



**HAL**  
open science

# Whole-arm Grasping Strategy for Soft Arms to Capture Space Debris

Camilla Agabiti, Etienne Ménager, Egidio Falotico

► **To cite this version:**

Camilla Agabiti, Etienne Ménager, Egidio Falotico. Whole-arm Grasping Strategy for Soft Arms to Capture Space Debris. RoboSoft 2023 - 6th IEEE-RAS International Conference on Soft Robotics, Apr 2023, Singapore, Singapore. pp.1-6. hal-04167893

**HAL Id: hal-04167893**

**<https://inria.hal.science/hal-04167893v1>**

Submitted on 24 Jul 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

# Whole-arm Grasping Strategy for Soft Arms to Capture Space Debris\*

Camilla Agabiti<sup>1a</sup>, Etienne Ménager<sup>2a</sup> and Egidio Falotico<sup>1</sup>, *Member, IEEE*

**Abstract**—In this work, we present a whole-arm grasping strategy for soft arms whose task is to capture space debris. The non-cooperative nature of space debris and the characteristics of the space environment enforce high-level requirements for robotic arms, especially dexterity. Taking inspiration from the outstanding capabilities of the elephant trunk in grasping, we formulated a grasping strategy based upon the identification of contact points on the object to force the bending of the arm and induce the wrapping around the object, as the animal model does. This strategy is implemented by leveraging on coupled Finite Element simulations of a trunk-like soft arm and Reinforcement Learning tools to learn the grasping. The results show that the robot successfully learns the task by moving the proximal part closer to the object and using the distal one to wrap around the object. We show that the obtained policy is valid for diverse object sizes and positions. Our grasping strategy is the first example of bio-inspired whole-arm grasping for a soft arm in space. We believe that, in the near future, this strategy will enable new grasping capabilities in soft arms.

## I. INTRODUCTION

Since the beginning of the Space Age, the number of man-made objects in space has been constantly increasing, and active debris removal is now urgent [1]. Among the proposed approaches to remove space debris, the capture performed by robotic manipulators mounted on chaser satellites (S/C) is promising [2]. So far, no such space debris removal mission has been carried out yet. Nevertheless, several studies in the literature propose capture architectures where the debris catching is entrusted to the end-effector of rigid manipulators at selected gripping points [3], [4], [5], [6]. Compared to ground-based applications, robotic manipulators mounted on S/C undergo external actions due to micro-gravity and the space environment. In addition, space debris are “non-cooperative” objects, implicating not only the impossibility of self-controlling their orientation but also the absence of appropriate mechanisms for engagement with the manipulator end-effector. Therefore, the capture of space debris is still facing many challenges, mostly dealing with the design of dexterous robotic manipulators, as well as versatile grasping of objects not designed to be grasped [7].

\*This work was supported by the European Union’s Horizon 2020 FET-Open program under Grant 863212 through the PROBOSCIS Project, by the European Union’s Horizon 2020 FET Proactive program under Grant 863212 through the GROWBOT Project, and by the European Union’s Horizon 2020 FET under Grant 945539 through the Human Brain Project

<sup>1</sup>Camilla Agabiti and Egidio Falotico are with the BioRobotics Institute, Scuola Superiore Sant’Anna, Pontedera 56025, Italy, and also with the Department of Excellence in Robotics and AI, Scuola Superiore Sant’Anna, Pisa 56127, Italy. [camilla.agabiti@santannapisa.it](mailto:camilla.agabiti@santannapisa.it), [egidio.falotico@santannapisa.it](mailto:egidio.falotico@santannapisa.it)

<sup>2</sup>Etienne Ménager is with Université de Lille, Inria, CNRS, Centrale Lille, UMR 9189 CRISTAL, F-59000 Lille, France, [etienne.menager@inria.fr](mailto:etienne.menager@inria.fr)

<sup>a</sup>Camilla Agabiti and Etienne Ménager contributed equally to the work

In this scenario, soft robotics offers powerful intrinsic properties simplifying the grasping of non-cooperative objects. In recent years, continuum and flexible structures existing in nature have inspired researchers to develop new robotic systems endowed with enhanced capabilities due to their soft bodies, such as high deformability, and adaptability to complex and unpredictable environments [8].

