Contact-aware Shaping and Maintenance of Deformable Linear Objects With Fixtures

Kejia Chen[1], Zhenshan Bing[1], Fan Wu[1], Yuan Meng[1], Sami Haddadin[1], Alois Knoll[1†]

***Abstract*— Studying the manipulation of deformable linear objects has significant practical applications in industry. In this paper, we propose a new framework to control and maintain the shape of deformable linear objects with two robot manipulators utilizing environmental contacts. The framework is composed of a shape planning algorithm which automatically generates appropriate positions to place fixtures, and an object-centered skill engine which includes task and motion planning to control the motion and force of both robots based on the object status. The status of the deformable linear object is estimated online utilizing visual as well as force information. The framework manages to handle a cable routing task in real-world experiments using two Panda robots and especially achieves contact-aware and flexible clip fixing with challenging fixtures.**

I. INTRODUCTION

The manipulation of deformable linear objects (DLOs) is a common and yet critical step in various industrial manufacturing processes. One typical example could be the wire harness assembly, where cables need to be installed on a board or panel [1]. Due to challenges in accurate modeling, real-time state estimation and multi-robot manipulation of the DLO [2], the handling of such deformable cables still heavily relies on manual labor. Researchers have proposed various frameworks based on different solutions to each sub-problem. Early works focused on achieving a desired manipulation solely with robot contacts [3]–[5]. Following the pioneering work by Zhu et al. [6], more recent studies have focused on achieving complex shapes of DLOs using environmental contacts, including fixtures and clips [7]–[10]. However, previous works employed circular fixtures or loosely fitted channel fixtures (as shown in Fig. 1(b) and (c)), which results in minimal contact forces and is unable to maintain the shape of the DLO. Additionally, most of the works consider only visual perception and robot motion planning [7]–[9], even though the task includes abundant force information. Moreover, it's important to mention that all of these works began by placing the fixtures in predetermined or random positions, without addressing the issue of finding appropriate fixture positions based on a desired shape.

In this paper, we propose a new framework to manipulate a DLO, which addresses the limitations of prior work by generating fixture positions automatically from the desired shape, leveraging clip-like fixtures (see Fig. 1(d)), and integrating force sensing for manipulation. Real-world experiments have validated the effectiveness of our framework in shape control of DLO and demonstrated its advantage over traditional motion planning for achieving a flexible clip fixing process.

[1]K. Chen, Z. Bing, F. Wu, Y. Meng, S. Haddadin and A. Knoll are with the Department of Informatics, Technical University of Munich, Germany. kejia.chen@tum.de

[†]The authors acknowledge the financial support by the Bavarian State Ministry for Economic Affairs, Regional Development and Energy (StMWi) for the Lighthouse Initiative KI.FABRIK (Phase 1: Infrastructure as well as the research and development program under, grant no. DIK0249). Please note that S. Haddadin has a potential conflict of interest as a shareholder of Franka Emika GmbH.

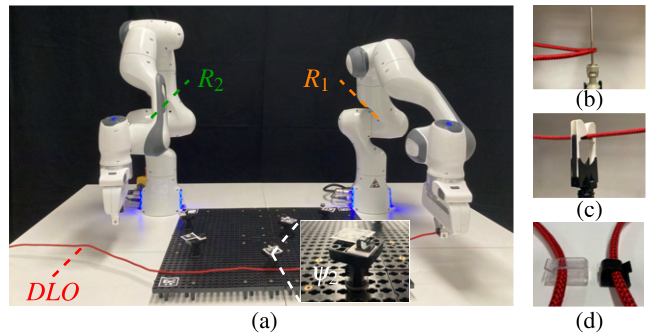

Fig. 1: *(a) Setup of real-world experiment. The clip fixture ψ is effective in maintaining the shape of the cable, but also poses a challenge in terms of handling. (b) Circular fixture. (c) Channel fixture. (d) Clip fixtures.*

II. PROBLEM FORMULATION AND METHOD OVERVIEW

The proposed framework consists of two primary components, as illustrated in Fig. 2: the fixture placing algorithm (in orange box) and a skill engine for task and motion planning (in blue box).

Taking the desired shape S^* as input, the fixture placing algorithm generates appropriate fixture positions. Fixtures placed on generated positions should help the DLO to maintain its shape. The task and motion planning system then controls the motion and force of both robots based on an online estimation of the DLO status, which is defined as a 4-tuple $(S_t, \mathbf{x}_h, T, C)$.

