Elastica: A Compliant Mechanics Environment for Soft Robotic Control

Noel Naughton, Jiarui Sun, Arman Tekinalp, Tejaswin Parthasarathy, Girish Chowdhary, and Mattia Gazzola

***Abstract*—Soft robots are notoriously hard to control. This is partly due to the scarcity of models and simulators able to capture their complex continuum mechanics, resulting in a lack of control methodologies that take full advantage of body compliance. Currently available methods are either too computational demanding or overly simplistic in their physical assumptions, leading to a paucity of available simulation resources for developing such control schemes. To address this, we introduce Elastica, an open-source simulation environment modeling the dynamics of soft, slender rods that can bend, twist, shear, and stretch. We couple Elastica with five state-of-the-art reinforcement learning (RL) algorithms (TRPO, PPO, DDPG, TD3, and SAC). We successfully demonstrate distributed, dynamic control of a soft robotic arm in four scenarios with both large action spaces, where RL learning is difficult, and small action spaces, where the RL actor must learn to interact with its environment. Training converges in 10 million policy evaluations with near real-time evaluation of learned policies.**

***Index Terms*—And learning for soft robots, control, modeling, reinforcement learning, simulation and animation.**

I. Introduction

THE introduction of soft materials in robotics has long been seen as key to accessing capabilities that are new or complementary to traditionally rigid structures via enhanced dexterity, safety, versatility, and adaptability, with opportunities in industry, agriculture, health care, and defense [1]–[3].

A major challenge in fulfilling this potential is that soft robots are notoriously hard to control [4], [5]. While this is partly related to material and fabrication constraints, from an algorithmic perspective two aspects distinctly set them apart from rigid-body robots. First, the controller needs to orchestrate virtually infinite degrees of freedom via a finite set of actuators. This renders soft robots characteristically hyper-redundant and underactuated [6]. Second, continuum systems are subject to highly non-linear and long range stress propagation effects. Localized loads are communicated throughout the entire structure, potentially inducing global (and sometimes dramatic) shape reconfigurations [7]. Thus, control strategies for compliant robots are inextricably connected to their complex physics. Failure to model and capture such physics often amounts to failing at control.

Reinforcement learning (RL) methods have been proposed to address this complexity, however, the current lack of numerical models able to rigorously, accurately, and efficiently account for the mechanics at play has restricted work to simple, static cases [3], [8]–[11]. The availability of solvers that capture elastic effects would not only provide a useful predictive tool, but would also enable the opportunity to take advantage of compliant deformation modes and instabilities to simplify the control problem. Indeed, it has been shown that the ‘mechanical intelligence’ synthesized by elastic modes [12] can be leveraged to coordinate complex locomotion behaviors [13] as well as topological transitions that can be harnessed for work [7].

Along these lines, compliance allows us to think of obstacles and boundaries as potential allies. In the case of robotic arms and manipulators, solid interfaces are classically dealt with through additional constraints or penalties that render the control problem harder to solve [14], [15]. This active obstacle avoidance strategy is justified in rigid-link robots because impacts with obstacles can cause damage and also to prevent geometric frustration and locking into undesired poses. In contrast, compliant robots can safely conform to, and therefore *exploit*, solid boundaries to correct imprecise actuation, re-distribute excessive loads, or favorably reshape themselves. This contrast may be intuitively summarized as avoiding obstacles versus leaning against them.

To facilitate exploration of these concepts and development of control strategies for soft, slender robotic structures, we introduce Elastica, an open-source simulation environment for solving soft mechanics problems. Our contribution here is the implementation and demonstration of Elastica in a soft robotic context that entails inertial dynamics, distributed actuation and control, and environmental loads as well as its coupling with the RL formalism for closed-loop control in a continuous state-action space. Elastica’s physics engine implements a methodology based on Cosserat rods [16], which are slender, three-dimensional, continuum elements that can bend, twist, shear, and stretch at every cross-section. Their clean mathematical formulation naturally accommodates environmental loads, making them particularly attractive for modeling interface effects

Manuscript received October 15, 2020; accepted February 20, 2021. Date of publication March 3, 2021; date of current version March 23, 2021. This letter was recommended for publication by Associate Editor C. Duriez and Editor C. Laschi upon evaluation of the reviewers’ comments. This work was supported by ONR MURI N00014-19-1-2373, NSF/USDA NRI 2.0 #2019-67021-28989; NSF EFRI #1830881, NSF CAREER #1846752, Blue Waters project (OCI-0725070, ACI-1238993) and XSEDE allocation TG-MCB190004 on TACC’s Stampede2. *(Corresponding author: Mattia Gazzola.)*

The authors are with the Grainger College of Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA (e-mail: nnaught2@illinois.edu; jsun57@illinois.edu; armant2@illinois.edu; tp5@illinois.edu; girishc@illinois.edu; mgazzola@illinois.edu).

This letter has supplementary downloadable material available at https://doi.org/10.1109/LRA.2021.3063698, provided by the authors.

Digital Object Identifier 10.1109/LRA.2021.3063698

such as contact, self-contact, friction, or hydrodynamics. Hence, they are well suited to aid the development of robotic arm counterparts that are soft, flexible, and tailored to reaching and manipulation tasks in unstructured, dynamic environments [6], [17], [18].

Our approach aims to fill the gap between conventional, spring-and-damper rigid body solvers that cannot capture elastic behavior and high-fidelity finite elements methods (FEM), which are mathematically cumbersome and often prohibitively expensive. Elastica's methods have been shown to be both accurate and to strike a valuable compromise between these two approaches. Their accuracy and practical utility has been demonstrated in a number of engineering [16] and biophysical contexts encompassing individual and complex assemblies of Cosserat rods: from design and fabrication of bio-hybrid soft robots made of muscle tissue, neurons, and artificial scaffolds [20]–[22], to dynamics modeling of intricate biological systems such as human elbow joints, snakes, and feathered wings [19].

While Cosserat rod-based methods have been applied to soft robotic control in the past, such work has considered static solvers in simplified environments, drastically restricting the types of control schemes explored. Elastica does not suffer this limitation; it solves the full linear and angular momentum balance equations, formulated to explicitly account for inertial effects as well as endogenous (actuation) and exogenous (environmental) loads. These effects can then be incorporated into novel control strategies, like those discovered via RL.

Here, Elastica is interfaced with Stable Baselines [23], illustrating the coupling of physics and control. We demonstrate the ability of five state-of-the-art, model-free RL methods to deal with increasingly challenging scenarios in which an actor learns to control a soft arm's deformation. Our goal is not necessarily to establish RL as the method of choice for such problems but to illustrate how Elastica enables benchmarking and development of control methods in a soft mechanics context. Nonetheless, our results show RL to be both suitable and convenient for this context given the difficulty of deriving suitable analytical descriptions for model-based techniques.