In our approach, we take inspiration from the outstanding abilities of the elephant to execute dexterous grasping tasks with its trunk. Indeed, the elephant trunk is an example of a continuum biological structure capable of conforming to the environment through its intrinsic mechanical compliance; it can dexterously grasp and manipulate objects featuring different shapes, sizes, physical states, and weights by relying on its whole-arm adaptation. Compliant whole-arm grasping has gained increasing interest in continuum robots operating in unstructured environments. Just to give one example, in 2022, Zhang et al. presented a continuum and compliant cable-driven robot inspired by the structural characteristics of seahorse tails in grasping [9].

In our work, we present a whole-arm grasping strategy inspired by the elephant trunk for a soft arm, whose task is to capture space debris. To the best of our knowledge, this is the first example that not only a soft arm is proposed for the capture of space debris, but also that a soft arm, by means of its body compliance, implements a bio-inspired whole-arm grasping strategy for this application.

Recently, potentialities of soft robots in space have been surveyed [10]. Promising space applications involve soft robots in on-orbit missions, such as the on-orbit inspection or maintenance tasks and the capture of space debris. However, high-impact advantages of soft robots come also with relevant burdens, since the modeling and control of soft robots are challenging [11], [12]. In soft robotics, a continuum approach is required, complexities from the material and geometrical sides must be considered as well as dynamic effects originating from the interaction with the environment [12]. Nowadays, one of the most accurate modeling tools is the Finite Element Method (FEM), capable of handling large deformations, complex geometries, and material non-linearities, besides being computationally costly [12].

To face the challenge of soft robots control, machine learning approaches have been recently introduced [13]. For example, in 2021, Naughton et al. developed *Elastica*, a simulator based on Cosserat continuum models, in which the dynamic control of soft arms has been achieved by implementing Reinforcement Learning (RL) algorithms [14], [15]. In addition, recently, *SofaGym* was released as an open-source software coupling FEM with RL for different

control tasks [16]: the authors demonstrate the possibility to control soft robots in a learning environment and undergoing deformations induced by soft actuators and by means of their interactions.

Fostering the features of RL algorithms to cope with the challenges of controlling soft robots in the execution of complex tasks, in this work, we formulated the whole-arm grasping inspired by the elephant trunk based on FEM for the modeling of the soft arm and RL for the grasping learning. We developed our methodology in such a way as to harness the compliance and deformability of the soft arm, which is controlled in a simulation environment. In our method, we defined specific points candidate for the contact on the object to be grasped, as well as some control points on the soft arm. The use of contact points on the object is something already employed in rigid robotics [17], [18], while the examples in soft robotics are still a few [19]. As an additional aspect of novelty, in our approach, the contact points are not defined as control points, but rather they induce the desired deformation of the soft arm and its wrapping around the object to perform grasping.

## II. MATERIALS AND METHODS

### A. Soft robot modeling with FEM

In this work, we propose to model a soft arm using FEM. FEM techniques have already been used to simulate soft robots before transferring the behavior to real robots [20]. The simulation environment enables the testing and validation of our approach in conditions that are representative of the manipulation of objects in space. Indeed, we used FEM to model the interaction between the soft arm and another object, ideally representing the target to be captured. We modeled our problem using the FEM framework Simulation Open Framework Architecture (SOFA) [21]. The interest in this framework is in proposing real-time simulation tools for deformable robotics, including already implemented actuators. Considering the scene-graph-based simulation architecture of SOFA, we defined upstream the components to solve the physical system. For instance, among the available solvers, we adopted Euler implicit scheme for time integration. In SOFA, contacts are handled by collision detection algorithms relying on surface models of the objects. Lastly, in SOFA, simulated objects are represented as 3D meshes with a discrete number of nodes, whose state vectors (position, velocity, etc.) are known throughout the simulation. Simulations of our problem take into account the characteristics of the space environment: the external gravity force is null since it is assumed that both the arm and the target object are in a floating condition (the gravity force acting on them is completely balanced by the centrifugal force). In addition, we assume zero relative velocity between the arm and the object at the beginning of simulations.

### B. Background on Reinforcement Learning

In RL algorithms, an agent (for example a robot) is trained to take a sequence of decisions to perform a task. In the learning of this task, the agent interacts with an environment

through actions. Given an action and a current state, the state of the environment evolves and the agent receives a reward that quantifies the quality of his action. The goal of the agent is to maximize an objective function depending on the reward it receives.