- Real-time shape S_t. This is obtained from raw images captured by visual sensors, and can be represented as a sequence of vertices $V = \{v_i\}, i \in \{1,2,...N\}$ and edges $E = \{e_i\}, i \in \{1,2,...N-1\}$. S^* can be represented in a similar way as (V^*, E^*).
- Position $\mathbf{x}_h$. This is defined as the position of the DLO "head", i.e., the end that is grasped by the robot.
- Tensity $T \in \{0,1\}$. Tensity is tracked using force sensors as an indicator of deformation resistance. We define $T = 1$ if the component of external force f_{ext} along DLO is above a certain threshold.
- Contact $C \in \{0,1\}$. This state is obtained from force sensor to indicate DLO's contacts with the environment. Only when the DLO stays taut (T=1) are detected contacts with the environment reliable.

Based on the observed DLO status, our skill engine selects one from the two designed skill, namely, the shape tracking skill and the clip fixing skill. The shape tracking skill uses the visual information and generates joint trajectories for both robots, enabling them to track the desired shape S^* and to move to each fixture. The clip fixing skill uses T and C and applies force that changes in multiple stages, firstly stretching and then pushing the DLO into the fixture. These two skills are applied in a repetitive manner, until all the intended contacts have been established.

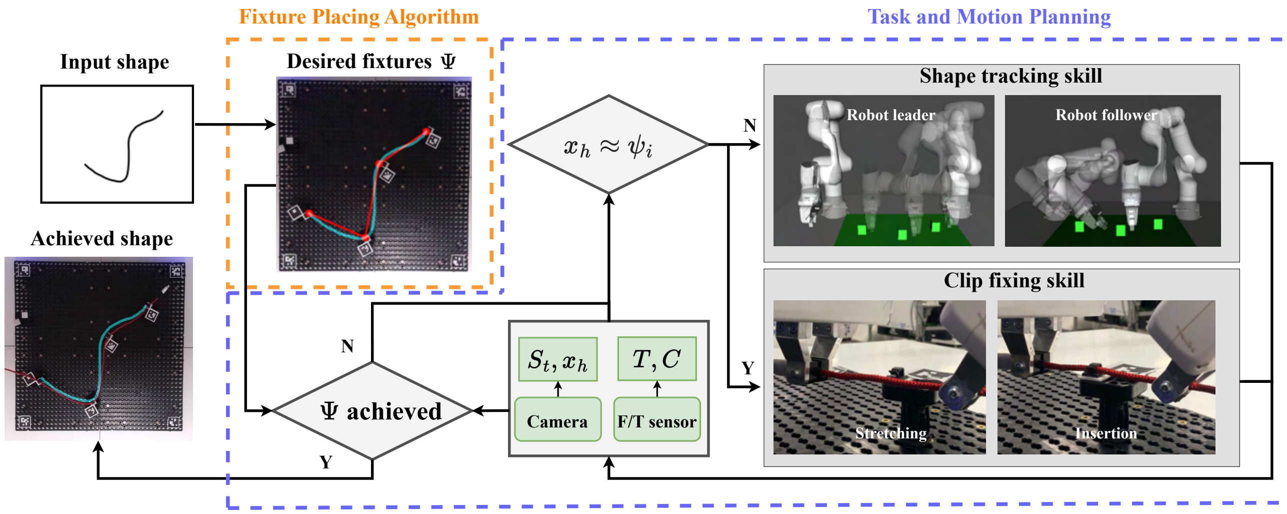

Fig. 2: *Framework Overview. The framework takes a shape (marked in green) as input, and the fixture placing algorithm (in the dashed orange box) generates appropriate positions to place clips. Based on information from the camera and force-torque sensors, the skill engine for task and motion planning (in the dashed blue box) then executes either the shape tracking skill (top) or the clip fixing skill (bottom) to establish contacts between the DLO and each fixture, and finally achieves the desired shape.*

III. Fixture Planning

Assuming that S^* lies on a plane M without any entanglements, the fixture placing algorithm generates a set of fixture positions $\Psi = \{\psi_i\}, \psi_i \in \mathbb{R}^2$ on M. Without loss of generality, we assume the DLO segments between each consecutive fixture pair are approximately straight (red line segments in Fig. 2). Under such an assumption, fixtures should be placed at where the desired shape "bends" the most. Therefore, we generate fixture position based on the radius of curvature (ROC) r_i of each vertex in V^*. The first set of fixtures are generated at the local maximums and minimums of the spline $r(v)$ of r_i, splitting S^* into a set of curve segments $S^* = \{s_i^*, \Psi\}, i \in \{1,2,...,L\}$. In each curve segments s_i, additional fixtures are generated progressively to further reduce the error J_s between the resulting shape $S = \{s_i, \Psi\}$ and S^*:

$$J_s = \sum_L \int_0^{l_i} ||s_i^* - s_i||. \tag{1}$$

This step is repeated until J_s is below a certain threshold J_s^*.

