Offline Reinforcement Learning for Shape Control of Deformable Linear Objects from Limited Real Data

Rita Laezza[1], Mohammadreza Shetab-Bushehri[2], Erol Özgür[2], Youcef Mezouar[2] and Yiannis Karayiannidis[3]

***Abstract*— Shape control is a challenging manipulation problem which requires driving an object to a desired shape. The difficulty comes from the under-actuation of the object being manipulated, which depends on both the object and environment dynamics. Several shape servoing techniques make use of purely geometric heuristics to manipulate the object into the desired shape. While reliable in contexts with approximately linear behavior and simple environmental contacts, they can fail in tasks with more complex dynamics. An alternative approach is to have a robot learn directly from real experience how to achieve a shape control task, e.g. using Reinforcement Learning (RL). In this work we investigate offline RL for shape control of a Deformable Linear Object (DLO) manipulation task, with long-term effects. We propose a data augmentation approach, to limit the amount of experimental data which needs to be collected. With this augmentation, the TD3+BC algorithm is able to outperform the classical shape servoing baseline.**

I. Introduction

Shape control is a type of manipulation problem which is unique to deformable objects, where the goal is not only to change the pose of an object, but also its shape. The classical approach to this problem is referred to as shape servoing [1]–[4]. Despite significant progress, classical methods suffer from several limitations, including computational complexity and modeling inaccuracy due to difficulties in identifying the mechanical parameters of the object and its interaction with the environment. Such methods mostly rely on an instantaneous error and local models, therefore objects with complex material properties remain to be explored, since their manipulation exhibits more long-term effects.

Recent progress in deformable object manipulation has been fueled by advances in Deep Learning (DL) research [5]–[10]. DL methods have the advantage of indirectly encoding the dynamics of the manipulation task, without requiring extensive engineering work in order to model the object-environment interaction. This is appealing due to the large variety of deformation behaviors across different classes of deformable objects [11]. Reinforcement Learning (RL) in particular, allows a robot to learn from experience and optimize a policy towards long-term objectives. However, a key obstacle to applying RL in real-world applications is the need to collect online data on a robotic setup, which is time consuming and potentially unsafe. An alternative is to first collect real data and then use it to train offline RL algorithms.

[1] Division of Systems and Control, Department of Electrical Engineering, Chalmers University of Technology, Sweden (laezza@chalmers.se)

[2] CNRS, Clermont Auvergne INP, Institut Pascal, Université Clermont Auvergne, France ({m.r.shetab, erolozgur}@gmail.com youcef.mezouar@sigma-clermont.fr)

[3] Department of Automatic Control, Lund University, Sweden (yiannis@control.lth.se)

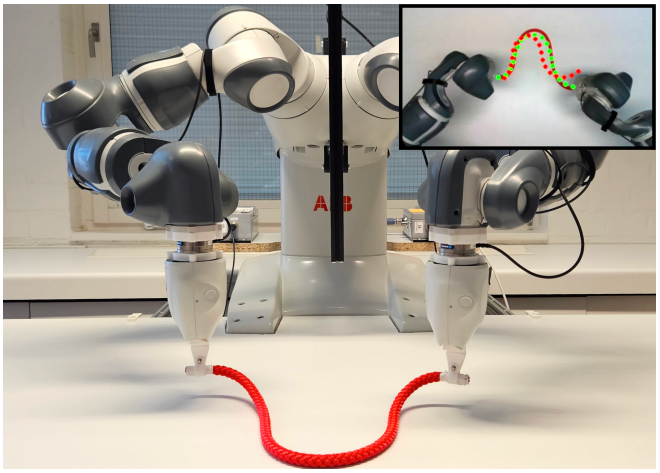

Fig. 1: Dual-arm ABB YuMi robot manipulates a rope on a table. A fixed Intel RealSense camera provides a top-view of the workspace. The field of view of the camera is shown in the top right corner, with the overlay of the current shape tracking in green and the desired shape in red.

In this work we tackle a Deformable Linear Object (DLO) shape control problem with surface interactions, using offline RL. The manipulated DLO is a rope, which due to its low bending stiffness and its rough texture, leads to complex dynamics when in contact with a surface. We investigate the efficacy of the TD3+BC offline RL algorithm [12] to learn a control policy, and compare it with a well-established shape servoing method proposed by Berenson et al. [1]. We further propose a simple data augmentation approach which improves the baseline results and enables learning of more complex manipulation techniques, which the shape servoing method is unable to perform.

