2021 IEEE International Conference on Robotics and Automation (ICRA 2021)
May 31 - June 4, 2021, Xi'an, China

2021 IEEE International Conference on Robotics and Automation (ICRA) | 978-1-7281-9077-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICRA48506.2021.9561766

ReForm: A Robot Learning Sandbox for Deformable Linear Object Manipulation

Rita Laezza[1*], Robert Gieselmann[2*], Florian T. Pokorny[2] and Yiannis Karayiannidis[1]

***Abstract*— Recent advances in machine learning have triggered an enormous interest in using learning-based approaches for robot control and object manipulation. While the majority of existing algorithms are evaluated under the assumption that the involved bodies are rigid, a large number of practical applications contain deformable objects. In this work we focus on Deformable Linear Objects (DLOs) which can be used to model cables, tubes or wires. They are present in many applications such as manufacturing, agriculture and medicine. New methods in robotic manipulation research are often demonstrated in custom environments impeding reproducibility and comparisons of algorithms. We introduce ReForm, a simulation sandbox and a tool for benchmarking manipulation of DLOs. We offer six distinct environments representing important characteristics of deformable objects such as elasticity, plasticity or self-collisions and occlusions. A modular framework is used, enabling design parameters such as the end-effector degrees of freedom, reward function and type of observation. ReForm is a novel robot learning sandbox with which we intend to facilitate testing and reproducibility in manipulation research for DLOs.**

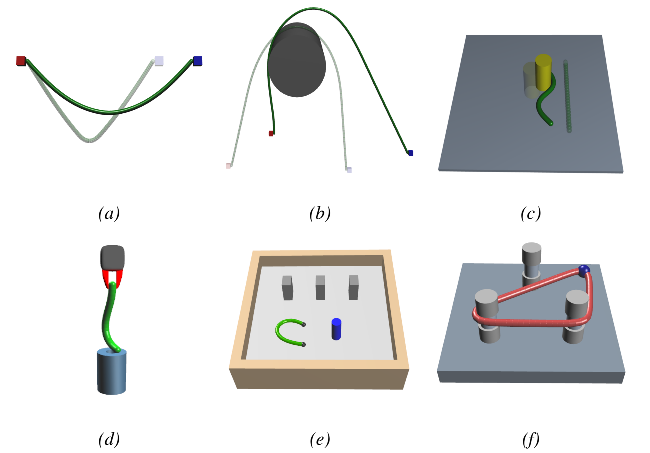


Fig. 1: Overview of environments included in ReForm: (a) BendWire (b) BendWireObstacle (c) PushRope (d) PegInHole (e) CableClosing (f) RubberBand. The first three are explicit shape control tasks, while the last three are implicit shape control tasks.

I. INTRODUCTION

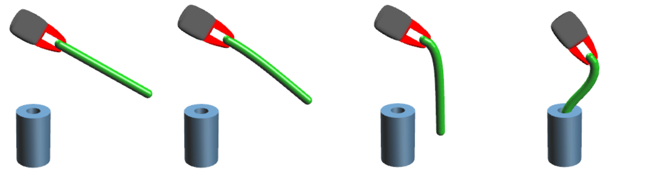
Countless manufacturing and every-day tasks require handling of non-rigid objects. Thus, it is important that they are properly studied, in all their variability. Yet, the dynamics of deformable objects are complex and inherently difficult to model and simulate [1]. This makes robotic manipulation of such objects using *learning-free* control a challenge. For this reason machine learning, in particular Reinforcement Learning (RL), has become increasingly popular for solving robotic manipulation tasks [2], [3]. Despite their success, RL methods are notoriously unstable and results hard to reproduce [4]. Furthermore, the complexity of the robotics system and the learning algorithm together, make it difficult to evaluate each component independently. There have been several efforts to create robotics benchmarks in order to facilitate comparisons between methods [5]–[13].

Nevertheless, current simulation benchmarks such as RL-Bench [7] or Meta-World [9] focus almost exclusively on the manipulation of rigid objects. Due to the large number of industrial applications, it is necessary to also address the challenges of deformable objects. Simulation environments are particularly important for learning algorithms, since they require large amounts of data which is costly to obtain from real systems [5]. Further, real-world experiments involving deformable parts are challenging due to effects such as irreversiblity of deformations and self-occlusion.

There are few works that sufficiently address deformable objects. Based on this observation, we developed ReForm, a novel simulation sandbox and benchmarking tool for deformable object manipulation. In particular, ReForm focuses on Deformable Linear Objects (DLOs) with an emphasis on the variation of mechanical properties, such as low compression strength (e.g. rope), elastic, plastic and elastoplastic behaviors. The motivation to focus on DLO's stems from the numerous manipulation tasks which are found across industries, such as manufacturing, surgery and agriculture [14]. Concurrent to our work, [15] recently presented SoftGym, a tool to benchmark soft object manipulation in simulation. Despite its focus on deformability, they employ a particle-based simulator which does not accurately model important material properties such as stiffness or plasticity.

