arXiv:2312.05056v1 [cs.RO] 8 Dec 2023

Robotic Control of the Deformation of Soft Linear Objects Using Deep Reinforcement Learning

Mélodie Hani Daniel Zakaria[1], Miguel Aranda[2], Laurent Lequièvre[1], Sébastien Lengagne[1], Juan Antonio Corrales Ramón[3] and Youcef Mezouar[1]

***Abstract*— This paper proposes a new control framework for manipulating soft objects. A Deep Reinforcement Learning (DRL) approach is used to make the shape of a deformable object reach a set of desired points by controlling a robotic arm which manipulates it. Our framework is more easily generalizable than existing ones: it can work directly with different initial and desired final shapes without need for relearning. We achieve this by using learning parallelization, i.e., executing multiple agents in parallel on various environment instances. We focus our study on deformable linear objects. These objects are interesting in industrial and agricultural domains, yet their manipulation with robots, especially in 3D workspaces, remains challenging. We simulate the entire environment, i.e., the soft object and the robot, for the training and the testing using PyBullet and OpenAI Gym. We use a combination of state-of-the-art DRL techniques, the main ingredient being a training approach for the learning agent (i.e., the robot) based on Deep Deterministic Policy Gradient (DDPG). Our simulation results support the usefulness and enhanced generality of the proposed approach.**

I. Introduction

The manipulation of deformable objects is currently a relevant topic in robotics research [1], [2]. In particular, the manipulation of Deformable Linear Objects (DLOs) has high relevance in automation applications: examples of interesting tasks that have been addressed include cable harnessing [3], [4], USB wire soldering [5], or vegetable plant manipulation [6]. One possible perspective on this problem is to study model-based manipulation planning, as done in [7], [8] for elastic rods. In this paper, we are instead interested in the online control of the robot to deform a DLO in a desired way in conditions of high uncertainty and with no knowledge of the object's mechanical deformation model. The works that addressed a similar scenario considered mostly 2D workspaces [3], [9], [10], while control in 3D is significantly more challenging due to the higher complexity of object modeling and perception. Some works addressed control in 3D for small deformations [4], [11]. Overall, while *classical* methods have achieved important progress in this field, the existing challenges motivate us to explore a solution based on Deep Reinforcement Learning (DRL) [12].

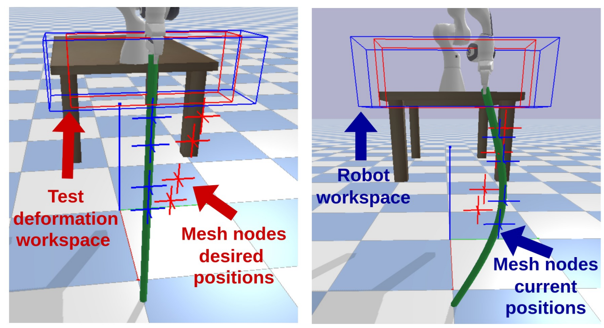

Fig. 1. The setup we consider, including an illustration of some elements of our solution. The robot deforms the soft linear object (green) by making the selected mesh nodes (i.e., the blue points) reach the desired corresponding positions (i.e., the red points). The points are marked as crosses. The robot tip position has to remain within the deformation workspace (i.e., the red box) for performing the desired deformation. The deformation workspace used in testing is bigger than the training workspace. The blue box delimits the robot's workspace, i.e., the robot's gripper tip cannot reach a position out of that box due to the robot's articular limits.

The robotics community has increasingly adopted the usage of DRL algorithms to control robots [13]. Most of these works involve working with rigid bodies with no or negligible deformations [1], [14]. However, soft object manipulation has many crucial applications, especially in household robotic assistance, medicine, and industry [14], [15]. In industrial automation, DRL has already been identified as interesting in tasks with high modeling uncertainty and need for high dexterity. For instance [16], [17] used reinforcement learning for industrial assembly, albeit without having to deal with deformable objects as we do here.

In the literature, the works based on DRL for manipulating deformable objects are, on the one hand, only formulated for simple tasks [1] such as hanging a cloth [14], [15] or moving a rope [12]. On the other hand, most of the soft objects used are 2D [1]: the mesh used to model the object is a 2D mesh, i.e., formed by 2D polygons such as triangles. Promoting progress in this regard, SoftGym [18] presented a set of benchmarks for manipulating soft objects (including 3D objects) using OpenAI Gym [19] and Python interface.

The main drawback of the existing techniques, whether used in simulation [12], [15] or in real experiments [14], is that they are not easily generalizable [1], [12], [13]. Their agent is trained to perform a manipulation from constant

[1]CNRS, Clermont Auvergne INP, Institut Pascal, Université Clermont Auvergne, Clermont-Ferrand, France. [2]Instituto de Investigación en Ingeniería de Aragón, Universidad de Zaragoza, Zaragoza, Spain. [3]Centro Singular de Investigación en Tecnoloxías Intelixentes (CiTIUS), Universidade de Santiago de Compostela, Santiago de Compostela, Spain. Corresponding author: Mélodie Hani Daniel Zakaria, e-mail: Melodie.HANI_DANIEL_ZAKARIA@uca.fr.