II. Problem Statement

Prior work has mainly approached the problem of rope manipulation as a sequence of pick and place motions along the rope on a smooth surface, to achieve a desired shape [13]–[15]. In contrast, we propose to solve the task as a bi-manual planar control problem. More specifically, we want to control 6 Degrees of Freedom (DOFs), where 4 DOFs relate to the translation on an xy-plane, while the other 2 DOFs relate to the orientation of the grippers about the z-axis. This setting precludes the need to re-grasp but brings new challenges:

i) the middle part of the rope is not directly affected by the movement of the grippers, due to low stiffness.
ii) contact between the rope and the workspace affects its shape, due to friction.

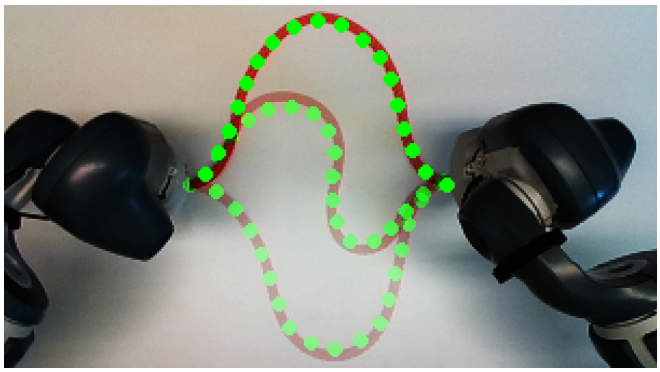

Fig. 2: Overlay of three possible DLO shapes with identical gripper poses. One can intuitively picture that for the top and bottom shapes, an additional top-down translation motion was necessary, while for the middle shape a counterclockwise rotation of the grippers must have occurred.

A key property is that multiple DLO shapes can be achieved with the grippers in the same pose, depending on the preceding motion, as shown in Fig. 2.

III. Methods

In order to tackle the proposed problem, we make use of a deformable object tracking method described in Section III-A. The tracked points are then used as input to both the Berenson [1] method for shape servoing and the Artificial Neural Network (ANN) policy. Section III-B, introduces the RL formulation of the problem and the proposed data collection and augmentation procedures.

A. DLO Tracking

To track the rope, we employ the algorithm developed by Shetab-Bushehri et al. [4], using a depth camera. The algorithm is initialized by forming a lattice around a reference point cloud of the DLO in its rest shape and then binding the two together by geometrical constraints. During tracking, correspondences between the reference point cloud and the point cloud captured by the depth camera are found in each frame. These correspondences along with a deformation model are then applied as constraints to the lattice. We use the As-Rigid-As-Possible (ARAP) model as the deformation model. As a result, the lattice's shape is updated and thus the DLO's reference point cloud becomes aligned with the captured point cloud, while keeping its local rigidity.

In practice, the reference point cloud of the DLO is obtained by creating a 3D model of a semi-cylinder with the dimensions of the (visible) rope. We then form a lattice with $18 \times 3 \times 3$ vertices around this reference point cloud, with the first dimension along the length of the rope. In order to separate the captured point cloud of the DLO from the background, we first employ a simple segmentation method on the RGB image, via a color filter. Then the depth information is aligned to the segmented image, and the points outside the DLO are disregarded. Finally, a subset of the vertices closest to each gripper are set as controlled points, meaning that their 3D coordinates are updated based on the grippers' poses in each frame. This helps improve the quality of the tracking, since the gripper's poses are known, and just need to be transformed into the camera frame.

B. Reinforcement Learning

The RL problem is formulated as a Markov Decision Process (MDP). We frame the task described in Section II as an episodic MDP, defined as a tuple $(\mathcal{S}, \mathcal{A}, p, r, \gamma)$, where γ is the discount factor and $\mathcal{S}$ and $\mathcal{A}$ are continuous state and action spaces, respectively. The probability density function $p(s_{t+1}|s_t, a_t)$ represents the probability of transitioning to state s_{t+1}, given the current state s_t and action a_t, with $s_t, s_{t+1} \in \mathcal{S}$ and $a_t \in \mathcal{A}$. The dynamics of the interaction between the robot and the DLO $p(s_{t+1}|s_t, a_t)$ are unknown. Instead, real data is collected with the experimental setup shown in Fig. 1, which is used to learn a deterministic policy $\pi(s) = a$, based on a reward function $r : \mathcal{S} \rightarrow \mathbb{R}$. The return is defined as the sum of discounted future rewards: $G_t = \sum_{k=t}^{T} \gamma^{k-t} r(s_k)$, where γ is a discount factor and, t and T are the current and terminal state's indices, respectively. RL algorithms aim to maximize the expected return conditioned on state-action pairs, i.e. the action-value $Q(s, a)$.