An important aspect of ReForm is the freedom given with respect to simulation and problem settings including the type of observation, actuation, reward or material. For instance, one could modify the stiffness of an object in a peg-in-hole task (see Figure 2). This enables users to quickly set up new experiments with custom parameters. ReForm consists of six core DLO manipulation tasks which are integrated using the popular OpenAI Gym [10] framework. Moreover, we provide a modular interface to allow the creation of entirely new manipulation tasks. In all environments, Cartesian manipulators are employed that provide a continuous control setting. In this regard, the active Degrees of Freedom (DOFs)

* Equal contribution.
[1] Division of Systems and Control, Department of Electrical Engineering, Chalmers University of Technology, Sweden {laezza , yiannis}@chalmers.se
[2] Division of of Robotics, Perception and Learning, Department of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, Sweden {robgie , fpokorny}@kth.se



(a) (b) (c) (d)

Fig. 2: Illustration of a peg-in-hole task for pegs with different stiffness values: (a) rigid, (b) flexible and (c) soft. Image (d) shows an example of the object being inserted.

can be modified by the user. To demonstrate the influence of the type of observation, controlled DOFs and rewards, we evaluate standard model-free reinforcement learning for a subset of the manipulation tasks.

II. Related Work

Compared to rigid objects, shape in addition to pose estimation is required to fully capture the state of a deformable object. While in some applications deformation is treated as a disturbance, in many others achieving a particular shape or deformation is the main objective [1]. The most commonly studied problems are textile flattening or folding, knot tying and shape control of ropes or cables [14], [16]–[19].

Matas et al, [17] studied an end-to-end reinforcement learning solution for cloth manipulation. Their method learns to predict torque signals directly from images in simulation. Using sim-to-real techniques they demonstrated that the learned policy can be transferred to the real world. Note that their method uses pre-training based on expert demonstrations. The work by [20] studies dual-arm manipulation of flexible cables. [21] applies deep reinforcement learning for a peg-in-hole task where the insertion is made of foam. In [22], the authors train an agent to insert a soft cable into a hole. While they demonstrated great results in simulation and on a real robot, their method also relies on expert demonstrations.

The variety of implementations of algorithms and experiments makes comparisons between methods and results difficult. The development of benchmarks is motivated by this need for reproducibility and comparability [4], [5]. Despite their success, deep learning methods are inherently difficult to compare due to stochasticity in the optimization process, dependence on initial parameters and random seeds. Moreover, deep reinforcement learning is known to be data inefficient and often takes millions of interactions until convergence. It also requires laborious hyperparameter tuning which hampers the ability to discern true algorithmic improvements from the amount of hand-tuning performed. [4] covered these and more issues of RL, focusing on policy gradient methods for continuous control tasks. In this regard, simulation benchmarks are particularly interesting as they provide constant conditions and allow to iterate quicker over different methods.

RLBench [7] is a recent simulation benchmark which includes 100 manipulation tasks. Unfortunately, only one of those, namely the rope-straightening task, involves a deformable object. Concept2Robot [8], which is a framework for learning manipulation concepts from human visual demonstrations and language instructions, includes the task of *folding something*. However, the simulation appears to be overly simplified[1]. Again, only one of 74 tasks addresses the challenges of deformable objects. The SURREAL [5] framework provides a smaller robotics benchmark, with six classical manipulation tasks, such as peg-in-hole, but none including non-rigid objects. This suite was also used in the RoboTurk [6] crowd-sourcing platform for imitation learning. Meta-World [9] is yet another benchmark which focuses on multi-task learning in 50 scenarios but also excludes deformation from its simulations. SAPIEN [13] is a household simulation benchmark which allows manipulation of articulated objects, e.g doors. iGibson [12] is a similar benchmark that allows interactions with the environment. Still, it mainly focuses on indoor navigation. AssistiveGym [11] provides six benchmark tasks for human-robot interaction, one of which involves dressing a human with a deformable textile.

SoftGym [15] is a recently proposed open-source benchmark for manipulating deformable objects in simulation. It contains control tasks for cloth-like objects, ropes and a fluid. At the core, the simulator models deformable bodies using a particle-based system. In contrast, our framework focuses on DLOs and uses specialized object classes to model realistic material properties such as stiffness, elasticity and plasticity. Elastica [23] is an open-source simulation environment for soft, slender rods. It was designed to simulate soft robotic actuators that can bend, twist, shear and stretch. Unlike Elastica, our system addresses control of DLOs from a general object manipulation perspective. Thus, we provide more flexibility with respect to tasks and material properties.