Overall, our results confirm the successful coupling of RL with Elastica to carry out challenging, dynamic control tasks that are not possible to model with other currently available solvers. They practically illustrate how compliant mechanics and solid boundaries can be used to our advantage. The software interface provided by Elastica allows the user to tap into well-developed control libraries as well as easily define control tasks, variables, actuation modalities, and physical environments, establishing Elastica as a useful testing ground for control strategies of distributed mechanics.

II. Related Work

Physical simulation environments. Because of the unique physics of soft, compliant robots, RL agents must be trained using special-purpose simulation frameworks [24]. Current simulation environments typically used for RL, such as PyBullet [25] and MuJoCo [26], simulate multi-joint dynamics via efficient recursive algorithms combined with modern velocity-stepping methods for contact dynamics. These methods capture the dynamics of rigid robots, however, they intrinsically fail to capture higher-order continuum elastic effects and their associated dynamics, limiting an RL policy's ability to fully exploit all available deformation modes.

Modeling of continuum robots. In a robotic context characterized by large deformations in 3D space, non-linear mechanics, continuous actuation, and interface effects, minimal theoretical models or first-order approximations based on springs and dampers are ill-suited to capture the dynamics of intrinsically soft bodies. At the other end of the spectrum, high-fidelity FEM has been used to simulate and design soft robotic components [27]. However, FEM also exhibits limitations such as often prohibitive computational costs, involved mathematical representation, numerical instabilities when subjected to the large deformations, and inaccuracies due to mesh distortion. Consequently, FEM has been relatively limited in the modeling of soft robots, particularly in combination with control, though recent work has begun to address these limitations [28], [29]. Real-time simulation speeds have also recently been achieved using both FEM [27] and an asynchronous multi-body framework (AMBF) [30].

Alternative approaches often seek to leverage geometric slenderness. Slender objects are then treated as one-dimensional elastic curves, significantly reducing mathematical complexity and computational costs while retaining physical accuracy. The graphics community has been active in this area, with spline-based strands [31], discrete rod models [32] (based on the unstretchable and unshearable Kirchhoff model [33]), and varying diameter rods [34] routinely used in a variety of realistic simulations, from elastic ribbons and woven cloth to entangled hair, muscles and tendons. Similar methods have been used in robotics [6] to model soft arms [11], [35]–[37], snake robots [38], and surgical manipulators [39]. Recently, a discrete differential geometry (DDG) based approach has also been introduced [40].

Though numerically efficient, these previous approaches are specialized to scenarios where either shear, stretch, twist, dynamic effects, or environmental loads are unimportant. Lately, advances in soft robotics related to artificial muscles [7], stretchable and shearable elastomers, and integration of biocomponents [20] has raised the need to extend these models and incorporate previously neglected effects. Thus, the more comprehensive Cosserat rod model [41] has been gaining attention with its utility recently demonstrated in a range of applications, from soft robotics to biophysics [19].

Reinforcement learning for soft robotic control. Soft robots are difficult to control with traditional methods due to their virtually infinite degrees of freedom and highly nonlinear continuum dynamics [3], [5], [8]. This has created fertile ground for using model-free RL to control soft robots. For example, Satheeshbabu *et al.* [9], [10] presented model-free approaches for position control of a soft spatial-continuum arm using variants of Deep Q-learning [42] and Deep Deterministic Policy Gradient (DDPG) [43]. Uppalapati *et al.* [11] showed DDPG-based control of a hybrid rigid-soft arm in cluttered agricultural environments. Notably, since these RL methods are sample expensive, these authors used static, semi-analytical models to

TABLE I
COMPARISON OF ELASTICA WITH PREVIOUSLY PUBLISHED SIMULATORS

Ref.	Model Class	Soft Physics	Dynamics	All Def. Modes	Env. Interact.	Coupled w/ RL
Elastica	dyn. Coss. rod	X	X	X	X	X
[10], [32]	static Kirch. rod	X				X
[36], [39]	static Coss. rod	X		X	X	
[40]	DDG	X	X		X	
[30]	AMBF	X	X		X	
[27]	SOFA	X	X	X	X	
[25], [26]	MuJoCo, PyBullet		X		X	X

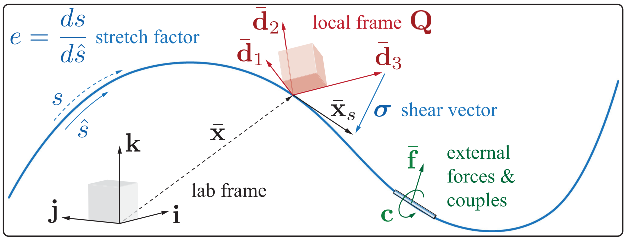

Fig. 1. Cosserat rod model used in Elastica [16], [19].

train the RL policies, subsequently evaluated on real soft robot arms in a laboratory setting. The resulting policies demonstrate feasibility of the RL approach for soft robotics, but the limiting assumptions of the models restrict the real world performance of the learned policies. Our framework instead provides modeling and simulation capabilities suitable for more realistic physical conditions.

Overall, in a soft robotic control context characterized by fast elastic dynamics, distributed actuation, and interfacial environmental effects, currently available methods present a range of deficiencies that significantly limit their scope and real-world application. Table I summarizes how current methods either do not model soft-body physics, do not model dynamics, or are too computationally demanding to be coupled with RL. In contrast, Elastica accurately resolves the dynamics of elastic bodies that can bend, twist, shear, and stretch and incorporates both internal muscular activity and external environmental loads at a computational cost linear with the number of discretization elements. As such, we believe Elastica will be a potent new tool for the soft robotics and control communities.

