Deep Reinforcement Learning in Soft Viscoelastic Actuator of Dielectric Elastomer

Lu Li[1,*], Junnan Li[2,*], Lei Qin[1], Jiawei Cao[1], Mohan S. Kankanhalli[3], *Fellow, IEEE*, Jian Zhu[1]

***Abstract*—Dielectric elastomer actuators (DEAs) have been widely employed as artificial muscles in soft robots. Due to material viscoelasticity and nonlinear electromechanical coupling, it is challenging to accurately model a viscoelastic DEA, especially when the actuator is of a complex or irregular configuration. Control of DEAs is thus challenging but significant. In this paper, we propose a model-free method for control of DEAs, based on deep reinforcement learning. We perform dynamic feedback control by considering the time-dependent behavior of DEAs. Our method is generic in that it does not require task-specific knowledge about the structure or material parameters of the DEA. The experiments show that our method is robust to achieve accurate control for the DEAs of different configurations, different prestretches, and at different times (the material property usually changes due to viscoelasticity effects). To the best of our knowledge, this work is the first effort to explore deep reinforcement learning for control of DEAs.**

***Index Terms*—Modeling, Control, and Learning for Soft Robots; Model Learning for Control**

I. Introduction

Soft actuators, with intrinsic flexibility and tendency to conform to the external environment, have emerged as an important research field in the last decade. Built of soft materials, soft actuators/robots can adapt their shape to various environments, which enables them to perform effectively in unstructured or human-centered environments, where safety is the primary concern [1], [2].

Many impressive soft actuators have been developed in the literature, such as pneumatic and fluidic elastomeric actuators [3], [4], shape memory alloys [5], [6], and ionic polymer-metal composites (IPMCs) [7], etc. Among these soft actuators, dielectric elastomer actuators (DEAs), known as artificial muscles, are of particular interest owing to their unique attributes, including fast response, large voltage-induced deformation, and quiet operation [8], [9]. A DEA consists of a thin membrane of elastomer, sandwiched between two compliant electrodes. When subject to voltage, the elastomer undergoes Maxwell stress and then generates a controllable deformation [10]. Different designs of DEAs have been explored to develop soft robots, such as jellyfish robot [11], hexapod robot [12], and artificial muscle [13]. A detailed survey on DEAs can be found in [10].

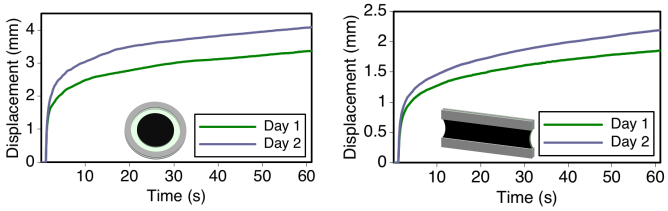


Fig. 1. DEAs exhibit non-linear time-dependent behaviors. A step voltage of 3kV and 5kV is applied to a circular DEA (left) and a rectangle DEA (right), respectively. In short-term (0 to 60 seconds), both DEAs demonstrate non-linear displacement increase. Next, we apply the same step voltage after 24 hours, the voltage-induced behavior of Day 2 is different from that of Day 1, which indicates a long-term time-dependent change in material property, due to viscoelasticity.

Despite their wide applications, control of DEAs is challenging, due to the following reasons. First, DEAs are usually made of viscoelastic materials and can exhibit complex time-dependent behaviors, such as creep and hysteresis [14], [15], [16]. Creep refers to the phenomenon where the displacement of a DEA drifts with time, and hysteresis is the phenomenon where the electromechanical response of a DEA is different during unloading as compared to loading. Second, the Maxwell stress is related to the applied electric field, which directly depends on the deformation of the actuators (i.e., thickness of the membrane). Consequently, the actual actuation, based on the electromechanical coupling, can be strongly nonlinear and time-dependent [17]. Figure 1 shows the displacement of two DEAs (the left for a circular DEA, and the right for a rectangle DEA) as a function of time, when they are subject to step voltage. The DEAs exhibit nonlinear time-dependent behaviors, which can be short-term (within 60 seconds) or long-term (across 24 hours), due to viscoelasticity effects.