Henderson et al. [4] argues that the choice of environment plays an important role when validating a new RL algorithm, because typically no single algorithm outperforms the others across all tasks. At present, deformable objects are underrepresented in robot learning benchmarks. For that reason, it is unclear how well state-of-the-art RL methods cope with the challenges inherent to the manipulation of non-rigid bodies.

III. ReForm

With the introduction of ReForm[2], we provide tools to experiment with DLOs in simulation and train agents for the manipulation of cables, ropes and wires in six different tasks. Our intention is to provide a robot learning sandbox to benchmark new methods and foster research on deformable object manipulation in simulation. We categorize different tasks as either *explicit shape control* problems, where the goal is to deform the object into a specific geometric configuration, or *implicit shape control* problems. For the latter, the exact shape of the object is not the primary objective, instead a set of high-level conditions must be fulfilled to solve the task. A few examples include hot-wire cutting [24], tube mounting

[1]Task 14 of Figure 3 in [8], shows two rigid bodies attached through a hinge constraint. Code is currently not available.

[2]https://sites.google.com/view/reformdlo/home

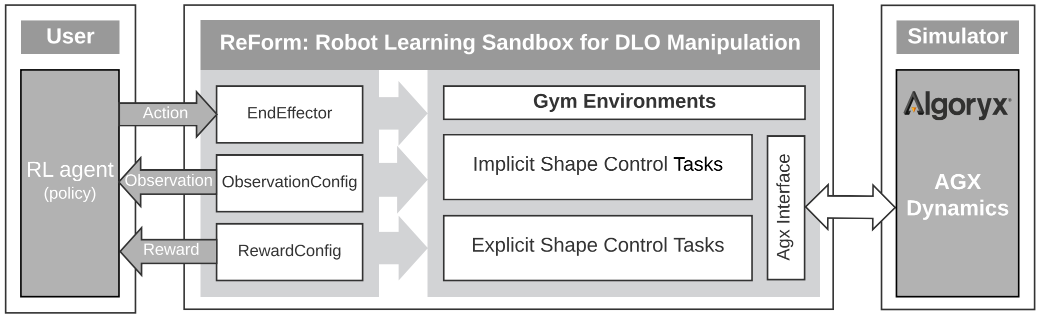


Fig. 3: System overview of ReForm sandbox for robot learning. It provides a modular design to make benchmarking of new deformable object manipulation strategies easy. Note that the RL agent can be replaced by an arbitrary control policy.

[25], knot tying and threading [14], etc. Both implicit and explicit shape control problems are addressed in this work.

Similar to previous robotics benchmark software, we provide a modular implementation, as shown in Figure 3. This is done by providing an OpenAI Gym [10] interface, which is a well established toolbox for RL research. ReForm is also designed to allow configuration freedom, making it easy to modify and extend. Furthermore, the benchmark provides over ten observation types, including ground-truth and image data. Agent actions are limited to task space control, defined as linear and angular velocities of the end-effector. These can also be defined in a modular fashion, as a set of motor constraints. For environments where the gripper is holding the DLO, it is possible to control grip compliance.

In the following, we present the main components of ReForm, starting with the multiphysics simulator, followed by the description of available observation types, the control interface and finally the reward interface.

a) Simulation Environment: Modelling and simulating DLOs or deformable objects, in general, is inherently difficult due to the complex mechanics. AGX Dynamics[3] provides unified lumped element models with implicit integration and direct, sparse factorization. Specifically, it offers a `Cable` class for which properties, such as Young modulus and Poisson's ratio, can be defined along stretch, twist and bend directions [26], [27]. Further, elastic `Cable` objects can be assigned plasticity properties by defining a yield point. Figure 4a shows the impact of this mechanical property on a shape control problem. Additionally, it offers specialized material classes which capture elastoplastic deformation in real-time. For these reasons, it provides an advantage over other physics simulators e.g. Mujoco [28], Bullet and SOFA [29]. While MuJoCo enables simulations of long object chains, it uses an explicit solver and cannot provide high stiffness. SOFA, on the other hand, offers only an iterative solver and was initially developed for interactive computational medical simulation.

Photorealism is another important aspect to be considered, specially for end-to-end approaches which aim to learn directly from images. Currently, ReForm uses OpenSceneGraph for rendering, but AGX Dynamics also supports more realistic game engines, e.g. Unity and Unreal. It is left as future work to make use of these capabilities.