III. COMPUTATIONAL ENVIRONMENT

Cosserat rod model. Based on Cosserat rod theory [41], we describe a rod (slender body, Fig. 1(a)) by a centerline $\bar{\mathbf{x}}(s, t) \in \mathbb{R}^3$ and rotation matrix $\mathbf{Q}(s, t) = \{\bar{\mathbf{d}}_1, \bar{\mathbf{d}}_2, \bar{\mathbf{d}}_3\}^{-1}$ which leads to a general relation between frames for any vector $\mathbf{v}$: $\mathbf{v} = \mathbf{Q}\bar{\mathbf{v}}$, $\bar{\mathbf{v}} = \mathbf{Q}^T\mathbf{v}$, where $\bar{\mathbf{v}}$ denotes a vector in the lab frame and $\mathbf{v}$ denotes a vector in the local frame. Here $s \in [0, L_0]$ is the material coordinate of a rod of rest-length L_0, L denotes the deformed filament length, and t is time. If the rod is unsheared, $\bar{\mathbf{d}}_3$ points along the centerline tangent $\partial_s\bar{\mathbf{x}} = \bar{\mathbf{x}}_s$ while $\bar{\mathbf{d}}_1$ and $\bar{\mathbf{d}}_2$ span the normal-binormal plane. Shearing and extension shift $\bar{\mathbf{d}}_3$ away from $\bar{\mathbf{x}}_s$, which can be quantified with the shear vector $\boldsymbol{\sigma} = \mathbf{Q}(\bar{\mathbf{x}}_s - \bar{\mathbf{d}}_3) = \mathbf{Q}\bar{\mathbf{x}}_s - \mathbf{d}_3$ in the *local* frame. The curvature vector $\boldsymbol{\kappa}$ encodes $\mathbf{Q}$'s rotation rate along the material coordinate $\partial_s\mathbf{d}_j = \boldsymbol{\kappa} \times \mathbf{d}_j$, while angular velocity $\boldsymbol{\omega}$ is defined by $\partial_t\mathbf{d}_j = \boldsymbol{\omega} \times \mathbf{d}_j$. We also define the velocity of the centerline $\bar{\mathbf{v}} = \partial_t\bar{\mathbf{x}}$ and, in the rest configuration, the bending $\mathbf{B}$ and shearing $\mathbf{S}$ stiffness matrices, second area moment of inertia $\mathbf{I}$, cross-sectional area A and mass per unit length ρ. Then, the dynamics of a slender body reads as

$$\rho A \cdot \partial_t^2\bar{\mathbf{x}} = \partial_s\left(\frac{\mathbf{Q}^T\mathbf{S}\boldsymbol{\sigma}}{e}\right) + e\bar{\mathbf{f}} \tag{1}$$

$$\frac{\rho\mathbf{I}}{e} \cdot \partial_t\boldsymbol{\omega} = \partial_s\left(\frac{\mathbf{B}\boldsymbol{\kappa}}{e^3}\right) + \frac{\boldsymbol{\kappa} \times \mathbf{B}\boldsymbol{\kappa}}{e^3} + \left(\mathbf{Q}\frac{\bar{\mathbf{x}}_s}{e} \times \mathbf{S}\boldsymbol{\sigma}\right)$$
$$+ \left(\rho\mathbf{I} \cdot \frac{\boldsymbol{\omega}}{e}\right) \times \boldsymbol{\omega} + \frac{\rho\mathbf{I}\boldsymbol{\omega}}{e^2} \cdot \partial_t e + e\mathbf{c} \tag{2}$$

where (1), (2) represent the linear and angular momentum balance, $e = |\bar{\mathbf{x}}_s|$ is the local stretching factor, and $\bar{\mathbf{f}}$ and $\mathbf{c}$ are the external force and couple line densities, respectively.

This representation entails a number of favorable features: *1)* it captures 3D dynamics accounting for all modes of deformation – bend, twist, shear, and stretch; *2)* continuum actuation, interface effects, and environmental loads can be directly combined with body dynamics via $\bar{\mathbf{f}}$ and $\mathbf{c}$, making their inclusion straightforward; *3)* its complexity scales linearly with axial resolution, compared to cubic for FEM, significantly reducing compute time. Discretization of the above system of equations, along with appropriate boundary conditions, allows modeling dynamics of multiple active or passive Cosserat rods interacting with each other and their environment. Interactions between rods are modeled using displacement-force relations as detailed in [16], [19].

IV. EXPERIMENTS

Simulation and problem setup. A particularly promising area of soft robotics is the development of continuum, compliant arms capable of reaching and manipulation tasks in complex, dynamic environments. Often inspired by octopus arms [6], [17], [18], these hyper-redundant, compliant robots promise a host of advantages such as increased maneuverability, dexterity, and safety. These robots are particularly amenable to being represented within Elastica as they can be accurately modeled as single, slender rods. Here, we consider four scenarios consisting of both large action spaces, where the RL actor must explore efficiently, and small action spaces, where the actor must learn to interact with its environment to accomplish the task. Together, these results illustrate the applicability and potential advantages of RL-based control in soft robotics. Code to reproduce all cases is available online and videos of the highest performing policies for all cases are available in the SI dataset

In all cases the goal is for the tip of the arm to reach a target location, complemented by additional, case-specific requirements. *Case 1:* tracking a randomly moving target in 3D space. *Case 2:* reaching to a randomly located stationary point and orienting the arm so the tip of the arm matches a randomly prescribed target orientation. *Case 3:* learning to interact with and exploit solid boundaries to enable underactuated maneuvering through structured obstacles. *Case 4:* underactuated maneuvering through an unstructured nest of obstacles. An episode reward score above