In offline RL, the goal is to learn a policy based solely on a fixed dataset, $\mathcal{D}$. This is advantageous in robotics, however it also adds new challenges, given that agents tend to estimate the value of unseen state-action pairs incorrectly. Fujimoto et al. [12] propose the TD3+BC algorithm to mitigate this problem, which works by modifying the policy update step of the TD3 algorithm [16] with a Behavior Cloning (BC) regularization term (in blue): $\pi = \arg\max_\pi \mathbb{E}_{(s,a)\sim\mathcal{D}} \big[\lambda Q(s, \pi(s)) - (\pi(s) - a)^2 \big]$, where $\lambda = \frac{1}{M} \sum_{(s_i,a_i)} \frac{\alpha}{|Q(s_i,a_i)|}$ is computed over batches of M state-action pairs and α controls the strength of the regularization.

RL Formulation: The shape of the DLO is represented as the mean xy coordinates of each 3×3 lattice cross-section, $\boldsymbol{q}^i \in \mathbb{R}^{18\times2}$, where $i \in \{c, d\}$ indicates the current and desired shapes. The position and orientation of the grippers is denoted by $\boldsymbol{p}_j^i \in \mathcal{W}_j \subset \mathbb{R}^2$, $\boldsymbol{o}_j^i \in [-\frac{\pi}{4}, \frac{\pi}{4}]$, where $\mathcal{W}_j$ is a safe[1] workspace, $j \in \{l, r\}$ indicates the left/right gripper and, $i \in \{c, d\}$ the current and desired poses.

The MDP state is then defined as a 1D vector:

$$s = [\bar{\boldsymbol{q}}^d,\ \bar{\boldsymbol{q}}^c,\ \boldsymbol{p}_l^c,\ \boldsymbol{p}_r^c,\ \boldsymbol{o}_l^c,\ \boldsymbol{o}_r^c] \in \mathbb{R}^{78} \tag{1}$$

where, the bar over the DLO shapes indicates flattened vectors, i.e. $\bar{\boldsymbol{q}}^i \in \mathbb{R}^{36}$.

The MDP action space $\mathcal{A}$, is defined as the gripper poses, i.e. $a = [\boldsymbol{p}_l^d,\ \boldsymbol{p}_r^d,\ \boldsymbol{o}_l^d,\ \boldsymbol{o}_r^d] \in \mathbb{R}^6$. A simple point-to-point motion is then generated with a timing law dependent on the distance between the current and desired gripper positions.

Finally, the reward function we intend to maximize is defined based on the root mean squared error (RMSE):

$$r(s) = -\sqrt{\frac{1}{N} \sum_{k=1}^{N} \left(\boldsymbol{q}_k^d - \boldsymbol{q}_k^c\right)^2} \tag{2}$$

where $N = 18$ is the number of lattice cross-sections. The reward was chosen to be comparable with shape-servoing approaches which attempt to minimize the RMSE.

[1]Each gripper moves inside its own mutually exclusive area to prevent collisions and DLO entanglements. Constraints are also added to keep the DLO from being overstretched and inside the field of view of the camera.