[3]https://www.algoryx.se/agx-dynamics/

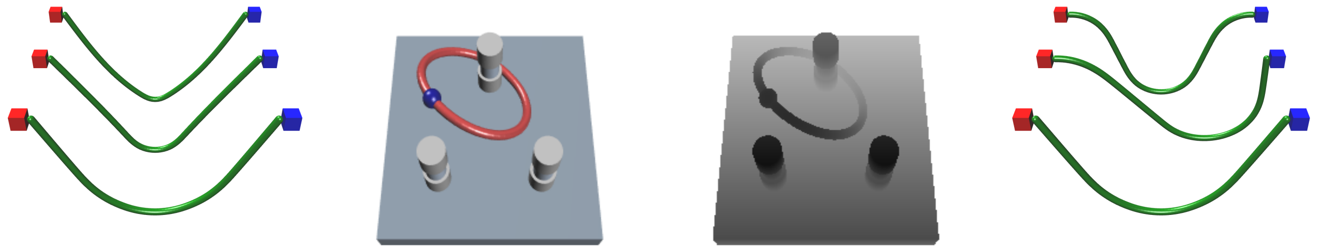
```
observation_config = ObservationConfig(
    goals=[ObservationConfig.ObservationType.
    DLO_POSITIONS]
    )
observation_config.set_img_rgb()
```

Listing 1: Example of an observation object. In this case, the observations consist of RGB image, while the goal is defined based on Cartesian coordinates of discretized DLO.

b) State Representation: Robust state estimation is an open problem in deformable object manipulation research [30], [31]. In ReForm, both visual observations such as RGB/depth (see Figure 4) and force/torque measurements are supported. Using the `ObservationConfig` class (see Listing 1) it is even possible to create composite input types.

For evaluating the success of implemented methods, it is also possible to obtain ground truth position and rotation of the segments that constitute the DLO. Finally, for shape control problems, in which the location of the DLO is not necessarily important, intrinsic metrics such as discrete curvature and torsion [32] can be useful state representations.

c) Control Strategy: In order to be platform-independent, we focus on task space control settings only. An `EndEffector` class is provided to define the controlled DOFs, along with velocity and acceleration limits (see Listing 2). Besides velocity control commands, there is also support for grip compliance, which allows for more complex DLO configurations, by varying the resistance to rotation along one axis, as seen in Figure 4d. Our system allows to include velocity and acceleration limits which is important because RL policies often produce jitter. The work in [33] highlights the importance of trajectory speed for dynamic manipulation strategies applied to non-rigid objects. Adding these constraints acts as a filter which prevents fast velocity changes and excessive forces. Note that this has a similar effect to using temporally correlated exploration noise, such as in [34], [35]. As a further benefit, this makes our control interface task agnostic, since agent actions always lie between $[-1, 1]$, but are automatically rescaled to a range appropriate for the task.



(a) (b) (c) (d)

Fig. 4: (a) Illustration of the effect of of plasticity on the shape of a DLO. All three shapes were generated with identical velocity trajectories, the only difference is the value of the yield point of the DLOs. This property relates to the transition from the elastic to the plastic domain. Examples of visual observations available in ReForm: (b) RGB image (c) Depth image. (d) Illustration of grip compliance changes on the shape of DLOs with identical mechanical properties, and identical trajectories.

```
gripper_right = EndEffector(
    name='gripper_right',
    controllable=True,
    observable=True,
    max_velocity=0.014,   # m/s
    max_acceleration=0.010   # m/(s*s)
    )
gripper_right.add_constraint(
    name='prismatic_joint_right',
    end_effector_dof=EndEffectorConstraint.Dof.
    X_TRANSLATION,
    compute_forces_enabled=True,
    velocity_control=True,
    compliance_control=False
    )
gripper_right.add_constraint(
    name='hinge_joint_right',
    end_effector_dof=EndEffectorConstraint.Dof.
    Y_COMPLIANCE,
    compute_forces_enabled=False,
    velocity_control=False,
    compliance_control=False
    )
```

Listing 2: Example of an end-effector object. Both velocity and acceleration limits are set at initialization (SI units). Note that since the simulation consists of a small DLO, it is important to keep velocities low and prevent large accelerations. If the end-effector is set to be `observable`, this enables force-torque measurements that can be used for the observations. Besides `TRANSLATION` constraints, there are `COMPLIANCE` constraints to control the compliance of the grip.

d) Reward Definition: Though ReForm can be used to evaluate methods other than RL, we follow the formalism from OpenAI Gym [10], and thus include a reward computation step. While standard implementations of Gym environments have a fixed reward, effectively making it part of the environment, our library provides an abstract `RewardConfig` class, which enables the reward definition to be part of the solution strategy. We find this to be particularly important since for many deformable object manipulation tasks, the definition of the reward function is not trivial [36]. While sparse rewards are generally applicable, a learning signal is only provided once the goal has been reached. It was shown by [3] that for some multi-goal scenarios this type of reward makes learning nearly impossible. Engineering dense rewards by hand is often a laborious task that might require significant domain expertise. In Section V, we evaluate the impact of the reward definition for a subset of our environments.