Mathematically, RL problems can be modeled as Markov Decision Process (MDP) with states  $s \in \mathcal{S}$ , actions  $a \in \mathcal{A}$ , a transition function  $p(s'|a, s)$  and a reward function  $r_t \sim \mathcal{R}(s_t, a_t)$ . The policy  $\pi(a|s)$  is the action to choose in a given state. We can define the objective function as the expected sum of discounted rewards:

$$\mathcal{J}(\pi) = \mathbb{E}_{\tau \sim \pi} \left[ \sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \right] \quad (1)$$

where  $\mathbb{E}$  is the mathematical expectation,  $\gamma \in [0, 1)$  is a discount factor and  $\tau$  is a sequence of states and actions when the agent follows the policy  $\pi$ . According to MDP theory, there exists an optimal policy that maximizes the objective function. Different approaches can be used to find the optimal policy, such as deep RL Proximal Policy Optimization (PPO) algorithm [22].

### C. Trunk-Robot for space debris manipulation

An existing design of a soft trunk-like cable-driven robot was selected as our soft robotic arm suitable for grasping tasks [20]. The trunk-robot has a conical shape and is actuated by eight cables passing through its silicone body. Four cables are attached to the middle of the trunk-robot and actuate the proximal part, while the other four go through its entire length, actuating the distal part (see Fig. 1). The length of the trunk-robot is about 200 mm.

At the state-of-art of soft robotic actuation technologies, cable-driven actuation was selected as the most appropriate in the harsh conditions of the space environment, such as the presence of vacuum and extreme temperature oscillations [10].

### D. Bio-inspired grasping strategy

The proposed whole-arm grasping strategy of the trunk-robot is founded on observed behaviours of elephants executing grasping tasks with their trunk in their natural environment. Specifically, in [23], on the basis of kinematic analysis of the trunk movements, the authors demonstrated that, in some tasks, elephants adopt recurrent strategies simplifying the biomechanical complexity of the trunk: for instance, both the manipulation and the transportation of objects with diverse size, shape, and weight, are supported by the propagation of a curvature front from the tip of the trunk towards the base. Moreover, they observed a joint-like movement of the trunk consisting of a sharp bend about the trunk half-length. These results reveal a modular behaviour of the elephant trunk in grasping tasks; the higher curvature values created in the distal part of the trunk (the part going from the middle section to the tip) with respect to the proximal part (from the elephant head to the trunk middle section) show that the distal part is mostly designated to dexterous and versatile tasks (such as curling or twisting).

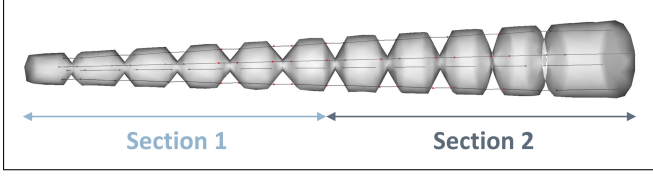


Fig. 1. Design of the soft cable-driven trunk-robot shown in simulation: distal part (section 1) and proximal part (section 2).

In addition, from the observation of the animal behaviour, it emerged that the proximal part of the trunk is devoted to the approach of the trunk to the target object. Given the insights revealed by the elephant behaviour in grasping tasks, in our trunk-robot we implemented the whole-arm grasping as a process in which:

- 1) The proximal part of the trunk approaches the object and the relative position between the trunk and the object is adjusted.
- 2) The distal part of the trunk executes the grasping of the object, curling and wrapping around it.

#### E. Formulation of the whole-arm grasping strategy based on contact points

We formulated the whole-arm grasping as a process harnessing the capabilities of the trunk-robot to adapt to the external environment by means of its soft structure. We selected three points on the trunk-robot longitudinal axis: on the proximal part, at the junction between the proximal and the distal part, and at the end of the distal part (see Fig. 2). The target debris to be captured is represented in simulations through a simplified cube having compatible dimensions with the trunk-robot grasping capabilities. To induce bending in the trunk-robot around the cube, we defined two additional points on the cube belonging to opposite faces (see Fig. 2). In the following, a detailed description of the grasping sequence is given (see Fig. 3A):

- 1) At the beginning of the simulation, the trunk-robot and the cube are at an initial configuration with zero relative velocity.
- 2) Afterwards, in the approach phase, the control point P belonging to the proximal part of the trunk-robot is commanded to approach the cube in correspondence to point M on the cube surface.
- 3) Next, in the deformation phase, the trunk-robot is induced to position the control point D of the distal part as close as possible to point N on the cube surface.
- 4) Lastly, in the wrapping phase, the trunk-robot is pushed to implement the wrapping around the cube by bringing the cube in close proximity to its point T.