All the generated fixture positions ψ_i are examined under the global constraint of distance between two neighboring fixtures $d(\psi_i, \psi_j)$. The generated positions are considered valid only if the distance lies in a space $\mathbb{C}(d)$, which is defined by robot gripper size l_g as the lower limit and by maximum allowable sag between consecutive fixtures J_d as the upper limit:

$$\mathbb{C}(d) = \{d \in \mathbb{R}^+ |\ d > l_g, f_{\text{deform}}(\kappa, d) < J_d\}, \tag{2}$$

where $f_{\text{deform}}(\kappa, d)$ is the sag of the DLO with stiffness κ between fixtures at a certain distance d to each other.

IV. DLO Manipulation

Inspired by the repetitive pattern in the manual cable routing process, we design two manipulation skills as well as an object-centered high-level skill engine to handle this routine.

A. Shape Tracking Skill

For each generated ψ_i, the shape tracking skill controls the motion of both robots to reach a desired pose pair $(\mathbf{p}_1^i, \mathbf{p}_2^i)$, facilitating the upcoming clip fixing skill. Each pose is defined as $\mathbf{p}^i = ((\mathbf{x}', z_d'), \mathbf{w}')^T$, where $\mathbf{x}' \in \mathbb{R}^2$ is the projection of a desired end-effector position $\mathbf{x} \in \mathbb{R}^3$ onto M and $\mathbf{w}'$ represents the orientation. The distance between M and robots, denoted as z_d', remains constant and obstacle-free throughout the movement, but decreases to zero once the robots approach a fixture. Therefore, we consider only the planning of $\mathbf{x}'$ and $\mathbf{w}'$ in the shape tracking.

The two robots collaborate in a "master-slave" manner. For the robot leader, the desired position $\mathbf{x}_1'$ is selected from V^* under a distance constraint to ψ_i. For the robot follower, the skill aligns its end-effector with that of the leader and grasps the DLO at the fixture. To select a proper grasping position $\mathbf{x}_2'$, the online shape of DLO S_t is tracked by a data-driven method FASTDLO [11].

Collision avoidance with the leader and the fixtures are also considered in the motion planning of the robot follower. This can be formulated as an optimization problem that following the selection of $\mathbf{x}_2'$, the optimal orientation $\mathbf{w}_2^*$ should keep the robot follower at the maximum distance to the robot leader and two neighboring fixtures:

$$\mathbf{w}_2^* = \arg\max_{\mathbf{w}_2'} \textstyle\sum_{\mathbf{p}_k \in \{\mathbf{p}_1, \psi_1, \psi_2\}} ||\mathbf{p}_2 - \mathbf{p}_k||. \tag{3}$$

As $\mathbf{p}_2^*$ is selected for the end-effector, we plan the motion of the robot follower in joint space with RRT* algorithm to avoid possible collision between two robot arms in the process of motion.

B. Clip Fixing Skill

The clip fixing skill controls both robots to apply forces on the grasped segment of DLO to push it into the clip. We formulate the fixing skill using an adaptive force impedance controller [12] [13]. The clip fixing skill is defined as a directed transition graph of manipulation primitives (MPs). A single MP consists of a desired linear velocity $\dot{\mathbf{x}}^d \in \mathbb{R}^3$ and feedforward force $\mathbf{f}^d \in \mathbb{R}^3$. All the transitions between MPs are triggered by changes in T and C status to achieve an adaptive and contact-aware fixing process. Once the DLO is fitted into the clip, it realizes the success and the stops the robots from applying forces or moving further.

