (2007).
18. D. K. Pai, "STRANDS: interactive simulation of thin solids using cosserat models," *Comput. Graph. Forum* **21**(3), 347–352 (2002).
19. J. Spillmann and M. Teschner, "CORDE: Cosserat Rod Elements for the Dynamic Simulation of One-Dimensional Elastic Objects," *ACM Siggraph/Eurographics Symposium on Computer Animation*, San Diego, California (2007) pp. 63–72.
20. J. Denny, E. Greco, S. Thomas and N. M. Amato, "MARRT: Medial Axis Biased Rapidly-Exploring Random Trees," *IEEE International Conference on Robotics and Automation*, Hong Kong, China (2014) pp. 90–97.
21. H. Blum, *Models for the Perception of Speech and Visual Form*, (M.I.T. Press, Cambridge, MA, 1967).
22. H. Zhu, Y. Liu and J. Zhao, "Generation of hierarchical multi-resolution medial axis for CAD models," *Adv. Eng. Soft.* **94**, 20–31 (2016).
23. T. K. Dey and W. Zhao, "Approximate medial axis as a voronoi subcomplex," *Com. Aided Design* **36**(2), 195–202 (2004).
24. G. K. Viswanathan, A. Murugesan and K. Nallaperumal, "A Parallel Thinning Algorithm for Contour Extraction and Medial Axis Transform," *IEEE International Conference on Emerging Trends in Computing, Communication and Nanotechnology*, Tirunelveli, India (2013) pp. 606–610.
25. Y. Liu, C. Xian, M. Li, H. Xia and S. Gao, "A local adaptation-based generation method of medial axis for efficient engineering analysis," *Eng. Comput.* **29**(2), 207–223 (2013).
26. S. M. Lavalle, "Rapidly-Exploring Random Trees: A New Tool for Path Planning," *Technical Report*, 293–308 (1998).
27. J. Nasir, F. Islam, U. Malik, Y. Ayaz, O. Hasan, M. Khan and M. S. Muhammad, "RRT*-Smart: a rapid convergence implementation of RRT*," *Int. J. Adv. Robot. Sys.* **10**, 299 (2013).
28. M. Brunner, B. Brüggemann and D. Schulz, "Hierarchical Rough Terrain Motion Planning Using an Optimal Sampling-Based Method," *IEEE International Conference on Robotics and Automation*, Karlsruhe, Germany (2013) pp. 5539–5544.
29. O. Adiyatov and H. A. Varol, "Rapidly-Exploring Random Tree Based Memory Efficient Motion Planning," *IEEE International Conference on Mechatronics and Automation*, Takamatsu, Japan (2013) pp. 354–359.
30. Y. Zhu and J. Meng, "Real-time collision detection and response techniques for deformable objects based on hybrid bounding volume hierarchy," *COMPEL* **28**(6), 1372–1385 (2009).