We translated the described whole-arm grasping strategy into a reward function by separately accounting for the contribution of each step of the grasping sequence. In detail, the cumulative reward  $r_t$ , at time  $t$ , is expressed as:

$$r_t = -\frac{\|\vec{PM}_t\|}{\|\vec{PM}_0\|} - \frac{\|\vec{DN}_t\|}{\|\vec{DN}_0\|} - \frac{\|\vec{CT}_t\|}{\|\vec{CT}_0\|} \quad (2)$$

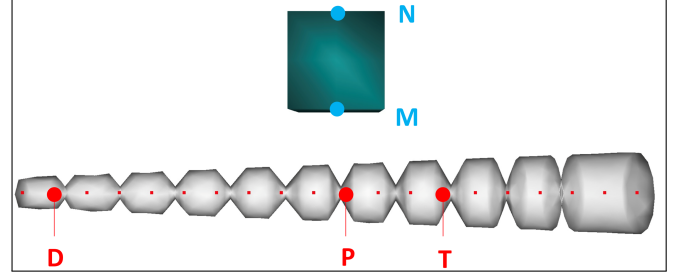


Fig. 2. Trunk-robot in the case of manipulation of a cube: points of interest for the contact on the trunk-robot (D, P and T) and the cube (M and N).

where the vectors are vector distances between one point on the trunk-robot and one point on the cube. In particular,  $\vec{PM}_t$ ,  $\vec{DN}_t$  and  $\vec{CT}_t$  are the vector distances at time  $t$  between point P and M, point D and N, and point C and T, respectively. We compute the Euclidean norm of these vectors and normalize by the initial distances between the points ( $\vec{PM}_0$ ,  $\vec{DN}_0$  and  $\vec{CT}_0$ ), to eliminate the dependence of the reward from the initial relative position of trunk-robot and cube. In addition, minus signs are used to find the fastest strategy. In the learning process of the grasping task, the goal is to maximize  $r_t$ . The novelty of this formulation is in the identification of candidate points for the contact between the trunk-robot and the target object, and in using the selected points to trigger the whole-arm grasping.

#### F. Formulation of the whole-arm grasping based on the trunk-robot shape control

By taking inspiration from the elephant trunk, we formalized an additional grasping strategy implementing explicit control of the curvature of the trunk-robot. The associated grasping sequence is described as follows (see Fig. 3B):

- 1) At the beginning of simulations, the trunk-robot and the cube are at their initial configuration with zero relative velocity
- 2) Afterwards, the control point P belonging to the proximal part of the trunk-robot is commanded to approach the cube by moving along the line connecting point P and the centre of mass C of the cube.
- 3) Then, the distal part of the trunk-robot initializes the bending.
- 4) Lastly, the distal part of the trunk-robot wraps around the cube.

In this case, the mathematical formulation of the reward can be written in the generic form:

$$r_t = \sum_{i,j,k} g_t(K(P_i, P_j, P_k)) + \|P - C\|_2 + h_t(V_C, C) \quad (3)$$

In the previous expression,  $K(P_i, P_j, P_k)$  is a function computing the curvature of the trunk-robot at a given point  $P_j$  belonging to its longitudinal axis, given the position of other two points on the same axis, namely  $P_i$  and  $P_k$ .  $g_t$  is a function applied to the curvature  $K$  to take into account the bending plane, normalizations and signs.  $P$  is the control