This work is funded by AURA through the ATTRIHUM project and from the EU Horizon 2020 research and innovation programme under grant agreement No 869855 (Project 'SoftManBot'). MA is funded via project PGC2018-098719-B-I00 (MCIU/AEI/FEDER, UE), and by the Spanish Ministry of Universities and the European Union-NextGenerationEU via a María Zambrano fellowship. JACR is funded by the Spanish Ministry of Universities through a 'Beatriz Galindo' fellowship (Ref. BG20/00143) and by the Spanish Ministry of Science and Innovation through the research project PID2020-119367RB-I00.

initial to constant target deformations, and it is not trained to deal with different configurations. As an example in DLO manipulation, in [12] the authors control the object shape from some initial states to some desired deformations that are not changeable.

This paper describes a new framework for the robotic control of the shape of DLOs. We use a combination of state-of-the-art DRL algorithms and techniques to build up our framework. We use learning parallelization to make our framework generalizable, i.e., we execute multiple agents in parallel on various environment instances. We focus our study on deformable linear objects. The contributions of our framework are:

1) Its generalizability, i.e., we train the agent only once (using a specific soft object), and it can deform the soft object starting from a different initial position and end up with a different desired shape. Moreover, the agent can make the soft object reach an untrained position, i.e., we train the agent on a small workspace and test it on a bigger one.
2) The complexity of the accomplished task. As shown in Figure 1, the robot deforms a foam bar by making some selected mesh nodes reach the correspondent desired positions in 3D space, potentially involving complex torsion motions. This is made possible by modeling the object with a 3D tetrahedral mesh and via our DRL system design.

We train and evaluate our approach in simulation. Our evaluation is carried out in diverse conditions and it validates the capability of the proposed approach.

II. Problem statement

We address the problem of controlling the deformation of a DLO using a robot arm that manipulates it. For simplicity, the robot grasps one end of the object, and the other end is fixed to the ground. The object is represented by a mesh and we describe its deformation by a set of selected mesh nodes. The objective is to control the arm so that the positions of the selected nodes are driven to prescribed values. The difficulty of this indirect control problem lies in the fact that the dynamical model of the system to be controlled is complex and uncertain. We propose a generalizable architecture to solve this problem based on DRL. The problem setup is illustrated in Figure 1 and our solution will be detailed in the remainder of the paper.

III. Background on Soft Object Manipulation using DRL

This section gives background on the problem of soft object manipulation using DRL. We will focus on discussing aspects that are particularly important for our application.

A. Representing deformable object shape

The most widely used technique is to represent the soft object shape through images [15] instead of modeling it since it is challenging to have a precise model [1]. In [14], a Neural Network detects the soft object shape thanks to supervised learning. The disadvantage of using images is that the computational cost increases, and it is hard to learn afterward (i.e., via DRL) because the resulting state-space is large [15]. In [12], a method based on geometry calculations is proposed to represent the object shape. Another method is based on selecting some mesh nodes in the object model describing the deformation and using their positions as state-space inputs [1], [15]. We preferred to use the latter technique because it is easier to set up, and it keeps the size of the space-state relatively small, which facilitates the training.

B. Techniques to deal with the manipulation complexity

The most common technique in the state-of-the-art is to combine imitation learning with reinforcement learning [1]. Imitation learning is used to reduce the complexity of the manipulation by using demonstrations given by an expert. Another method that we mentioned previously is to have a detailed perception of the object's shape through images [15]. The drawback of both methods is that they have a high computational cost and a large state-space [12], [15]. We prefer to use only a DRL algorithm and select a few mesh nodes that describe the deformation of the object as input to the state-space. This way, our state-space is small, which makes learning easier.

C. Physics-based simulator

Usually, the training of the agent is done in simulation, using a physics-based simulation engine [13]. OpenAI Gym [19] defines an architecture with the main components needed to train the agent, such as resetting the environment, making an action, getting an observation of the state of the environment, and computing the reward. The environment created on the simulator has to have such components. The most popular simulators for deformable object manipulation in the robotics community are MuJoCo [20] and Bullet [21]. We prefer to use PyBullet, the Python interface of Bullet, because it is powerful and open-source.

IV. Components of our DRL framework

In this section, we present standard components of DRL systems that we use in our framework. We focus on explaining how we incorporate them in the framework. The elements include the RL procedure, the Bellman equation, the DDPG algorithm, and the reward function.

A. RL procedure

We consider a classical trial-and-error RL procedure consisting of an agent (e.g., robot) interacting with the environment (e.g., the soft object) based on the policy to maximize rewards on discrete timesteps [22]. In each transition t, the agent starts from the state s_t, and takes an action a_t, which changes the state to a next state s'_t [23]. The state s_t and the action a_t are included in the continuous state space $\mathcal{S}$, and the continuous action space $\mathcal{A}$, respectively, i.e., $s_t \in \mathcal{S}$ and $a_t \in \mathcal{A}$.

The observation the agent got from the environment describes the changes that happened by moving from state s_t to s'_t. The reward r_t evaluates the action taken a_t according