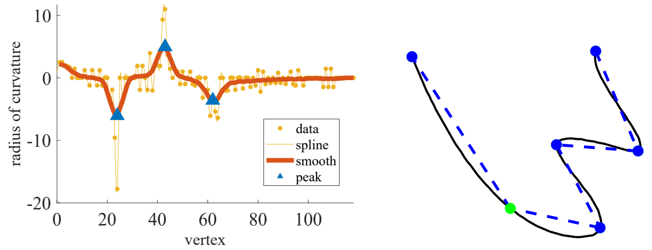

Fig. 3: *Fixture positions generated by the placing algorithm on a desired shape and corresponding ROCs.*

Pre-contact stretching To measure the contact force with environmental fixtures accurately, the object is firstly stretched by both robots in x' axis to be taut:

$$\dot{\mathbf{x}}_1^d = [0,\ 0,\ 0]^T, \quad \mathbf{f}_1^d = [f_s,\ 0,\ 0]^T, \tag{4}$$

where f_s is the desired magnitude of stretching force. The robot follower in this MP will move and apply forces in an opposite direction to the leader.

Contact establishment The taut DLO is then moved in the opposite of the clip opening direction $\mathbf{u}$:

$$\dot{\mathbf{x}}_1^d = -\mathbf{u}, \quad \mathbf{f}_1^d = [f_s,\ 0,\ 0]^T, \tag{5}$$

until it has contact with the fixture. C is updated by detecting the rise of external force projected in the moving direction f_u^{ext}. The position of the contact point is memorized as $\mathbf{x}_c$. The robot follower takes the same motion and forces as the robot leader.

Push in Once the contact happens, the DLO is pushed into the clip:

$$\dot{\mathbf{x}}_1^d = [0,\ 0,\ 0]^T, \quad \mathbf{f}_1^d = [f_s,\ 0,\ 0]^T + f_p \cdot (-\mathbf{u}), \tag{6}$$

until the DLO moves further than the contact point $\mathbf{x}_c \cdot (-\mathbf{u}) < \mathbf{x}_t \cdot (-\mathbf{u})$ and loses contact with the fixture $f_u^{\text{ext}} < F_c$. The robot follower takes the same motion and forces as the robot leader.

C. Skill-based planning

The skill engine at the high level selects from the two manipulation skills to establish contact with each fixture following the logic presented in Fig. 2. The shape tracking skill is employed until both robots reach appropriate poses close to ψ_i. After that, the follower will also grasp the DLO and the clip fixing skill is activated. This process is repeated until all fixtures generated in Section III are achieved.

V. Experiments

We used two 7 DOF Franka Emika Panda robots for the real-world DLO manipulation experiments, both of which are equipped with joint torque sensors and provide 6-axis force torque estimation at the end-effectors. Since the clips in our task are small, we mount each clip on an additional platform to form a fixture (see Fig. 1). The positions and orientations of fixtures are estimated from markers by an Azure Kinect camera on the top.

A. Fixture Positioning

To show the capability of our placing algorithm to handle more complex shapes, we run the test with l'_g set as half of the real gripper size $l'_g = 0.5 \cdot l_g$. One example of placing results are presented in Fig. 3. The blue markers represent fixtures placed at local maximum and minimum of ROC and the green markers represent fixture positions generated recursively to reduce J_s.

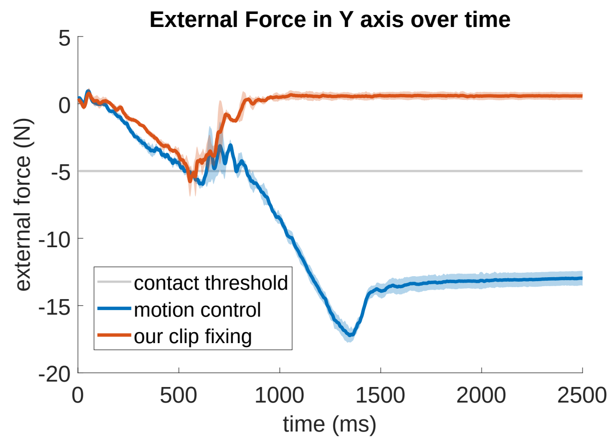

Fig. 4: *Clip fixing skill evaluation. The curves describe contact forces in* $\mathbf{u}$ *direction during clip fixing. The comparison between using the proposed clip fixing skill and using only motion control can also be seen in the video.*

B. Clip Fixing

To evaluate our contact-aware clip fixing skill, we compare the detected external force in the fixing process using our skill and using purely motion control. The detected f_{ext} from the robot leader is plotted in Fig. 4, where one falling represents one contact with the fixture. Our clip fixing skill experiences only one contact stage, and stops moving further or applying forces once the cable is fitted snugly in. In contrast, the fixing by motion control keeps moving regardless of contact, and hits the back end of the clip. In the real production process, this blindness may lead to damage to the clips.