Researchers have attempted to study the static and dynamic characterization [18], [19] and modeling of the physical mechanism underlying the material behavior of DEAs [20], [21], [16]. However, all these models are only valid for DEAs of simple structures, based on assumption of homogeneous deformation. Furthermore, the accuracy of these models highly

Manuscript received: September, 10, 2018; Revised January, 3, 2019; Accepted January, 31, 2019.

This paper was recommended for publication by Editor Kyu-Jin Cho upon evaluation of the Associate Editor and Reviewers' comments. This work was supported in part by MOE Tier 1, Singapore under Grant R-265-000-609-114, in part by ASTAR, Singapore under Grant R-265-000-629-305, and in part by the National Research Foundation, Prime Minister's Office, Singapore under its Strategic Capability Research Centres Funding Initiative.

[1]Lu Li, Lei Qin, Jiawei Cao and Jian Zhu are with the Department of Mechanical Engineering, National University of Singapore, Singapore mpelilu@nus.edu.sg; qinlei@u.nus.edu; jiaweic@u.nus.edu; mpezhuj@nus.edu.sg

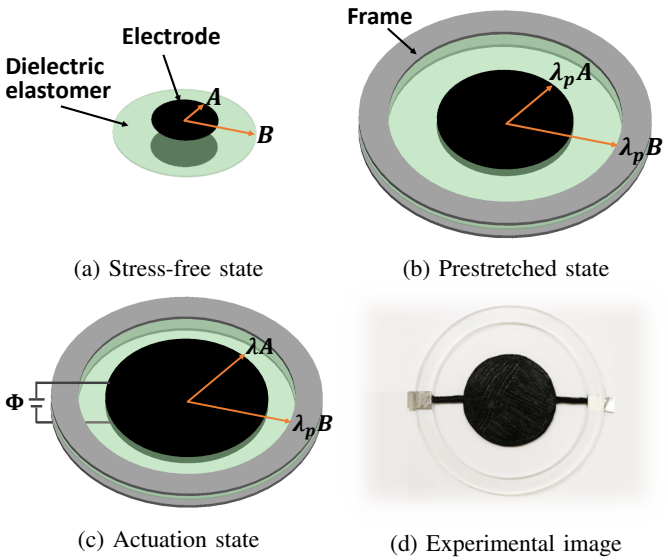
[2]Junnan Li is with the Graduate School of Integrative Science and Engineering, National University of Singapore, Singapore lijunnan@u.nus.edu

[3]Mohan S. Kankanhalli is with the School of Computing, National University of Singapore, Singapore mohan@comp.nus.edu.sg

*Equal contribution.

Digital Object Identifier (DOI): see top of this page.

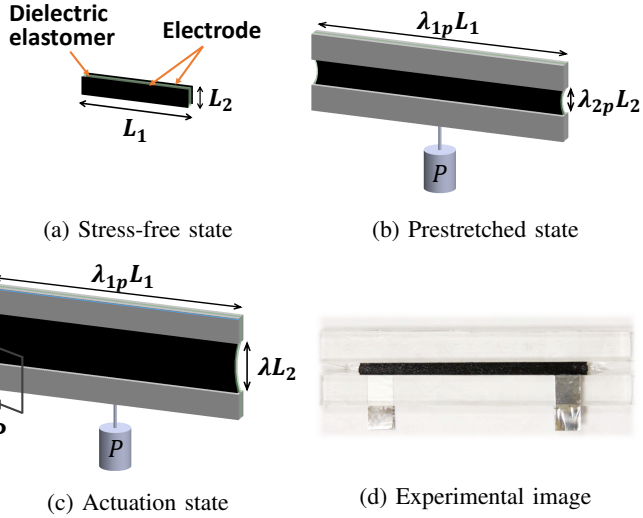
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LRA.2019.2898710, IEEE Robotics and Automation Letters



(a) Stress-free state (b) Prestretched state

(c) Actuation state (d) Experimental image

Fig. 2. Fabrication process of circular DEA.