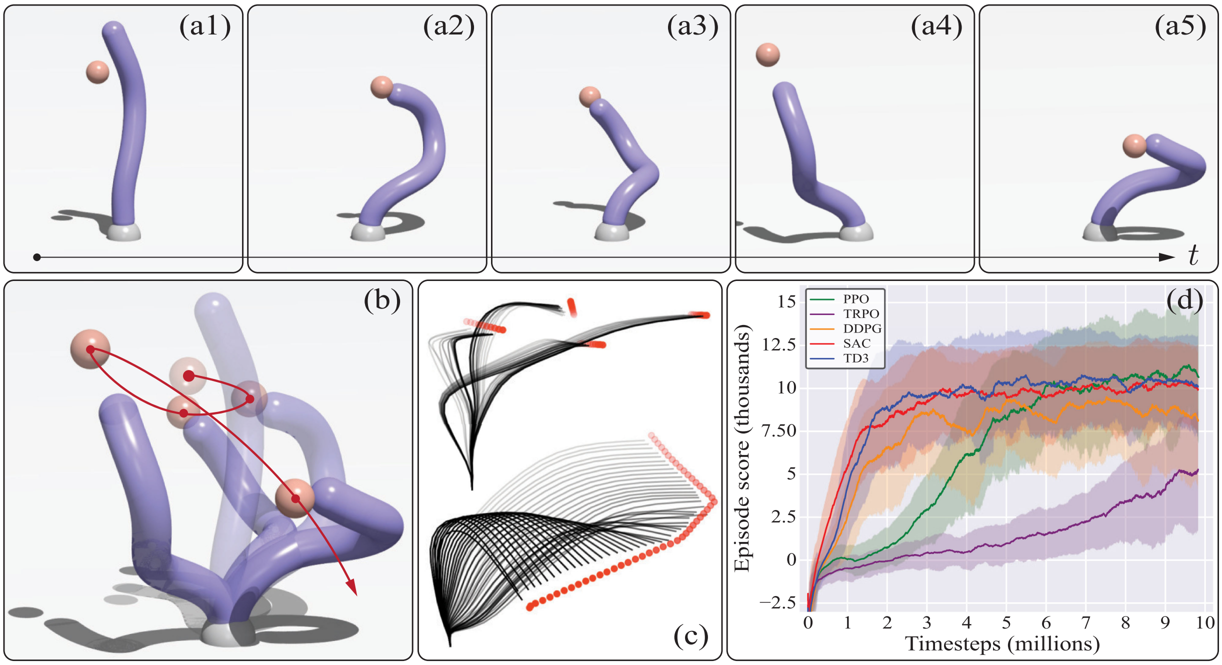

Fig. 2. a) Snapshots from a trained policy (SAC) over the course of one episode showing the arm successfully tracking a randomly moving target. b) Overlay of snapshots showing the random trajectory of the target. c) Trajectory of arm centerline and the target over successive timesteps. d) Learning results of the different algorithms. Algorithms were trained with 5 different random seeds for 10 millions timesteps. Curves are the rolling 250 sample average of combined results. Shaded regions are the standard deviation of the sample.

zero is indicative of (at least partially) successful task completion, with higher scores indicating faster and more consistent task completion.

The arm is modeled as a single Cosserat rod fixed upright at its base and free to move in 3D space. The arm has a Young's modulus of 10 MPa, leading to a bending stiffness typical of soft robotic arms [4]. Arm actuation occurs via application of internal torques distributed along the length of the arm. These continuous activation functions are modeled via splines characterized by N control points and vanishing values (i.e. zero couple) at the arm's extrema [16]. The arm is controlled by decomposing the overall actuation into orthogonal torque functions applied in the local normal and binormal directions (i.e. along $\mathbf{d}_1$ and $\mathbf{d}_2$), causing the arm to bend, and in the orthonormal direction $\mathbf{d}_3$, causing the arm to twist. Different actuation modes (only bending or bending/twisting) are provided for each case. Details of the action spaces, states, and rewards for each case, as well as specific simulation parameters used, are available in the SI dataset.

Selected RL methods. To investigate RL's ability to dynamically control a compliant robotic arm in Elastica, five model-free, policy-gradient RL methods were considered, consisting of two algorithms implemented as on-policy—Trust Region Policy Optimization (TRPO) [44]; Proximal Policy Optimization (PPO) [45]—and three off-policy algorithms—Soft Actor Critic (SAC) [46]; Deep Deterministic Policy Gradient (DDPG) [43]; Twin Delayed DDPG (TD3) [47]. These are considered to be state-of-the-art RL for continuous control with demonstrated performance in a variety of tasks. We used implementations provided by the Stable Baselines library [23]. Limited hyperparameter tuning was performed. While the lack of extensive hyperparameter tuning suggests that the scores reported here may not be the maximum attainable for each algorithm, the purpose of this work is not to adjudicate which of the selected algorithms is the best at these particular cases, but rather demonstrate their utility in combination with Elastica, and to establish a baseline against which these and other algorithms can be measured.

On a single CPU core, policy convergence took 10–20 hours (depending on the specific algorithm). For example, 10 million RL training steps in Case 1 (equivalent to ~4 hours of physical simulated time), TRPO and PPO required ~11 hours to complete while SAC, DDPG, and TD3 required ~22 hours. After training, simulating the arm for 10 physical seconds has a time-to-solution of ~16 seconds, i.e. near real-time.

V. Results and Discussion

Case 1 – 3D tracking of a randomly moving target. The first case consists of the tip of the arm continuously tracking a randomly moving target in 3D space as illustrated in Fig. 2(a)–(c). The reward function $R = -n^2 + \phi(n)$ penalizes the distance between the arm's tip and target $n = ||x_n - x_t||$ combined with a two-tier bonus reward $\phi(n)$ as the tip approaches the target. Actuation is allowed only in the normal and binormal directions (3D bending, but no twist). The actuation function in each direction is controlled by 6 equidistantly spaced control points leading to an action space with 12 degrees of freedom (DOF). The state $S = [x_a, v_a, x_t, v_t]$ is the location of 11 points spaced equidistantly along the arm x_a, the arm tip's velocity v_a, the target location x_t, and the target's velocity v_t. Hyperparameter tuning was limited to batch size (1000 to 128 k) for on-policy