IV. Manipulation Tasks

In this section, we describe the tasks that are currently implemented in ReForm. An overview of all environments is given in Figure 1. In general, we distinguish between explicit and implicit shape control. The former describes problems in which the goal is represented by a particular shape configuration of the deformable object. For some problems however it might be more convenient to define the primary task by means of a high-level description. Examples include assembly problems in which the final shape of the object is secondary. These kinds of tasks are captured by implicit shape control. In the following, we briefly describe the features and challenges of each task in ReForm:

BendWire: a thin wire is attached to two grippers using a hinge constraint. The goal is to deform the wire into a desired shape, hence it can be seen as explicit shape control. The material of the wire is stiff and exhibits elastoplastic properties. The fact that plastic deformations are usually hard or even impossible to reverse presents a key challenge. Even small deformations can significantly change the set of reachable wire states. DLOs of this kind include most metal wires, which are found throughout manufacturing as well as in medical applications, such as in dental braces. A similar problem was tackled by [20].

BendWireObstacle: this environment is similar to BendWire but includes a cylindrical obstacle in the workspace. While not necessary, the obstacle can be leveraged to facilitate the deformation task. With this setting, we open the possibility for extrinsic manipulation. The work by Zhu et al. [37] covered such a scenario for cable routing. This task is also an explicit shape control problem.

PushRope: a soft rope is located on a planar surface. A controllable pusher is used to bring the rope into a certain goal configuration. The rope exhibits little stiffness and deforms immediately after contact. Further, the friction between the rope and surface adds another interesting feature to the manipulation task. PushRope is a common explicit shape control benchmark which has been studied in [7], [36].

PegInHole: a soft peg is on one end rigidly attached to a gripper. The task is to insert the peg into a hollow cylindrical object. Compared to the previous environments, the goal is not represented by just a single configuration

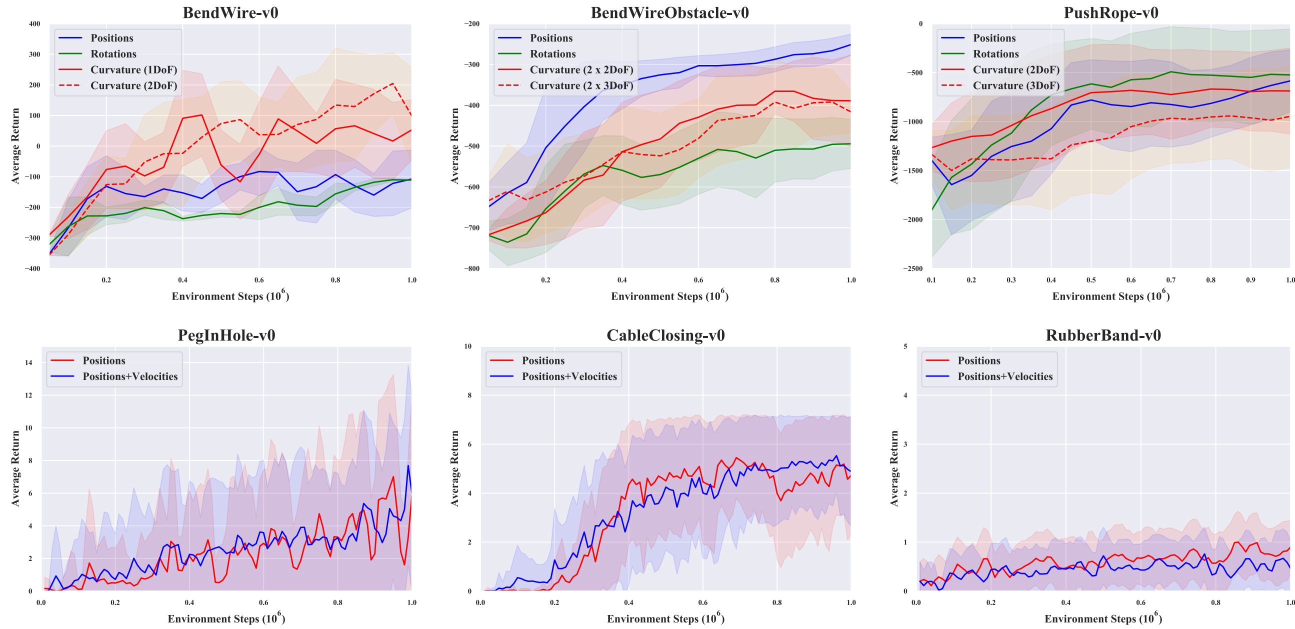


Fig. 5: Top row shows average return results for the three explicit shape control environments: BendWire, BendWireObstacle and PushRope. Results are averaged over 5 runs, shaded regions represent mean standard deviation during training. Results indicate the effect of different observation types and larger action spaces on learning. Bottom row shows average return results for the three implicit shape control environments: PegInHole, CableClosing and RubberBand. Results are presented for observations with just the DLO positions and with both positions and velocities. A dense reward definition was used; only one trial is shown.

of the object. Instead it is a set of configurations which satisfy the condition of being inserted. Clearly, this is an implicit shape control problem. Due to its simplicity and significance for assembly tasks, the peg-in-hole scenario has become a widely-studied problem in robotics research. We offer a simulated version that uses a deformable peg with a rigid hole. Note that [21] presented the converse problem in which a rigid peg is inserted into a deformable hole.