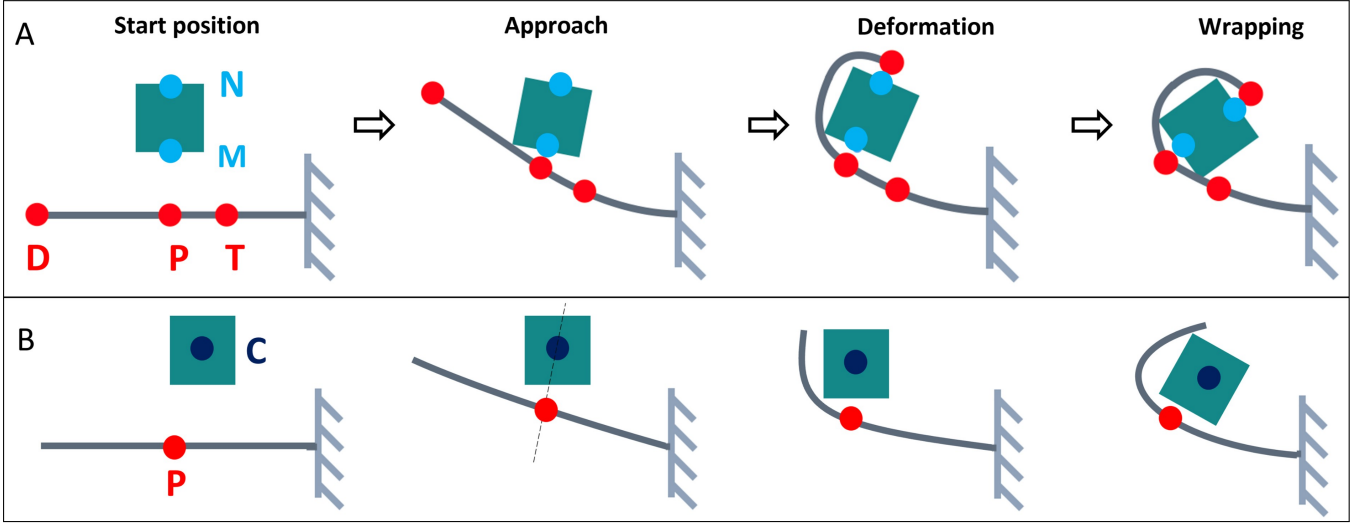


Fig. 3. 2D schematic of the steps taken by the trunk-robot to grasp the object: A) Start position: contact points on the trunk-robot (D, P, T) and the object (M, N), B) Approach: contact between the proximal part of the trunk-robot and the object (point P with point M), C) Deformation: Distal part bending and contact with the object (point D with N), D) Wrapping: Wrapping the trunk-robot around the object by pushing it towards point T.

point on the trunk-robot at the junction between the proximal part and the distal part (see Fig. 2), and  $C$  is the centre of mass of the cube. Lastly,  $h_i$  is a function applied on the position  $C$  and velocity  $V_C$  of the cube.

### G. Robotic implementation in SofaGym

The employed software for the simulation (SOFA) implements RL algorithms via the SofaGym plugin [22]. Here, we selected the PPO algorithm of RL, because it is on-policy and easily parallelized.

In the simulation environment, the observation space is represented by twenty points identified on the trunk-robot: these points are equally distributed along the trunk-robot longitudinal axis, and they are used to track the trunk-robot behaviour in simulations (see Fig. 2). The observation space also contains the geometry, dimensions, and position of the target cube to be grasped. On the other hand, the actuation space is defined by a continuous space of dimension eight, with reference to the number of cables actuating the trunk-robot.

To define the grasping as successful, we considered two factors:

- The distal part of the trunk-robot must achieve the wrapping configuration around the object, in order to ensure distributed contacts.
- The cube must be kept close to the trunk-robot.

Training of the trunk-robot occurs through episodes. Each episode is composed of a series of steps needed to evolve from the initial simulation state to the desired state. At the beginning of each new episode, the size of the cube and its position are randomized, always in accordance with the trunk-robot grasping capabilities. The cube sizes are defined with respect to a reference cube with measures  $10 \times 10 \times 10$  mm, along the  $x, y$ , and  $z$  directions. The cube sizes considered in our simulations are obtained by multiplying the

measures of the reference cube by a scale factor: the lower scale is  $[4, 3, 3]$  for  $x, y$ , and  $z$  respectively, while the higher scale is  $[7, 4, 4]$ . Instead, the cube positions are defined by the positions of the cube centre of mass and are expressed with respect to a reference frame fixed at the base of the trunk-robot. The lower bounds for positions are  $[-10, 25, 80]$  mm in  $x, y$ , and  $z$ , while the higher bounds are  $[10, 50, 110]$  mm. The  $z$ -axis corresponds to the trunk-robot longitudinal axis, while the  $x$ -axis and  $y$ -axis belong to the orthogonal plane. These values of cube sizes and positions represent the boundaries of our training set.

## III. RESULTS AND DISCUSSIONS

### A. Learning through the formulation based on contact points

The main steps implemented by the trunk-robot to grasp the cube are sequentially shown in Fig. 4A: starting from an initial configuration, the trunk-robot approaches the cube by positioning its proximal part, it proceeds with the generation of the bending to reach the selected contact points on the cube, and lastly, it deforms to achieve the wrapping.

Looking at the learning performances, the trunk-robot successfully learns the grasping after a few iterations. As Fig. 5 shows, the cumulative reward increases as the number of iterations increases.

In total, we performed 400 tests by applying the learned policy to the training set, which involves different combinations of the cube size and position. We obtained a success rate of 79,8% (319 successful trials over 400 trials). In most of the failed cases, the trunk-robot successfully implements the wrapping around the cube, but at some point, the cube is dropped. In our opinion, this is not due to the formulation of the grasping strategy, but to the absence of the definition of the end of episodes. In addition, we observed that the trunk-robot encountered difficulties when interacting with small-sized cubes (cubes having scales below  $[4, 3.5, 3.5]$ ).