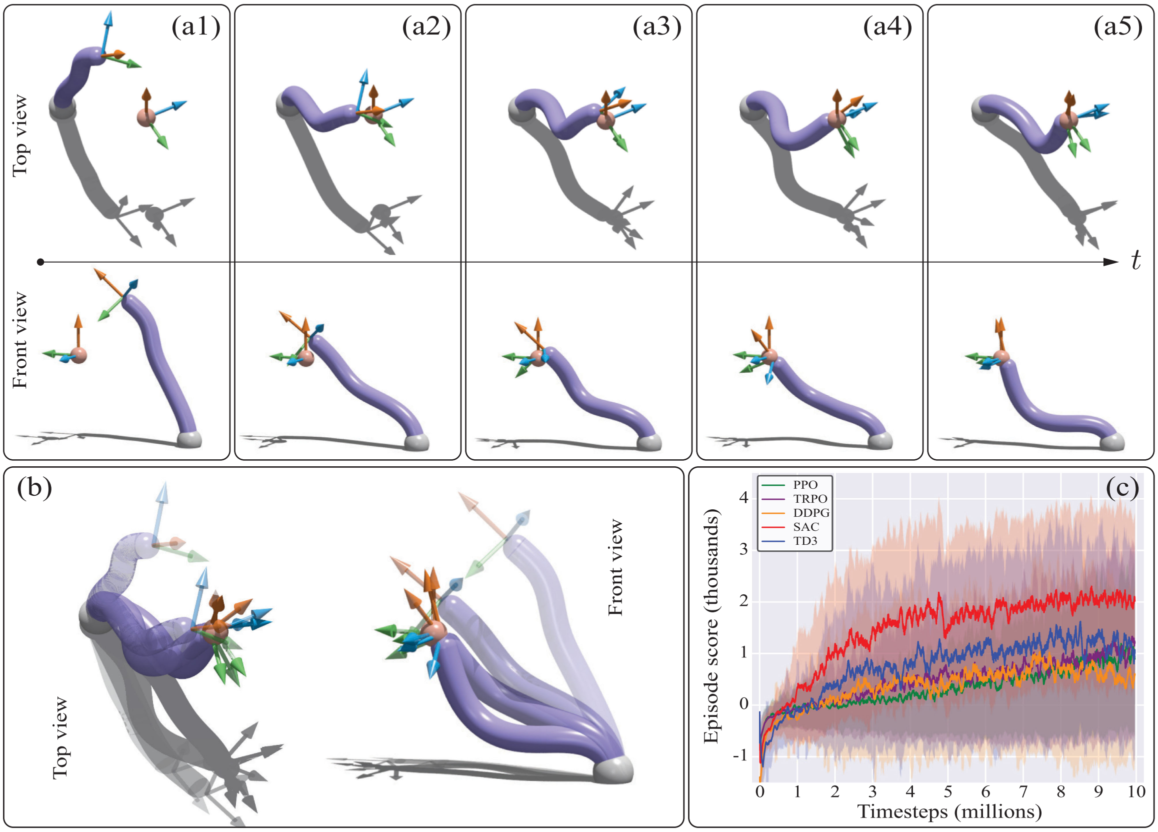

Fig. 3. a) Snapshots from a trained policy (SAC) over the course of one episode showing the arm successfully reaching the target and then orienting itself to match the target orientation. b) Overlay of the snapshots. c) Learning results of the different algorithms. Algorithms were trained with 5 different random seeds for 10 millions timesteps. Curves are the rolling 250 sample average of combined results. Shaded regions are the standard deviation of the sample.

methods (TRPO, PPO), and replay buffer size (100 k to 2 M) for off-policy methods (SAC, DDPG, TD3). Policies were trained for 10 million (TRPO, PPO) or 5 million timesteps (SAC, DDPG, TD3). Hyperparameter tuning results are available in the SI dataset.

For on-policy methods, the best performance is achieved with a batch size of 8000 for TRPO and 32,000 for PPO. For off-policy methods, all three methods achieve best performance with a replay buffer of 2 million. Learning curves of the best performances (for 10 million training timesteps) are shown in Fig. 2(d). All methods learn to satisfactorily track the target, albeit with differences. SAC, TD3, and PPO achieve similar scores, with SAC and TD3 converging the fastest. DDPG initially learns at a similar rate but converges to a score ∼20% lower. TRPO achieves the lowest score. Qualitatively, some of the learned policies occasionally exhibit relatively high frequency motions (see supplementary video) due to the short update interval. While this could be addressed with a reward function penalty term, it is not considered here to keep the problem statement as unconstrained as possible.

Case 2 – Reaching target with defined orientation. Manipulating objects by changing their orientation is a key use case for robotic arms. Case 2 consists of reaching to a randomly-located, stationary target while reshaping to match a desired end-effector orientation (Fig. 3(a)–(b)). The target coordinate frame is defined with the axial direction ($\bar{\mathbf{d}}_3$) pointing up vertically and the normal-binormal directions ($\bar{\mathbf{d}}_1$, $\bar{\mathbf{d}}_2$) randomly rotated in-plane. The reward function, $R = -n^2 - 0.5p^2 + \phi(n,p)$, is like Case 1 but adds a penalty p for the difference in the tip and target's orientation and a bonus $\phi(n,p)$ as the orientations align and the tip reaches the target. To satisfactory solve this problem, inclusion of twist is required. As with bending actuation, twist is controlled by six equidistantly distributed control points. Bending and twisting actuation yields an 18 DOF action space. The state $S = [x_a, v_a, q_a, x_t, v_t, q_t]$ adds two quaternions q_a and q_t to represent the orientation of the arm tip and target.

Hyperparameter tuning was done in the same manner as Case 1. Best performance is seen for a batchsize of 16,000 for both TRPO and PPO and, as in Case 1, the best replay buffer size for SAC, DDPG and TD3 is 2 million samples. Learning curves for the best algorithms are shown in Fig. 3(c). All algorithms learn to at least partially complete the task, and SAC outperforms all others with a final score almost twice the second best. All other algorithms exhibit similar performance. Notably, TRPO and PPO have similar performance, in contrast to PPO outperforming TRPO in Case 1. Finally, all algorithms exhibit large variance, explained by the fact that not all target location/orientation pairs are physically attainable by the arm.

Cases 1 and 2 demonstrate that RL methods can learn to control soft bodies in 3D space and effectively manipulate their pose via distributed deformation modes generally not available to their rigid counterparts, particularly twist. Next, we challenge these methods to learn to advantageously interact with the environment through the addition of obstacles.

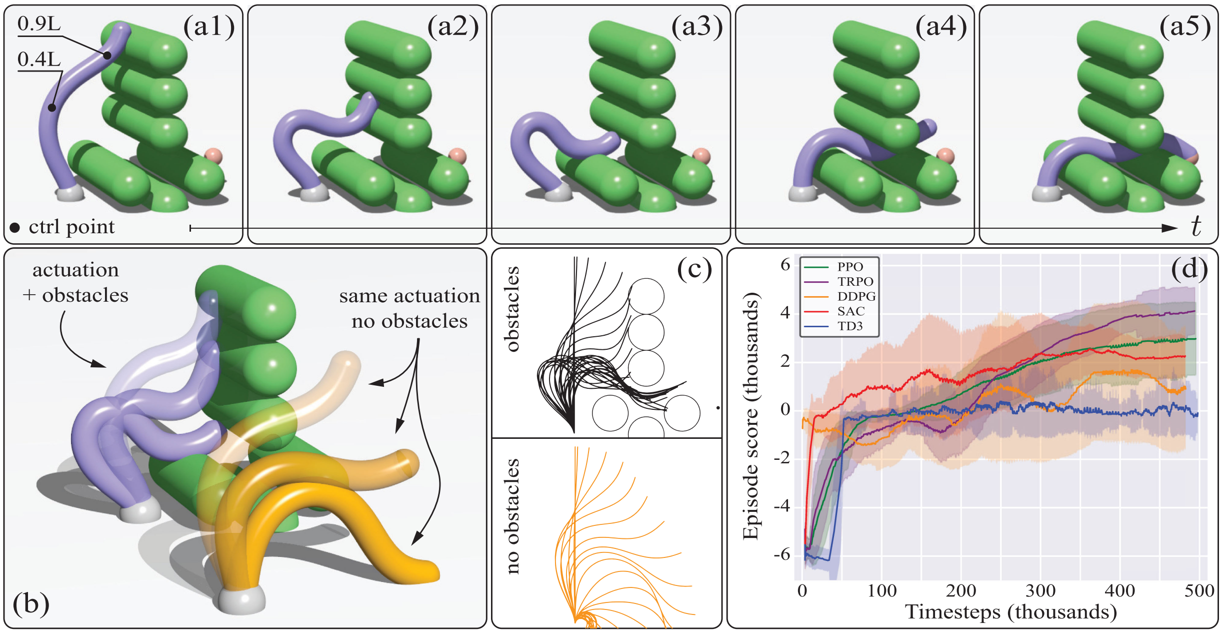

Fig. 4. a) Snapshots from a trained policy (TRPO) showing the arm leveraging obstacles to maneuver through the opening and reach the target. b) Comparison of the behavior for of the applied actuation in the presence and absence of obstacles. c) Timelapse of arm centerlines with and without obstacles showing how interaction with obstacles is key to successfully maneuvering through the opening. d) Learning curve for algorithms trained with 5 random seeds for 500 thousand timesteps. Curves are the rolling 50 sample average of the combined results. Shaded regions are the standard deviation of the sample.