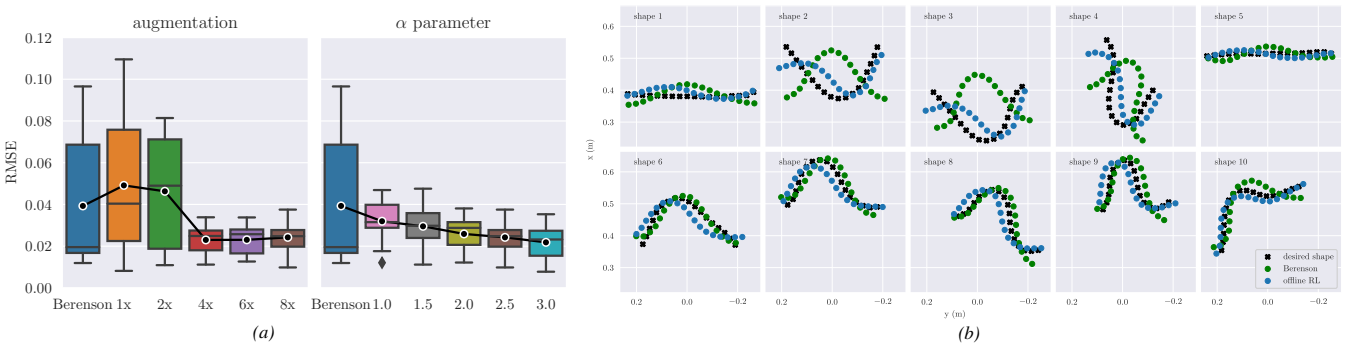

Fig. 3: (a) Boxplots with results for the augmentation experiment (left) and the impact of BC regularization (right), determined by the α parameter. Note that the 8x *entry on the left is the same as the* 2.5 *entry on the right. The black dots indicate the mean across the 10 test shapes. (b) Comparison between final shapes and desired shape. Note how the offline RL policy succeeded in inverting the curvature, while the shape-servoing remained at a local minimum.*

Data Collection: Since offline RL lacks the possibility of exploration, we must ensure adequate coverage of the state-action space to make learning feasible [17]. To that end, we developed a data collection procedure which deforms the rope into varied shapes. It works by randomly sampling positions from the safe workspace, $\mathcal{W}_j$, and orientations from a range $[-\frac{\pi}{4}, \frac{\pi}{4}]$ rad. For each iteration, there is a 0.3 probability of sampling a new gripper pose for either the left, right or both grippers. With the remaining 0.1 probability, a predetermined semi-random motion sequence is executed which leads to inversion of the DLO's curvature. A point-to-point motion of the grippers is then used to generate an episode, where intermediate DLO shapes and gripper poses are saved as transitions and the final shape is used as $\boldsymbol{q}^d$.

Data Augmentation: We propose a simple data augmentation procedure that improves the performance of the offline RL method, similar to the concept of Hindsight Experience Replay (HER), which learns from failed experience [18]. Thanks to our goal-conditioned policy (i.e. state includes the goal), it is possible to artificially generate new episodes by setting intermediate DLO shapes within the dataset, as desired shapes and recomputing the reward accordingly. This helps reduce the volume of experimental data needed.

IV. Experiments

In this section, we describe the experimental setup and present results validating our approach for data augmentation. We further explore the impact of the BC regularization term, of the TD3+BC algorithm.

Experimental setup: A small dataset of 1010 episodes was created, with the last 10 being used for testing alone. Offline RL policies were then trained using the TD3+BC algorithm with 1×10^6 environment steps. Note that earlier policies may be better, but finding a criteria for early stopping in offline RL is an open research question, since there is no clear measure for over-fitting [19].

To evaluate each policy, the current state is given as input to the ANN and its output is used to update the grippers' desired poses, at 2 Hz. A Cartesian controller is constantly driving the grippers to the current desired poses. A sequence of 10 test shapes determines the value of $\boldsymbol{q}^d$, for a total of 40 s (without a reset). This time limit was chosen to help speed up the testing, while still allowing enough time for the policies to converge to a final shape. Note that the same methodology was used to test the shape servoing baseline [1], and there was a non-exhaustive attempt to tune its parameters.

Augmentation: To evaluate the benefit of the augmentation procedure, four levels of augmentation were tested, namely $\{1\text{x}, 2\text{x}, 4\text{x}, 6\text{x}, 8\text{x}\}$ where 1x refers to no augmentation, 2x indicates that the amount of data was doubled, etc. The regularization parameter was fixed to $\alpha = 2.5$, as in the original paper. From Fig. 3a (left), it is clear that the augmentation indeed helps achieve a lower average RMSE. This is likely due to the increase of desired shapes found in the augmented dataset. The positive effect seems to plateau after 4x augmentation.

BC Regularization: Other offline RL algorithms were initially explored before choosing TD3+BC, but seemed unable to tackle this problem. TD3+BC outperformed them all, including plain BC. In order to investigate the impact of the α regularization term, we used the 8x augmentation dataset and varied the value of α. The results, shown in Fig 3a (right), indicate that larger values of α (i.e. decreasing impact of BC) lead to better results.

The best mean RMSE across tests was achieved with the 8x augmentation and $\alpha = 3.0$, for an average RMSE $= 0.0219 \pm 0.0086$ m, outperforming the baseline which in turn had an average RMSE $= 0.0393 \pm 0.0343$ m. This error difference is mostly due to the successful inversion of the DLO curvature for the initial shapes, shown in Fig. 3b.