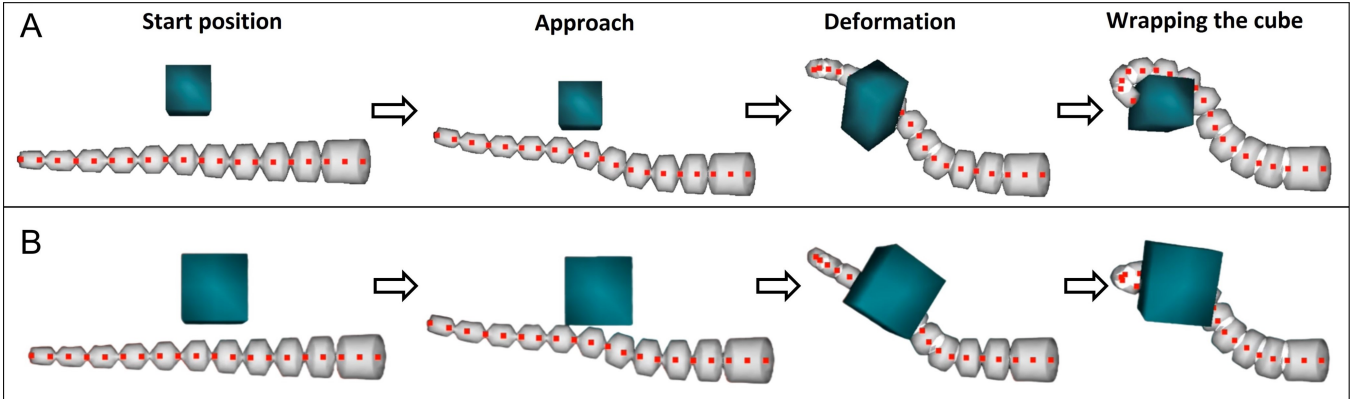


Fig. 4. Grasping steps implemented by the trunk-robot to grasp a cube. A) Test on training set. B) Test on cube size belonging to test set #2.

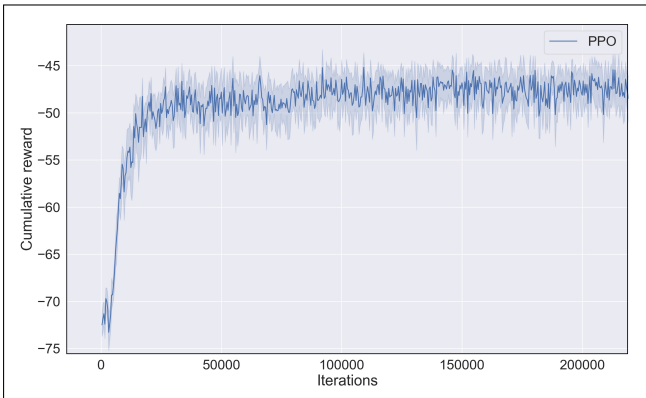


Fig. 5. Learning performance: learning curve as cumulative reward vs. number of iterations.

### B. Validation of the learned policy

In order to evaluate the capability of the learned policy to generalize, the policy has been validated using cube sizes not included in the training set. We considered 3 test sets, and we performed 75 tests in total. For each test set, we performed 25 tests by fixing a value of the cube size unknown in the training, while randomizing the cube positions. For each test set, we obtained the following results:

- Test set #1, cube scale = [8, 3.5, 3.5]. In this test set, we obtained a success rate of 92% (23 successful grasping tests over 25 tests). We observed that the grasping failed due to disturbing interactions between the cube and the trunk-robot in the initial part of the trial.
- Test set #2, cube scale = [5.5, 4.8, 4.8]. In this case, the success rate is 64% (16 successful grasping over 25). The main reason for the failure is due to the big size of the cube, which was almost at the boundaries of the trunk-robot grasping capabilities.
- Test set #3, cube scale = [3, 4, 4]. In this last case, the success rate is 76% (19 successful grasping over 25). Here, some grasping failed due to the small size of the cube, as already observed during the training.

An example of a successful grasping sequence obtained by testing the policy on a cube size unknown in the training

is shown in Fig. 4B.