CableClosing: a cable is controlled on both ends by planar actuators. The task is successfully completed once the cable fully encloses a goal obstacle. There are several static obstacles in the scene. In order to reach the goal, the object has to be circumnavigated around the obstacles. Again, this describes an implicit shape control problem. This task is quite unique, although it can be compared with the work by [14], where they used objects in the environment to tie different types of knots.

RubberBand: a purely elastic circular rubber band is connected to a gripper by a ball joint. The task is to find a policy which wraps the rubber band around three poles, making it an implicit shape control task. This environment incorporates complicated contact mechanics between the deformable object and the poles. Further, it describes a wrapping task which bridges the gap to industrial applications, such as [38]. Though a rubber band is technically not a linear object, it can be seen as a DLO connected with itself.

V. Benchmarking Experiments

In order to establish a first baseline, we performed an evaluation of all six environments using model-free reinforcement learning. For each, we trained and evaluated DDPG agents [35] as implemented in [39]. We used two hidden layers for both policy and value networks, with $[300, 400]$ neurons each. Due to the different nature of explicit and implicit shape control tasks, we evaluate them separately. For each task, the DDPG algorithm was applied using $1{\times}10^6$ steps of environment interaction.

a) Explicit Shape Control: For these environments, the goal is to control the end-effector(s) in order to deform a DLO into a desired shape. Each task has different challenges including plasticity, interaction with a rigid body and low compression strength. In order to evaluate the impact of design choices such as observation type, action space and reward definition, we run experiments varying these parameters. Figure 5 (top row) shows the effect of three different observation types: ground truth positions, rotations and curvature. No observation type seems to work best across all tasks. It is curious that for the wire environments, there is a clear winner but they do not coincide, with curvature working better for BendWire and DLO segment positions for BendWireObstacle. Results for the PushRope task were quite close and suffered from high variance, making it difficult to draw conclusions.

Figure 5 (top row) further shows the impact on augmenting the action space. For the wire tasks, this was done by adding control of grip compliance, which can lead to very different shapes (see Figure 4d). For the PushRope task, a third DOF was added, effectively allowing the pusher to move over the rope, thus making exploration much harder. As expected, increasing the action space led to slower learning for this task. However, for the wire tasks it is not easy to draw a clear

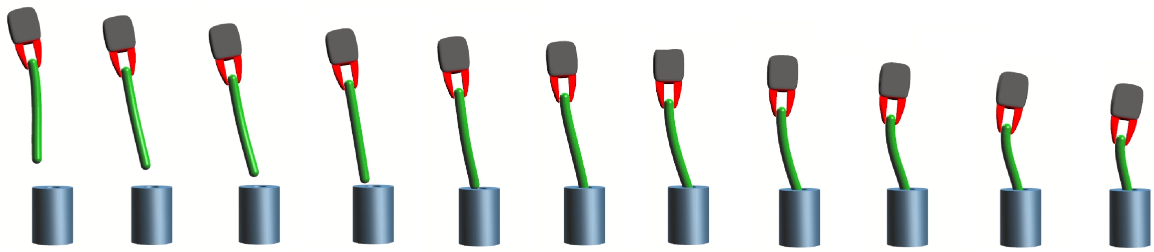


Fig. 6: Example of successful trajectory for the PegInHole task learned by the DDPG agent.

conclusion. For the BendWire environment it even seems to help performance. This may be because the goal shape consists of a sharp angle (see Figure 1a), making the extra DOF useful for creating sharper deformations.