Case 3 – Maneuvering between structured obstacles. A major advantage of compliant robots is their ability to maneuver around obstacles without damaging themselves or the obstacles. To explore the ability of model-free methods to learn to interact with and take advantage of solid boundaries, a target is placed behind a wall of obstacles with an opening through which the arm must reach (Fig. 4). The target is placed in the normal plane so only in-plane actuation is required. The obstacles and target are in the same configuration for all episodes and the reward is the same as Case 1. Importantly, no penalty is included to avoid contact with obstacles. Indeed, we do not see them as additional constraints but rather features to be exploited. Our focus here is on learning to interact and use obstacles to complement control, and we do not consider the generalization of the environment to arbitrarily located obstacles.

The obstacles are arranged such that it is impossible for the arm to fit through the opening without bending around or conforming to them. This results in a problem that cannot be solved by a rigid-link arm with a small number of DOFs but may be solvable by a compliant arm with comparable DOFs. To explore the interplay of underactuation and boundaries, only two control points at locations 0.4 L and 0.9 L along the arm are used. The rationale being that actuation at the mid-control point (0.4 L) can organize approximate global deformation sufficient to point the tip towards the opening and subsequently push the arm in that general direction. Actuation at 0.9 L helps navigate the obstacles by bending the tip to determine which surfaces the arm slides along when pushed.

The state is the same as in Case 1 but with the addition of obstacle locations: $S = [x_a, v_a, x_t, x^n_{obs}]$. Limited hyperparameter tuning was performed for off-policy methods while on-policy methods used hyperparameters from Case 2. Because the target location was not random, only 500 thousand training timesteps were needed. On-policy methods, TRPO in particular, were successful, as shown in Fig. 4(a), extensively using the obstacles to correct and redirect the imprecise actuation intrinsic to the challenging and extremely underactuated two DOF setup. Off-policy methods were found to explore the action space vigorously, leading to large external contact forces from slamming the arm into obstacles that caused numerical instabilities at the selected numerical discretization resolution and prevented successfully learning. Although actuation constraints could remove this instability, we purposefully allowed the system to remain unconstrained to test if the algorithms could learn to remain stable.

Case 4 – Maneuvering between unstructured obstacles. The final case expands on the arm's ability to interact with its environment by asking it to find its way through a nest of unstructured obstacles that are in the same configuration for each episode (Fig. 5(a)–(b)). The reward function, state, and actuation control points are the same as Case 3. However, to navigate through the nest, 3D bending is necessary. Therefore, internal torques are allowed to act in the normal and binormal directions, resulting in an action space with four DOFs. As with Case 3, minimal hyperparameter tuning was performed. Policies were trained for 1 million timesteps (Fig. 5(c)). Performance is similar to Case 3 with TRPO and PPO successfully learning to complete the task and off-policy methods generally unable to select non-catastrophically violent actions. As with Case 3, this problem is extremely challenging, if not impossible, for a rigid-link robot with comparable DOFs. In contrast, the compliant arm is able to maneuver though the nest by extensively leaning against various surfaces to redirect the tip towards the target.

The key aspect of the underactuated control demonstrated here is the coupling of the compliant arm with its environment. A compliant robot can solve this problem with only two control