(a) Stress-free state (b) Prestretched state

(c) Actuation state (d) Experimental image

Fig. 3. Fabrication process of rectangle DEA.

depends on the fitting parameters of soft materials. Some researchers have also explored feedback control of DEAs. Some works investigated solutions based on PID control [22], [23], [9], [24], while others explored nonlinear control methods such as internal model control with gain scheduling [25], adaptive feedforward control [26], cerebellar-inspired control [27], and fuzzy logic control [28]. However, it should be noted that these control methods require detailed material parameters and rely on substantial heuristic design and tuning to specific DEAs, which greatly limits their potential in practical applications.

In this work, we propose a generic model-free method for accurate time-series control of DEAs. Based on a recently proposed deep reinforcement learning (RL) algorithm, our method takes advantage of off-policy training of deep Q-functions [29], [30]. The method has the following advantages. First, it does not require any task-specific domain knowledge, such as the structure, design or material parameters of DEAs. Second, the method formulates DEA control as a time-continuous problem, and learns a nonlinear and time-dependent control policy that better addresses the viscoelasticity effects. Third, unlike previous methods that tune the parameters on test trajectories, our method can learn to accurately control new test trajectories unseen during training. Last but not least, the method can quickly adapt to changes in material property or different actuator structures, by leveraging knowledge learned from past experiences. To the best of our knowledge, this work is the first effort to explore deep reinforcement learning for control of DEAs.

The rest of the paper is organized as follows. In Section II, we introduce the background on RL. In Section III, we describe the DEA designs, and delineate our control method. We show both quantitative and qualitative experimental results in Section IV. Section V concludes the paper.

II. Background on Reinforcement Learning

In this section, we will introduce the reinforcement learning (RL) problem, and describe existing algorithmic foundations on which we build our method.

Reinforcement learning has been widely applied in robotic tasks [31] that involves time-continuous control. In reinforcement learning, the goal is to control an agent (e.g. robot, actuator), so as to maximize a reward function which, in the context of robotics, is the task that the robot attempts to accomplish. Reinforcement learning is formulated as a Markov Decision Process (MDP) [32], where the next state of the agent depends on its current state and its action. At each timestep t, the agent in state $\boldsymbol{x}_t$ executes an action $\boldsymbol{a}_t$ following a policy $\pi(\boldsymbol{a}_t|\boldsymbol{x}_t)$, transitions to the next state x_{t+1}, and receives a reward $r(\boldsymbol{x}_t, \boldsymbol{a}_t)$. Here, we consider a finite horizon task, and define return $R_t = \sum_{i=t}^{T} \gamma^{i-t} r(\boldsymbol{x}_t, \boldsymbol{a}_t)$, which is the γ-discounted, cumulative reward from time t to T. The goal is to find the optimal policy $\pi^\star$ that maximizes the expected sum of returns $R = \mathbb{E}_\pi[R_1]$ given the initial state.

Q-learning is an off-policy model-free reinforcement learning algorithm favorable for robotic applications because of its data efficiency. It learns a greedy deterministic policy $\pi(\boldsymbol{a}_t|\boldsymbol{x}_t) = \delta(\boldsymbol{a}_t = \boldsymbol{\mu}(\boldsymbol{x}_t))$ by iterating between learning the Q-function corresponding to a policy, $Q^\pi(\boldsymbol{x}_t, \boldsymbol{a}_t) = \mathbb{E}_{r_{i\geq t}, \boldsymbol{x}_{i\geq t}, \boldsymbol{a}_{i\geq t} \sim \pi}[R_t|\boldsymbol{x}_t, \boldsymbol{a}_t]$, and updating the policy by greedily maximizing the Q-function, $\boldsymbol{\mu}(\boldsymbol{x}_t) = \arg\max_a Q^\pi(\boldsymbol{x}_t, \boldsymbol{a}_t)$. Let θ^Q parametrize the action-value function, and β be an arbitrary exploration policy, the learning objective is to minimize the Bellman error $L(\theta^Q)$, where we fix the target y_t:

$$L(\theta^Q) = \mathbb{E}_{\boldsymbol{x}_t, \boldsymbol{a}_t \sim \beta, \boldsymbol{x}_{t+1}, r_t}[(Q(\boldsymbol{x}_t, \boldsymbol{a}_t|\theta^Q) - y_t)^2]$$
$$y_t = r(\boldsymbol{x}_t, \boldsymbol{a}_t) + \gamma Q(\boldsymbol{x}_{t+1}, \boldsymbol{\mu}(\boldsymbol{x}_{t+1}))$$