### C. Discussions

The results show that the learned strategy is successful to grasp cubic objects with various sizes and positions. Through the validation, we demonstrated that the policy can be generalized to other cube sizes unknown in the training. However, the trunk-robot sometimes failed in grasping small-sized cubes. We also observed that the trunk-robot, while executing the grasping, recurrently hits the cube in the initial approach phase, causing undesired cube rotations: this behavior disturbs the learning since the opposite face of the cube defined in the reward by the control point N, does not necessarily keep this configuration during the grasping process. However, despite the highly dynamic environment, in most cases, the trunk-robot exploits those interactions to accomplish the grasping smoothly.

### D. Learning through the formulation based on the robot shape control

With this formulation of the reward function, the trunk-robot was not able to learn the cube grasping. An example of a failed strategy is shown in Fig. 6. We observed that the trunk-robot largely perturbed the speed and/or the position of the cube. However, even by penalizing these actions, the trunk-robot chose to keep a slightly undeformed configuration to minimize the received penalization in case of failed grasping.

## IV. CONCLUSIONS

In this work, we presented a whole-arm grasping strategy for a soft arm to capture space debris. The developed strategy has been inspired by the elephant trunk grasping

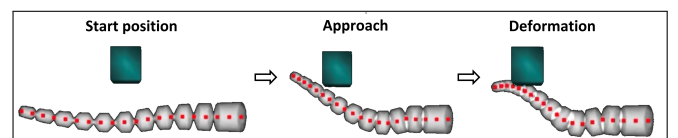


Fig. 6. Example of a failed grasp in the case of learning through explicit deformation of the trunk-robot.

capabilities; the distal part of the elephant trunk is mostly devoted to the execution of dexterous tasks, such as curling for grasping, while the proximal part is mainly used for approaching the target object. Our bio-inspired approach introduced a modular behavior of the soft arm, commanding the proximal part mainly to the positioning, while the distal one to the wrapping. The strategy has been implemented in the framework of SofaGym, a toolkit enabling to couple finite element methods with reinforcement learning techniques. A trunk-like soft arm has been selected to test the methodology and the algorithmic approach, even though the strategy can be extended to other robot designs. We implemented two different grasping strategies, one based on the explicit deformation of the robot, and the other based on the identification of contact points. Related to the strategy based on the explicit deformation of the trunk-robot, we believe that the lack of correlation between the trunk-robot deformation and the cube could be the reason for the failure of this strategy. On the other hand, the strategy based on contact points showed successful results: we formulated a process in which the robot approaches the object, actuates to reach the bending, and simultaneously wraps the body around the object. By using selected contact points, our aim was not to control the grasped object, but instead to give implicit information on the trunk-robot deformation.

The simulation results showed that the strategy is effective to achieve the capture of cubic objects in a zero-gravity environment. The validation of the learned policy on unknown conditions shows that the strategy can be generalized to cubic objects featuring varying sizes, but always consistent with the trunk-robot grasping capabilities. The encountered difficulties in grasping small-sized objects could be overcome by using a finer space discretization of the robot to refine the grip between the objects in contact, or by using more contact points to control the bending of the trunk-robot: this would also induce the trunk-robot to use a wider area to catch the object and ensure a firmer grasp. The use of more contact points or the automatic detection of these points may be an interesting direction to explore in the future. For future work, we may also include a stabilization phase in the grasping strategy, preventing the cube to escape. An additional extension of the work would be the consideration of other geometries for the grasped object. Also, as a further development, the combination of the presented strategy with obstacle avoidance tasks would induce the trunk-robot to interact with the object only at the final wrapping phase. Indeed, since in space, even small actions can induce large motion states in the involved bodies, the absence of interactions in the initial positioning phase could result in a minimization of relative motion or rotation rates between the object and the trunk-robot, also making the grasping more stable.

These preliminary results have been obtained only in simulation. In the future, we plan to test the methodology on a physical prototype as well as in a realistic scenario. Diverse facilities have already been developed to emulate zero-gravity conditions and test robotic arms operations, such

as neutral buoyancy or parabolic flights [7].