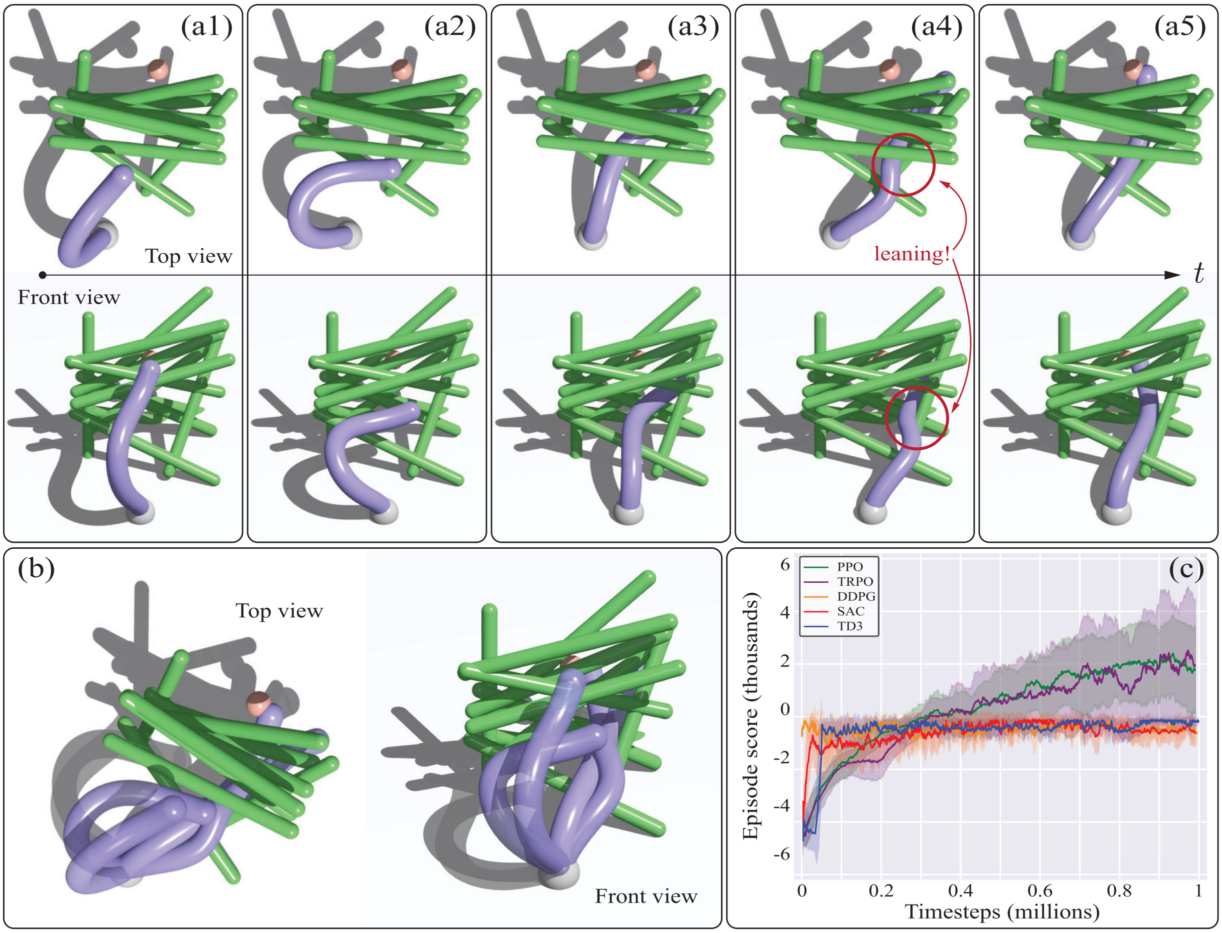

Fig. 5. a) Snapshots from a trained policy (TRPO) showing the arm successfully maneuvering through the obstacle nest to reach the target. b) Timelapse of arm maneuvering through obstacles. c) Learning curve for algorithms trained with 5 random seeds for 1 million timesteps. Curves are the rolling 50 sample average of the combined results. Shaded regions are the standard deviation of the sample.

points because of its ability to lean against and conform to obstacles, which in turn corrects the arm's inaccurate actuation. Fig. 4(b)–(c) demonstrates this point, showing how the same action produces different arm deformations when interacting with obstacles versus when obstacles are not considered. Further, Fig. 5 a illustrates how the compliant arm leans against obstacles to maneuver through them. If a traditional approach of avoiding obstacles via penalty terms was employed [14], the flexible arm would be unable to complete these tasks as two DOFs do not provide the necessary finesse to maneuver without encountering the boundaries (see SI dataset for results). In contrast, when the arm is allowed to interact with its environment, RL finds it natural to make use of boundaries as a resource, thus effectively simplifying the control problem. However, this is only possible when elastic effects are properly considered, demonstrating how Elastica can help spur the development of efficient control strategies for soft robots that make full use of their compliance, unlike traditional rigid-body simulators.

VI. Conclusion

To fully realize the promised benefits of soft robots it is necessary to develop control methods that exploit their unique physical properties. This is complicated by the difficulty of accurately modeling compliant structures in a simulation environment. Currently available simulation testbeds are insufficient in this regard. To address this, we introduced Elastica, an open-source physics environment for simulating assemblies of soft, slender, and compliant rods (as well as rigid-body structures). We interfaced Elastica with preexisting RL implementations to enable simulation-based learning for dynamic control of soft robots. We showed that state-of-the-art RL algorithms (TRPO, PPO, DDPG, TD3, and SAC) can learn to control a soft arm's dynamic behavior and complete successively challenging tasks, with PPO demonstrating the most consistent performance. We further demonstrated how modeling the arm's compliant mechanics and interaction with the environment can help simplify the control problem. Source code for Elastica is available online, allowing these cases to serve as benchmarks for new control and learning algorithms.

Software and Data Availability

The open-source Python implementation of Elastica is available at www.github.com/GazzolaLab/PyElastica. A supplementary dataset with implementation details, hyperparameter tuning results, and code for the different cases presented is available at www.cosseratrods.org/Elastica+RL.

REFERENCES

[1] M. Cianchetti, C. Laschi, A. Menciassi, and P. Dario, “Biomedical applications of soft robotics,” *Nat. Rev. Mater.*, vol. 3, no. 6, pp. 143–153, 2018.

[2] P. Polygerinos *et al.*, “Soft robotics: Review of fluid-driven intrinsically soft devices; manufacturing, sensing, control, and applications in human-robot interaction,” *Adv. Eng. Mater.*, vol. 19, no. 12, 2017, Art. no. 1700016.

[3] G. Chowdhary, M. Gazzola, G. Krishnan, C. Soman, and S. Lovell, “Soft robotics as an enabling technology for agroforestry practice and research,” *Sustainability*, vol. 11, no. 23, 2019, Art. no. 6751.

[4] D. Rus and M. T. Tolley, “Design, fabrication and control of soft robots,” *Nature*, vol. 521, no. 7553, pp. 467–475, 2015.

[5] T. G. Thuruthel, Y. Ansari, E. Falotico, and C. Laschi, “Control strategies for soft robotic manipulators: A. survey,” *Soft Robot.*, vol. 5, pp. 149–163, no. 2, 2018.

[6] D. Trivedi, C. D. Rahn, W. M. Kier, and I. D. Walker, “Soft robotics: Biological inspiration, state of the art, and future research,” *Appl. bionics and biomechanics*, vol. 5, no. 3, pp. 99–117, 2008.

[7] N. Charles, M. Gazzola, and L. Mahadevan, “Topology, geometry, and mechanics of strongly stretched and twisted filaments: Solenoids, plectonemes, and artificial muscle fibers,” *Phys. Rev. Lett.*, vol. 123, 2019, Art. no. 208003.

[8] K. Chin, T. Hellebrekers, and C. Majidi, “Machine learning for soft robotic sensing and control,” *Adv. Intell. Syst.*, vol. 2, no. 6, 2020, Art. no. 1900171.

[9] S. Satheeshbabu, N. K. Uppalapati, G. Chowdhary, and G. Krishnan, “Open loop position control of soft continuum arm using deep reinforcement learning,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2019, pp. 5133–5139.

[10] S. Satheeshbabu, N. K. Uppalapati, T. Fu, and G. Krishnan, “Continuous control of a soft continuum arm using deep reinforcement learning,” in *Proc. 3rd IEEE Int. Conf. Soft Robot.*, 2020, pp. 497–503.

[11] N. K. Uppalapati, B. Walt, A. Havens, A. Mahdian, G. Chowdhary, and G. Krishnan, “A berry picking robot with a hybrid soft-rigid arm: Design and task space control,” in *Proc. Robot.: Sci. Syst.*, Corvalis, Oregon, USA, 2020, Paper 27.

[12] R. Pfeifer, M. Lungarella, and F. Iida, “Self-organization, embodiment, and biologically inspired robotics,” *Science*, vol. 318, no. 5853, pp. 1088–1093, 2007.

[13] M. Gazzola, M. Argentina, and L. Mahadevan, “Gait and speed selection in slender inertial swimmers,” *Proc. Natl. Acad. Sci.*, vol. 112, no. 13, pp. 3874–3879, 2015.

[14] O. Khatib, “Real-time obstacle avoidance for manipulators and mobile robots,” in *Auton. Robot Veh.* Springer, 1986, pp. 396–404.