V. Concluding Remarks

We have presented preliminary results on the effectiveness of offline RL for a shape control problem exhibiting complex dynamics. We proposed data collection and augmentation procedures which enable learning directly from real data. We validated our procedures on a real-world experiment and compared it with a baseline shape servoing method. We also investigated the impact of BC regularization on the TD3+BC algorithm, and concluded that by decreasing its relative weight, better results can be achieved.

A particularly important result, was the ability of the offline RL method to learn long-term effects, demonstrated by the **successful inversion of the DLOs curvature**, instead of settling at a local minimum as the baseline. While these results are encouraging, more work needs to be done.

REFERENCES

[1] D. Berenson, "Manipulation of deformable objects without modeling and simulating deformation," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4525–4532, IEEE, 2013.

[2] M. Aranda, J. A. Corrales Ramon, Y. Mezouar, A. Bartoli, and E. Özgür, "Monocular visual shape tracking and servoing for isometrically deforming objects," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7542–7549, 2020.

[3] M. Shetab-Bushehri, M. Aranda, Y. Mezouar, and E. Özgür, "As-rigid-as-possible shape servoing," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3898–3905, 2022.

[4] M. Shetab-Bushehri, M. Aranda, Y. Mezouar, and E. Ozgur, "Lattice-based shape tracking and servoing of elastic objects," *arXiv preprint arXiv:2209.01832*, 2022.

[5] J. Matas, S. James, and A. J. Davison, "Sim-to-real reinforcement learning for deformable object manipulation," in *Conference on Robot Learning*, pp. 734–743, 2018.

[6] Z. Hu, T. Han, P. Sun, J. Pan, and D. Manocha, "3-D deformable object manipulation using deep neural networks," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4255–4261, 2019.

[7] C. Shin, P. W. Ferguson, S. A. Pedram, J. Ma, E. P. Dutson, and J. Rosen, "Autonomous tissue manipulation via surgical robot using learning based model predictive control," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 3875–3881, 2019.

[8] R. Jangir, G. Alenya, and C. Torras, "Dynamic cloth manipulation with deep reinforcement learning," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4630–4636, IEEE, 2020.

[9] B. Thach, B. Y. Cho, A. Kuntz, and T. Hermans, "Learning visual shape control of novel 3D deformable objects from partial-view point clouds," *arXiv preprint arXiv:2110.04685*, 2021.

[10] R. Laezza and Y. Karayiannidis, "Learning shape control of elastoplastic deformable linear objects," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4438–4444, 2021.

[11] J. Sanchez, J.-A. Corrales, B.-C. Bouzgarrou, and Y. Mezouar, "Robotic manipulation and sensing of deformable objects in domestic and industrial applications: a survey," *The International Journal of Robotics Research*, vol. 37, no. 7, pp. 688–716, 2018.

[12] S. Fujimoto and S. S. Gu, "A minimalist approach to offline reinforcement learning," *Advances in neural information processing systems*, vol. 34, pp. 20132–20145, 2021.

[13] P. Sundaresan, J. Grannen, B. Thananjeyan, A. Balakrishna, M. Laskey, K. Stone, J. E. Gonzalez, and K. Goldberg, "Learning rope manipulation policies using dense object descriptors trained on synthetic depth data," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9411–9418, IEEE, 2020.

[14] M. Yan, Y. Zhu, N. Jin, and J. Bohg, "Self-supervised learning of state estimation for manipulating deformable linear objects," *IEEE robotics and automation letters*, vol. 5, no. 2, pp. 2372–2379, 2020.

[15] S. Huo, A. Duan, C. Li, P. Zhou, W. Ma, and D. Navarro-Alarcon, "Keypoint-based bimanual shaping of deformable linear objects under environmental constraints using hierarchical action planning," *arXiv preprint arXiv:2110.08962*, 2021.

[16] S. Fujimoto, H. Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *International conference on machine learning*, pp. 1587–1596, PMLR, 2018.

[17] S. Levine, A. Kumar, G. Tucker, and J. Fu, "Offline reinforcement learning: Tutorial, review, and perspectives on open problems," *arXiv preprint arXiv:2005.01643*, 2020.

[18] M. Andrychowicz, F. Wolski, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, O. Pieter Abbeel, and W. Zaremba, "Hindsight experience replay," *Advances in neural information processing systems*, vol. 30, 2017.

[19] A. Kumar, A. Singh, S. Tian, C. Finn, and S. Levine, "A workflow for offline model-free robotic reinforcement learning," *arXiv preprint arXiv:2109.10813*, 2021.