## REFERENCES

- [1] C. P. Mark and S. Kamath, "Review of active space debris removal methods," *Space policy*, vol. 47, pp. 194–206, 2019.
- [2] E. Papadopoulos, F. Aghili, O. Ma, and R. Lampariello, "Robotic manipulation and capture in space: A survey," *Frontiers in Robotics and AI*, vol. 8, 2021.
- [3] K. Yoshida, D. Dimitrov, and H. Nakanishi, "On the capture of tumbling debris by a space robot," in *2006 International Conference on Intelligent Robots and Systems*, vol. 8. IEEE/RSJ, 2006.
- [4] R. Lampariello, "On grasping a tumbling debris object with a free-flying robot," in *19th IFAC Symposium on Automatic Control in Aerospace*, vol. 46, no. 19, 2013, pp. 161–166.
- [5] L. Felicetti, P. Gasbarri, and A. Pisculli, "Design of robotic manipulators for orbit removal of spent launcher's stages," *Acta Astronautica*, vol. 119, pp. 118–130, 2016.
- [6] N. Mavrakis, Z. Hao, and Y. Gao, "On-orbit robotic grasping of a spent rocket stage: grasping stability analysis and experimental results," *Frontiers in Robotics and AI*, vol. 8, 2021.
- [7] A. Flores-Abad, O. Ma, K. Pham, and S. Ulrich, "A review of space robotics technologies for on-orbit servicing," in *Progress in Aerospace Sciences*, vol. 68, 2014, pp. 1–26.
- [8] S. Kim, C. Laschi, and B. Trimmer, "Soft robotics: a bioinspired evolution in robotics," *Trends in Biotechnology*, vol. 31, no. 5, 2013.
- [9] J. Zhang, Y. Hu, Y. Li, K. Ma, Y. Wei, J. Yang, Z. Wu, H. Rajabi, H. Peng, and J. Wu, "Versatile like a seahorse tail: A bio-inspired programmable continuum robot for conformal grasping," *Advanced Intelligent Systems*, 2022.
- [10] Y. Zhang, P. Li, J. Quan, L. Li, G. Zhang, and D. Zhou, "Progress, challenges and prospects of soft robotics for space applications," *Advanced Intelligent Systems*, vol. 8, 2022.
- [11] T. G. Thuruthel, Y. Ansari, E. Falotico, and C. Laschi, "Control strategies for soft robotic manipulators: A survey," *Soft Robotics*, vol. 5, no. 2, 2018.
- [12] C. Armanini, C. Messer, A. T. Mathew, F. Boyer, C. Duriez, and F. Renda, "Soft robots modeling: a literature unwinding," *arXiv:2112.03645 [cs.RO]*, 2022.
- [13] D. K. et al, "Review of machine learning methods in soft robotics," *PloSone*, vol. 16, no. 2, 2021.
- [14] N. Naughton, J. Sun, A. Tekinalp, T. Parthasarathy, G. Chowdhary, and M. Gazzola, "Elastica: A compliant mechanics environment for soft robotic control," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3389–3396, 2021.
- [15] A. Centurelli, L. Arleo, A. Rizzo, S. Tolu, C. Laschi, and E. Falotico, "Closed-loop dynamic control of a soft manipulator using deep reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4741–4748, 2022.
- [16] E. Ménager, P. Schegg, E. Khairallah, D. Marchal, J. Dequidt, and et al., "Sofagym: An open platform for reinforcement learning based on soft robot simulations," *Soft Robotics*, 2022.
- [17] Q. V. Le, D. Kamm, A. F. Kara, and A. Y. Ng, "Learning to grasp objects with multiple contact points," in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010.
- [18] H. Douan, P. Wang, G. Xu, W. Wei, and X. Shen, "Robotics dexterous grasping: The methods based on point cloud and deep learning," *Frontiers in Neurobotics*, vol. 15, 2021.
- [19] C. Choi, W. Schwarting, J. Delpreto, and D. Rus, "Learning object grasping for soft robot hands," *2010 IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2370–2377, 2018.
- [20] E. Coevoet, A. Escande, and C. Duriez, "Soft robots locomotion and manipulation control using fem simulation and quadratic programming," in *2019 2nd IEEE International Conference on Soft Robotics (RoboSoft)*. IEEE, 2019.
- [21] F. Faure, C. Duriez, H. Delingette, J. Allard, B. Gilles, S. Marchesseau, H. Talbot, H. Courtecuisse, G. Bousquet, I. Peterlik, and S. Cotin, *SOFA: A Multi-Model Framework for Interactive Physical Simulation*.
- [22] E. Schulman, F. Wolski, A. R. P. Dhariwal, M. Thieffry, and O. Klimov, "Proximal policy optimization algorithms," *arXiv:1707.06347 [cs.LG]*, p. 7, 2017.
- [23] P. Dagenais, S. Hensman, V. Haechler, and M. C. Milinkovitch, "Elephants evolved strategies reducing the biomechanical complexity of their trunk," *Current Biology*, vol. 31, no. 21, pp. 4727–4737, 2021.