[15] H.-S. Chang *et al.*, “Energy shaping control of a cyberoctopus soft arm,” in *Proc. 59th IEEE Conf. Decis. Control.*, 2020, pp. 3913–3920.

[16] M. Gazzola, L. Dudte, A. McCormick, and L. Mahadevan, “Forward and inverse problems in the mechanics of soft filaments,” *Roy. Soc. Open Sci.*, vol. 5, no. 6, 2018, Art. no. 171628.

[17] I. D. Walker *et al.*, “Continuum robot arms inspired by cephalopods,” in *Proc. Unmanned Ground Veh. Technol. VII*, Int. Soc. Opt. Photon., vol. 5804, 2005, pp. 303–314.

[18] M. Calisti *et al.*, “An octopus-bioinspired solution to movement and manipulation for soft robots,” *Bioinspiration Biomimetics*, vol. 6, no. 3, 2011, Art. no. 036002.

[19] X. Zhang, F. K. Chan, T. Parthasarathy, and M. Gazzola, “Modeling and simulation of complex dynamic musculoskeletal architectures,” *Nat. Commun.*, vol. 10, no. 1, pp. 1–12, 2019.

[20] J. Wang *et al.*, “Computationally assisted design and selection of maneuverable biological walking machines,” *Adv. Intell. Syst.*, 2021, Art. no. 2000237.

[21] G. J. Pagan-Diaz *et al.*, “Simulation and fabrication of stronger, larger, and faster walking biohybrid machines,” *Adv. Funct. Mater.*, vol. 28, no. 23, 2018, Art. no. 1801145.

[22] O. Aydin *et al.*, “Neuromuscular actuation of biohybrid motile bots,” *Proc. Natl. Acad. Sci.*, vol. 116, no. 40, pp. 19 841–19 847, 2019.

[23] A. Hill *et al.*, “Stable Baselines,” 2018. [Online]. Available: https://github.com/hill-a/stable-baselines

[24] S. Bhagat, H. Banerjee, Z. Ho Tse, and H. Ren, “Deep reinforcement learning for soft, flexible robots: Brief review with impending challenges,” *Robot.*, vol. 8, no. 1, p. 4, Jan 2019.

[25] E. Coumans and Y. Bai, “Pybullet, a Python Module for Physics Simulation for Games, Robotics and Machine Learning,” 2016–2020. [Online]. Available: http://pybullet.org

[26] E. Todorov, T. Erez, and Y. Tassa, “Mujoco: A. physics engine for model-based control,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2012, pp. 5026–5033.

[27] E. Coevoet *et al.*, “Software toolkit for modeling, simulation, and control of soft robots,” *Adv. Robot.*, vol. 31, no. 22, pp. 1208–1224, 2017.

[28] O. Goury and C. Duriez, “Fast, generic, and reliable control and simulation of soft robots using model order reduction,” *IEEE Trans. Robot.*, vol. 34, no. 6, pp. 1565–1576, Dec. 2018.

[29] R. K. Katzschmann *et al.*, “Dynamically closed-loop controlled soft robotic arm using a reduced order finite element model with state observer,” in *Proc. 2nd IEEE Int. Conf. Soft Robot.*, 2019, pp. 717–724.

[30] A. Munawar, N. Srishankar, and G. S. Fischer, “An open-source framework for rapid development of interactive soft-body simulations for real-time training,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 6544–6550.

[31] D. K. Pai, S. Sueda, and Q. Wei, “Simulation of 3D neuro-musculo-skeletal systems with contact,” in *Proc. Adv. Comput. Motor Control III. Symp. Soc. Neurosci. Meeting*, 2004.

[32] M. Bergou, M. Wardetzky, S. Robinson, B. Audoly, and E. Grinspun, “Discrete elastic rods,” *ACM Trans. Graph.*, vol. 27, no. 3, pp. 1–12, 2008.

[33] G. Kirchhoff, “Ueber das gleichgewicht und die bewegung eines unendlich dünnen elastischen stabes,” *J. Für Die Reine Und Angewandte Mathematik*, vol. 56, pp. 285–313, 1859.

[34] B. Angles *et al.*, “Viper: Volume invariant position-based elastic rods,” *Proc. ACM Comput. Graph. Interactive Techn.*, vol. 2, no. 2, pp. 1–26, 2019.

[35] S. H. Sadati *et al.*, “TMTDyn: A matlab package for modeling and control of hybrid rigid-continuum robots based on discretized lumped systems and reduced-order models,” *Int. J. Robot. Res.*, 2019, Art. no. 0278364919881685.

[36] C. Armanini, F. Dal Corso, D. Misseroni, and D. Bigoni, “From the elastica compass to the elastica catapult: An essay on the mechanics of soft robot arm,” *Proc. Roy. Soc. A*, vol. 473, no. 2198, 2017, Art. no. 20160870.

[37] F. Connolly, C. Walsh, and K. Bertoldi, “Automatic design of fiber-reinforced soft actuators for trajectory matching,” *Proc. Natl. Acad. Sci.*, vol. 114, no. 1, pp. 51–56, 2017.

[38] G. Cicconofri and A. DeSimone, “A study of snake-like locomotion through the analysis of a flexible robot model,” *Proc. Roy. Soc. A*, vol. 471, no. 2184, 2015, Art. no. 20150054.

[39] M. Mahvash and P. Dupont, “Stiffness control of surgical continuum manipulators,” *IEEE Trans. Robot.*, vol. 27, no. 2, pp. 334–345, Apr. 2011.

[40] W. Huang, X. Huang, C. Majidi, and M. K. Jawed, “Dynamic simulation of articulated soft robots,” *Nat. Commun.*, vol. 11, no. 1, pp. 1–9, 2020.

[41] E. Cosserat and F. Cosserat, *Théorie des corps déformables*. Paris, France: Cornell University Library, 1909.

[42] V. Mnih *et al.*, “Human-level control through deep reinforcement learning,” *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015.

[43] T. P. Lillicrap *et al.*, “Continuous control with deep reinforcement learning,” in *Proc. 4th Int. Conf. Learn. Representations*, 2016.

[44] J. Schulman, S. Levine, P. Moritz, M. I. Jordan, and P. Abbeel, “Trust region policy optimization,” in *Proc. Int. Conf. Mach. Learn.*, pp. 1889–1897, 2015.

[45] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms,” 2017, *arXiv:1707.06347*.

[46] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, “Soft Actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor,” in *Proc. Int. Conf. Mach. Learn.*, pp. 1861–1870, 2018.

[47] S. Fujimoto, H. van Hoof, and D. Meger, “Addressing function approximation error in actor-critic methods,” in *Proc. Int. Conf. Mach. Learn.*, pp. 1587–1596, 2018.