In our method, the Q-function is parametrized by a neural network. We also employ the *experience replay* [29] strategy in our training, that allows past real-word samples to be used repeatedly.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LRA.2019.2898710, IEEE Robotics and Automation Letters

III. DEA Control Method

A. Actuator Overview

In this work, we study two commonly used DEAs, namely circular DEA [33], [34] and rectangle DEA [17], [16]. Figure 2 shows the fabrication process and working mechanism of the circular DEA. It consists of an active region covered with compliant electrodes and a passive region without electrodes. At the stress-free state, a VHB4910 membrane with radius $B = 25$mm is sandwiched by compliant electrodes (carbon grease) with radius $A = 15$mm. At the prestretched state, the elastomer is stretched equal-biaxially by $\lambda_p = 4$, and secured by two annular frames with inner radius of 60mm. When subject to high voltage (actuation state), the active region expands to radius λA at the expense of the passive region. The experimental image is shown in Figure 2(d).

The working mechanism of the rectangle DEA is illustrated in Figure 3. At the stress-free state, a VHB membrane of horizontal width L_1 and vertical length L_2 is sandwiched by compliant electrodes. At the prestretched state, the elastomer is horizontally stretched by $\lambda_{1p} = 5$, fixed to rigid clamps, and vertically stretched by $\lambda_{2p} = 2$, due to a constant mechanical force P. When voltage is applied, the vertical length expands to λL_2. The experimental image is shown in Figure 3(d).

In our experiment, a laser sensor (ILD 1700, MicroEpsilon) is employed to measure the radial displacement $(\lambda - \lambda_p)A$ for the circular DEA (see Figure 2), and the vertical displacement $(\lambda - \lambda_{2p})L_2$ for the rectangle DEA (see Figure 3). We measure the displacement every 0.1 seconds, and apply voltage to adaptively control the actuator. In order to prevent dielectric break-down failure, the input voltages for the circular and rectangle DEA are bound to the range $[0, 4]$ kV and $[0, 5.5]$ kV, respectively.

B. RL for DEA Control

In DEA control, the objective is to apply a sequence of voltage, $V = \{v_t\}_{t=1}^T$, such that the actuator achieves certain displacement, $D = \{d_t\}_{t=1}^T$, which closely follows a target displacement sequence, $Y = \{y_t\}_{t=1}^T$. A schematic of the control process is shown in Figure 4. At each timestep t (every 0.1 second), the RL agent receives an observation of the environment in the form of the state vector $\boldsymbol{x}_t$, based on which it executes an action a_t that determines the control voltage v_t applied to the actuator. First, we define the state $\boldsymbol{x}_t$ as

$$\boldsymbol{x}_t = \{t, d_t, v_{t-1}, y_{t+1}, y_{t+1+\tau}, y_{t+1+2\tau}, \hat{v}_t\}. \quad (1)$$

$\boldsymbol{x}_t$ contains both the actuator's current status (i.e. current time t, displacement d_t), its voltage v_{t-1} at the previous timestep, and its future target displacement. Because of viscoelasticity of DEA material, voltage has a long-term impact beyond the current timestep. Therefore, the agent needs to consider future target displacement $\{y_{t+1+\tau}, y_{t+1+2\tau}\}$ when performing an action. τ controls how far in the future the agent plans for. We experiment with different values of τ and find $\tau = 5$ gives the best performance, which allows the agent to plan for the target displacement 10 timesteps (1 second) ahead.