C. Cable Routing

Finally, we evaluate the whole framework with a drawn desired shape as input and a red usb extension cable as object to be manipulated. Fixtures are placed at four generated positions. The motion planning in Section IV-A is implemented with MoveIt [14] using the Open Motion Planning Library [15]. After the robot leader guides the DLO to one clip fixture, the robot follower approaches the DLO while avoiding collision with the leader and grasps it. Both robots then apply stretching and pushing force to the DLO to insert it into the clip fixture. As contacts with all fixtures are established, the robots release their grasp on the DLO, leaving it securely fixed in place by the clips. The whole motion and clip fixing process can be found in the video at `https://youtu.be/YbVDOgT3vc4`.

VI. Conclusion

We presented a framework to manipulate DLO using environmental contacts, with DLO status estimated online from visual and force information. Our fixture placing algorithm firstly generates appropriate positions of fixtures based on desired shape. An object-centered skill engine then selects designed manipulation skills based on DLO status to establish contact with each fixture. In the real-world experiments, our framework successfully controls two robots to manipulate the DLO into the desired shape. Future work will be using grippers with more degrees of freedom to achieve more complex shapes.

References

[1] Y. She, S. Wang, S. Dong, N. Sunil, A. Rodriguez, and E. Adelson, "Cable manipulation with a tactile-reactive gripper," *The International Journal of Robotics Research*, vol. 40, no. 12-14, pp. 1385–1401, 2021.

[2] H. Yin, A. Varava, and D. Kragic, "Modeling, learning, perception, and control methods for deformable object manipulation," *Science Robotics*, vol. 6, no. 54, p. eabd8803, 2021.

[3] B. Jia, Z. Hu, J. Pan, and D. Manocha, "Manipulating highly deformable materials using a visual feedback dictionary," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 239–246.

[4] J. Zhu, B. Navarro, P. Fraisse, A. Crosnier, and A. Cherubini, "Dual-arm robotic manipulation of flexible cables," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 479–484.

[5] D. Kruse, R. J. Radke, and J. T. Wen, "Collaborative human-robot manipulation of highly deformable materials," in *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2015, pp. 3782–3787.

[6] J. Zhu, B. Navarro, R. Passama, P. Fraisse, A. Crosnier, and A. Cherubini, "Robotic manipulation planning for shaping deformable linear objects with environmental contacts," *IEEE Robotics and Automation Letters*, vol. 5, no. 1, pp. 16–23, 2019.

[7] S. Huo, A. Duan, C. Li, P. Zhou, W. Ma, H. Wang, and D. Navarro-Alarcon, "Keypoint-based planar bimanual shaping of deformable linear objects under environmental constraints with hierarchical action framework," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5222–5229, 2022.

[8] S. Jin, W. Lian, C. Wang, M. Tomizuka, and S. Schaal, "Robotic cable routing with spatial representation," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5687–5694, 2022.

[9] G. A. Waltersson, R. Laezza, and Y. Karayiannidis, "Planning and control for cable-routing with dual-arm robot," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 1046–1052.

[10] F. Süberkrüb, R. Laezza, and Y. Karayiannidis, "Feel the tension: Manipulation of deformable linear objects in environments with fixtures using force information," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 11 216–11 222.

[11] A. Caporali, K. Galassi, R. Zanella, and G. Palli, "Fastdlo: Fast deformable linear objects instance segmentation," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9075–9082, 2022.

[12] C. Yang, G. Ganesh, S. Haddadin, S. Parusel, A. Albu-Schaeffer, and E. Burdet, "Human-like adaptation of force and impedance in stable and unstable interactions," vol. 27, no. 5, pp. 918–930.

[13] L. Johannsmeier, M. Gerchow, and S. Haddadin, "A framework for robot manipulation: Skill formalism, meta learning and adaptive control," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 5844–5850.

[14] D. Coleman, I. Sucan, S. Chitta, and N. Correll, "Reducing the barrier to entry of complex robotic software: a moveit! case study," *arXiv preprint arXiv:1404.3785*, 2014.

[15] I. A. Sucan, M. Moll, and L. E. Kavraki, "The open motion planning library," *IEEE Robotics & Automation Magazine*, vol. 19, no. 4, pp. 72–82, 2012.