For both the observation and action experiments, the same dense reward was used, computed as negative the L2 distance between the current curvature and the desired curvature. Further, a small positive reward is given when the agent is close to the target shape. However, to analyse the impact of our reward definition, we also implemented a sparse reward which provides a negative signal until the goal is reached, after which a positive signal is given. For each environment, the final curvature error for both sparse and dense rewards were compared. Results are nearly identical across all tasks, with 0.13 $\pm$0.07 (dense) and 0.13$\pm$0.20 (sparse) for BendWire; 0.23 $\pm$0.08 (dense) and 0.24$\pm$0.10 (sparse) for BendWireObstacle; and 0.14 $\pm$0.03 (dense) and 0.15$\pm$0.01 (sparse) for PushRope.

b) Implicit Shape Control: For all of the implicit shape control tasks, the state observation consists of the position of the DLO segments and the grippers. We further evaluated a second setting which also takes into account the velocity of the involved bodies. Note that the definition of a continuous reward function is not obvious for these tasks due the presence of obstacles in the workspace. Instead, we provided intermediate feedback for reaching task-specific subgoals. In the PegInHole task, the agent receives a positive reward of +1 for each cable segment being inserted. In the CableClosing environment, the agent receives a positive reward of +1 for partially enclosing the pole and an additional reward of +1 for fully enclosing it. In the RubberBand task, the agent receives a positive reward of +1 for each pole being enclosed by the rubber band and an additional reward of +5 at task completion. Reversing the progress is penalized accordingly using negative rewards.

The corresponding training returns are shown in Figure 5 (bottom row). It can be seen that the policies converge in the PegInHole and CableClosing environments. Figure 6 shows a trajectory generated by the learned policy for the PegInHole insertion task. Yet, we could not achieve statisfying results in the RubberBand environment. The DDPG standard implementation applies Gaussian noise to the predicted actions in order to explore the state-action space. We believe that this strategy is insufficient to reach high-reward regions in the RubberBand case and suggest the use of more sophisticated exploration methods. Furthermore, the wobbly dynamics present an additional challenge of this control task. Note that we did not observe significant improvements when including velocities into the observations. The increased dimensionality of the input might have overshadowed the benefits of using this additional type of information. After all, the issue of dealing with high-dimensional state-spaces is inherent to the manipulation of deformable objects.

VI. Conclusion

In this work we introduced ReForm, a new sandbox for robot learning including six DLO manipulation environments. The development of our framework is motivated by the need for more benchmarks which capture the complexities of deformable objects. We evaluated the performance of DDPG agents for all environments using different types of observations, action spaces and reward functions. The results present a first baseline for future investigations.

During the Robotics Debates[4] workshop at ICRA 2020, the issue of whether “Robotics research is over-reliant on benchmark datasets and simulation” was debated. Though there has been no final consensus about the need for simulation benchmarks, several arguments in favor of it are i. the ability to replicate results; ii. the idea that “simulators democratize robotics”, since researchers with limited resources can still test their methods; iii. in related fields such as computer vision and natural language understanding, benchmark datasets have long been an important part of these communities.

In future work, we seek to extend ReForm by adding new manipulation tasks. Moreover, photorealistic rendering and procedural generation abilities will enable our framework for domain-adaptation and sim-to-real research.

Acknowledgements

This work was supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. Simulation software was provided by Algoryx AB. The computations were enabled by resources provided by the Swedish National Infrastructure for Computing (SNIC) at C3SE partially funded by the Swedish Research Council through grant agreement no. 2018-05973.

[4]https://roboticsdebates.org/

REFERENCES

[1] 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.

[2] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1238–1274, 2013.

[3] M. Andrychowicz, F. Wolski, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, O. P. Abbeel, and W. Zaremba, "Hindsight experience replay," in *Advances in neural information processing systems*, pp. 5048–5058, 2017.

[4] P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup, and D. Meger, "Deep reinforcement learning that matters," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[5] L. Fan, Y. Zhu, J. Zhu, Z. Liu, O. Zeng, A. Gupta, J. Creus-Costa, S. Savarese, and L. Fei-Fei, "Surreal: Open-source reinforcement learning framework and robot manipulation benchmark," in *Conference on Robot Learning*, pp. 767–782, 2018.

[6] A. Mandlekar, Y. Zhu, A. Garg, J. Booher, M. Spero, A. Tung, J. Gao, J. Emmons, A. Gupta, E. Orbay, S. Savarese, and L. Fei-Fei, "Roboturk: A crowdsourcing platform for robotic skill learning through imitation," in *Conference on Robot Learning*, 2018.

[7] S. James, Z. Ma, D. R. Arrojo, and A. J. Davison, "Rlbench: The robot learning benchmark & learning environment," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3019–3026, 2020.

[8] L. Shao, T. Migimatsu, Q. Zhang, K. Yang, and J. Bohg, "Concept2robot: Learning manipulation concepts from instructions and human demonstrations," *Robotics: Science and Systems*, 2020.

[9] T. Yu, D. Quillen, Z. He, R. Julian, K. Hausman, C. Finn, and S. Levine, "Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning," in *Conference on Robot Learning*, pp. 1094–1100, 2020.

[10] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," 2016.

[11] Z. Erickson, V. Gangaram, A. Kapusta, C. K. Liu, and C. C. Kemp, "Assistive gym: A physics simulation framework for assistive robotics," *IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[12] F. Xia, W. B. Shen, C. Li, P. Kasimbeg, M. E. Tchapmi, A. Toshev, R. Martín-Martín, and S. Savarese, "Interactive gibson benchmark: A benchmark for interactive navigation in cluttered environments," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 713–720, 2020.