$\hat{v}_t$ is an estimated value of the control voltage that can achieve displacement $d_{t+1} \approx y_{t+1}$. We calculate $\hat{v}$ with the following steps. First, we apply N step voltages $\{V\}_{n=1}^N$ with different magnitudes to the actuator. For each V_n with magnitude u_n, the actuator produces a displacement sequence $\{d_t^n\}_{t=1}^T$. (Figure 1 shows examples of step-voltage displacement.) Then for each timestep t, we fit a 4-th order polynomial function $v = P_t(d)$ to a set of experimental data points $\{(u^1, d_t^1), (u^2, d_t^2), ..., (u^n, d_t^n)\}$. P_t models an approximate mapping from displacement to its control voltage. Finally, we substitute the target displacement at $t+1$ into P_t to get the estimated voltage, $\hat{v}_t = P_t(y_{t+1})$. Figure 5 shows an example of fitting P_t and using $\{\hat{v}_t\}_{t=1}^T$ to control a circular DEA.

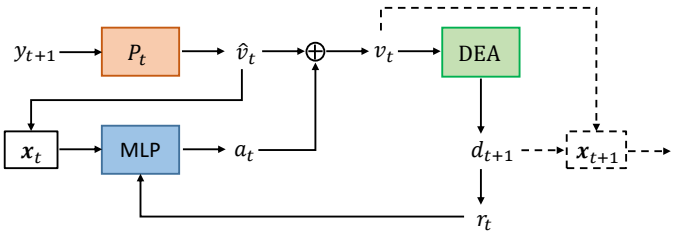
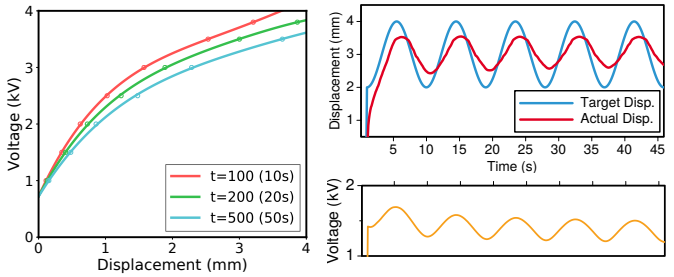


Fig. 4. Schematic of the proposed control process. $\boldsymbol{x}_t$ is the state defined in equation (1). The multilayer perceptron (MLP) takes $\boldsymbol{x}_t$ as input and outputs an action a_t, which determines an offset added to the approximated voltage $\hat{v}_t$. The control voltage v_t drives the DEA to the displacement d_{t+1}. Then the RL agent calculates the reward r_t, and transitions to the next state $\boldsymbol{x}_{t+1}$.



(a) (b)

Fig. 5. (a) Examples of the fitted P_t with $t = 100, 200, 500$, for a circular DEA. (b) Results of using the approximate voltage $\hat{v}_t = P_t(y_{t+1})$ (shown in orange) to control the actuator. The target displacement sequence is a sine wave.

We aim to learn an offset voltage added to the approximated control voltage $\hat{v}_t$ for more accurate control. In our RL formulation, we define a discrete action space $\mathcal{A}$, which contains 21 actions. Each action defines an offset value a_t that is added to $\hat{v}_t$. The first 10 actions are negative offsets ranging from -0.04 to -0.4, with an equal step size of 0.04. The rest 11 actions are non-negative offsets ranging from 0 to 0.1, with an equal step size of 0.01. Therefore, $\mathcal{A} = \{-0.4, -0.36, ..., -0.04, 0, 0.01, 0.02, ..., 0.1\}$. At each timestep t, the agent executes an action $a_t \in \mathcal{A}$ according to the Q-function. The action decides the voltage applied to the actuator: $v_t = a_t + \hat{v}_t$.

After executing an action, the agent transitions to the next state $\boldsymbol{x}_{t+1}$ with displacement d_{t+1}. The reward at timestep t is defined as the absolute error between d_{t+1} and the target displacement y_{t+1}: $r_t = |d_{t+1} - y_{t+1}|$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LRA.2019.2898710, IEEE Robotics and Automation Letters

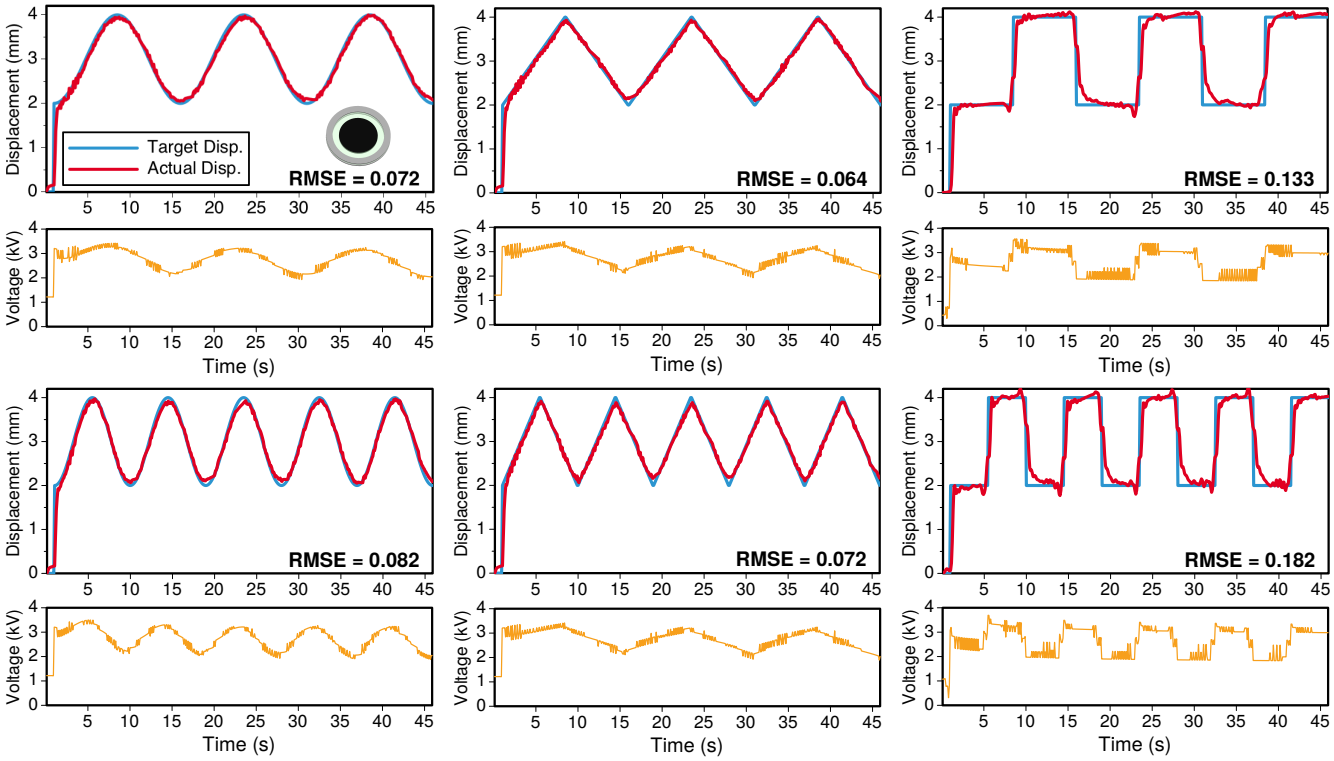


Fig. 6. Results using the proposed RL method to control a circular DEA. Test displacement trajectories include sine waves, triangle waves and square waves, of two different frequencies. The actual displacements (shown in red) are close to the target displacements (shown in blue). The control voltage for each trajectory is shown in orange.

C. Network

In our RL control method, the Q-function $\boldsymbol{\mu}(\boldsymbol{x})$ is parametrized by a multilayer perceptron (MLP). We use a two-hidden-layer network with 50 units each. ReLU is used as hidden activations, and Batch Normalization is applied after each hidden layer. A fully-connected layer followed by Softmax layer is added after the second hidden layer, which generates a probability distribution over possible actions. At each timestep, the action with the highest probability is taken.