[13] F. Xiang, Y. Qin, K. Mo, Y. Xia, H. Zhu, F. Liu, M. Liu, H. Jiang, Y. Yuan, H. Wang, L. Yi, A. X. Chang, L. J. Guibas, and H. Su, "Sapien: A simulated part-based interactive environment," *ArXiv*, vol. abs/2003.08515, 2020.

[14] M. Saha and P. Isto, "Motion planning for robotic manipulation of deformable linear objects," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pp. 2478–2484, IEEE, 2006.

[15] X. Lin, Y. Wang, J. Olkin, and D. Held, "Softgym: Benchmarking deep reinforcement learning for deformable object manipulation," *arXiv preprint arXiv:2011.07215*, 2020.

[16] 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.

[17] 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.

[18] M. Ruan, D. Mc Conachie, and D. Berenson, "Accounting for directional rigidity and constraints in control for manipulation of deformable objects without physical simulation," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 512–519, IEEE, 2018.

[19] Y. Wu, W. Yan, T. Kurutach, L. Pinto, and P. Abbeel, "Learning to manipulate deformable objects without demonstrations," *Robotics: Science and Systems XVI*, Jul 2020.

[20] 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)*, pp. 479–484, IEEE, 2018.

[21] J. Luo, E. Solowjow, C. Wen, J. A. Ojea, and A. M. Agogino, "Deep reinforcement learning for robotic assembly of mixed deformable and rigid objects," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2062–2069, IEEE, 2018.

[22] M. Vecerík, T. Hester, J. Scholz, F. Wang, O. Pietquin, B. Piot, N. M. O. Heess, T. Rothörl, T. Lampe, and M. A. Riedmiller, "Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards," *ArXiv*, vol. abs/1707.08817, 2017.

[23] N. Naughton, J. Sun, A. Tekinalp, G. Chowdhary, and M. Gazzola, "Elastica: A compliant mechanics environment for soft robotic control," 2020.

[24] B. T. Simon Duenser, Roi Poranne and S. Coros, "Robocut: Hot-wire cutting with robot-controlled flexible rods," *ACM Transactions on Graphics (Proc. ACM SIGGRAPH)*, vol. 39, Jul 2020.

[25] M. Rambow, T. Schauß, M. Buss, and S. Hirche, "Autonomous manipulation of deformable objects based on teleoperated demonstrations," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2809–2814, IEEE, 2012.

[26] M. Servin and C. Lacoursière, "Rigid body cable for virtual environments," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 4, pp. 783–796, 2008.

[27] H. Lang, J. Linn, and M. Arnold, "Multi-body dynamics simulation of geometrically exact Cosserat rods," *Multibody System Dynamics*, vol. 25, no. 3, pp. 285–312, 2011.

[28] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033, IEEE, 2012.

[29] F. Faure, C. Duriez, H. Delingette, J. Allard, B. Gilles, S. Marchesseau, H. Talbot, H. Courtecuisse, G. Bousquet, I. Peterlik, *et al.*, "Sofa: A multi-model framework for interactive physical simulation," in *Soft tissue biomechanical modeling for computer assisted surgery*, pp. 283–321, Springer, 2012.

[30] 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*, 2020.

[31] T. Tang and M. Tomizuka, "Track deformable objects from point clouds with structure preserved registration," *The International Journal of Robotics Research*, p. 0278364919841431, 2018.

[32] D. Carroll, E. Hankins, E. Kose, and I. Sterling, "A survey of the differential geometry of discrete curves," *The Mathematical Intelligencer*, vol. 36, no. 4, pp. 28–35, 2014.

[33] R. Jangir, G. Alenya, and C. Torras, "Dynamic cloth manipulation with deep reinforcement learning," *arXiv preprint arXiv:1910.14475*, 2019.

[34] P. Wawrzynski, "Control policy with autocorrelated noise in reinforcement learning for robotics," *International Journal of Machine Learning and Computing*, vol. 5, no. 2, p. 91, 2015.

[35] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[36] X. Lin, H. S. Baweja, and D. Held, "Reinforcement learning without ground-truth state," *arXiv preprint arXiv:1905.07866*, 2019.

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

[38] M. Murase, K. Yamazaki, and T. Matsubara, "Kullback leibler control approach to rubber band manipulation," in *2017 IEEE/SICE International Symposium on System Integration (SII)*, pp. 680–685, IEEE, 2017.

[39] A. Stooke and P. Abbeel, "rlpyt: A research code base for deep reinforcement learning in pytorch," *arXiv preprint arXiv:1909.01500*, 2019.