During training, we generate random displacement sequences Y_{train} for the actuator to follow. We store each training transition $(\boldsymbol{x}_t, a_t, r_t, \boldsymbol{x}_{t+1})$ into a replay buffer [29], and sample mini-batches of size 100 from the replay buffer to train the network. Network training is decoupled from transition collecting, which allows the actuator to be controlled in real time, without experiencing delays due to computational cost of back-propagation. We employ ϵ-greedy exploration strategy with a decaying ϵ from 0.9 to 0.05. The network is trained using RMSprop [35] optimizer with an initial learning rate of 0.01, which decays to 0.0001 after 100 epochs. In addition, we use the target network [29] to stabilize training. We will release our code for the community.

IV. Experiments

In this section, we conduct experiments using the circular DEA and the rectangle DEA as described in Section III-A. In order to test the performance of the proposed method, we design six common displacement trajectories, including sine waves, triangle waves, and square waves, of two different frequencies. In order to show the generalizability of our method, all of the test trajectories are not included in the training examples.

Each of the test trajectories is of 45 seconds long, which corresponds to 450 timesteps. At each training epoch, we generate a random training trajectory of the same length (45 seconds). Then we wait for another 40 seconds after every trajectory, which allows the actuator to fully reset to its initial state. Hence, each epoch lasts 85 seconds. We train our network for 100 epochs, and perform test every 5 epochs. We use the average RMSE across all test trajectories to quantitatively measure the performance of our model.

A. Results

Figure 6 and Figure 7 show the qualitative test results for circular DEA and rectangle DEA, respectively. For sine wave and triangle wave, the actual displacements closely follow the target displacements. The square waves are more challenging because of the abrupt slope. Nevertheless, because our model learns to plan for future displacements, it can achieve relatively sharp increase (or decrease) without overshooting.

In Figure 8, we show the average test RMSE as training progresses. For both circular DEA and rectangle DEA, the test error has a decreasing trend. The error rapidly decreases at the initial stage of training, and converges to a small value in later epochs.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LRA.2019.2898710, IEEE Robotics and Automation Letters

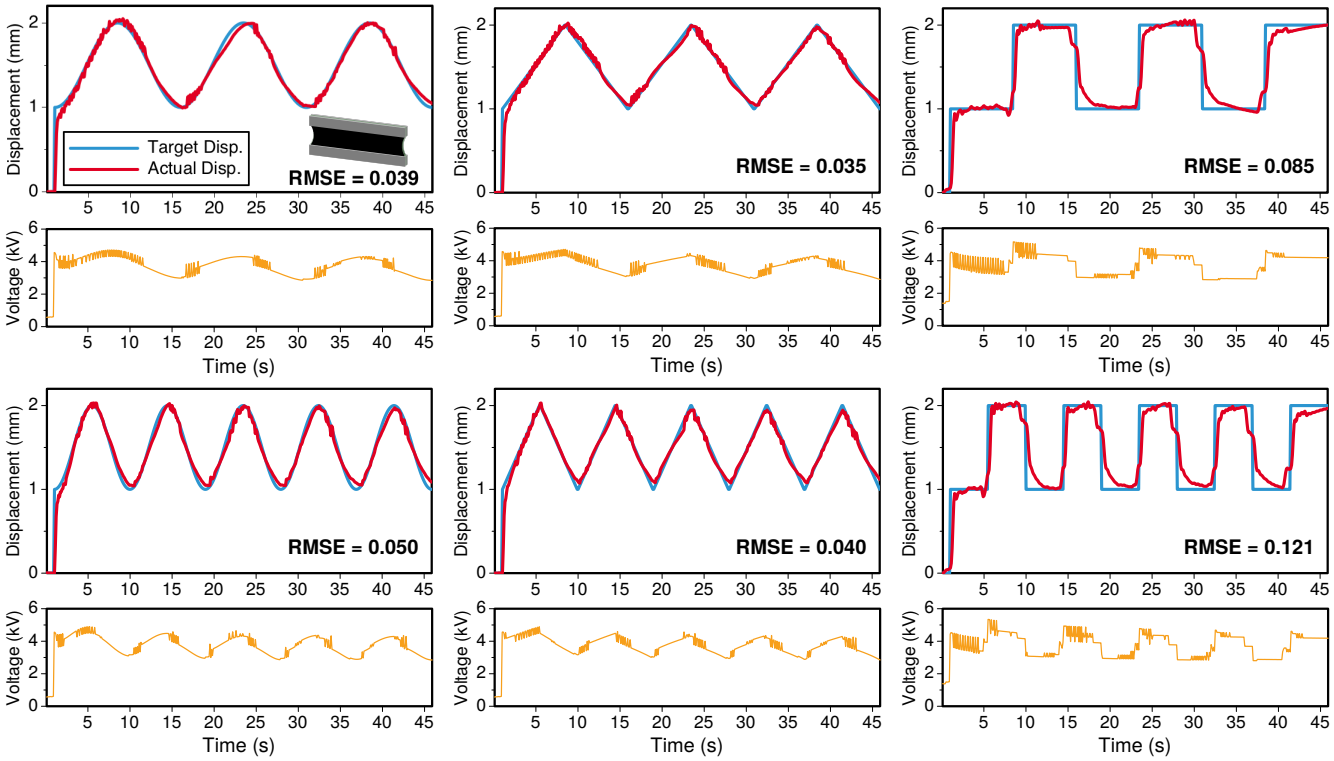


Fig. 7. Results using the proposed RL method to control a rectangle DEA. Test displacement trajectories include sine waves, triangle waves and square waves, of two different frequencies. The actual displacements (shown in red) are close to the target displacements (shown in blue). The control voltage for each trajectory is shown in orange.

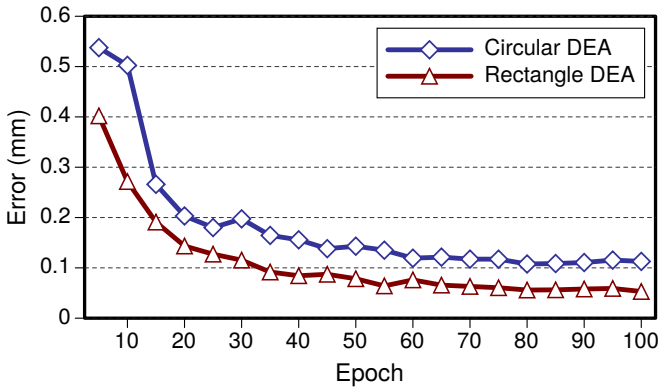


Fig. 8. Average test RMSE for circular DEA and rectangle DEA.

B. Comparison with baseline controller

To validate the performance of the proposed controller, a conventional proportional-integral (PI) controller is implemented for comparison [36]. Since the derivative action is sensitive to noise, we use PI instead of PID controller to ensure the stability of the actuator during operation in the case of noisy measurement. The controller is given in the following form:

$$v(t) = K_p(e(t) + \frac{1}{T_i}\int e(t)dt), \tag{2}$$

where K_p is the proportional gain and T_i is the integration time. The controller parameters are manually tuned to optimize the performance on the test trajectories, with $K_p = 5, T_i = 0.002$ for the circular DEA and $K_p = 2, T_i = 0.001$ for the rectangle DEA.

TABLE I: Comparison of RL controller with PI controller. Numbers indicate average RMSE across test trajectories.

	RL Controller	PI Controller
Circular DEA	0.108	0.132
Rectangle DEA	0.061	0.103

Table I shows the comparison between the proposed RL controller and the PI controller. We test the PI controller using the same six test trajectories as the RL controller. For both circular and rectangle DEA, the proposed RL controller has smaller average RMSE compared to the PI controller. Figure 9 shows some qualitative examples. Compared to the proposed method, the PI controller has an undesirable delayed response and severe overshoot for the square wave. Unlike the PI controller which is specifically tuned using test trajectories, the RL controller has no knowledge of the test trajectories. Despite this disadvantage, the RL controller is able to outperform the PI controller by learning control policies from training data, which can generalize to new trajectories.

C. Robustness to material/structural changes

Our proposed RL control method is robust to changes in the material property or structure of the DEA. Here we perform

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LRA.2019.2898710, IEEE Robotics and Automation Letters

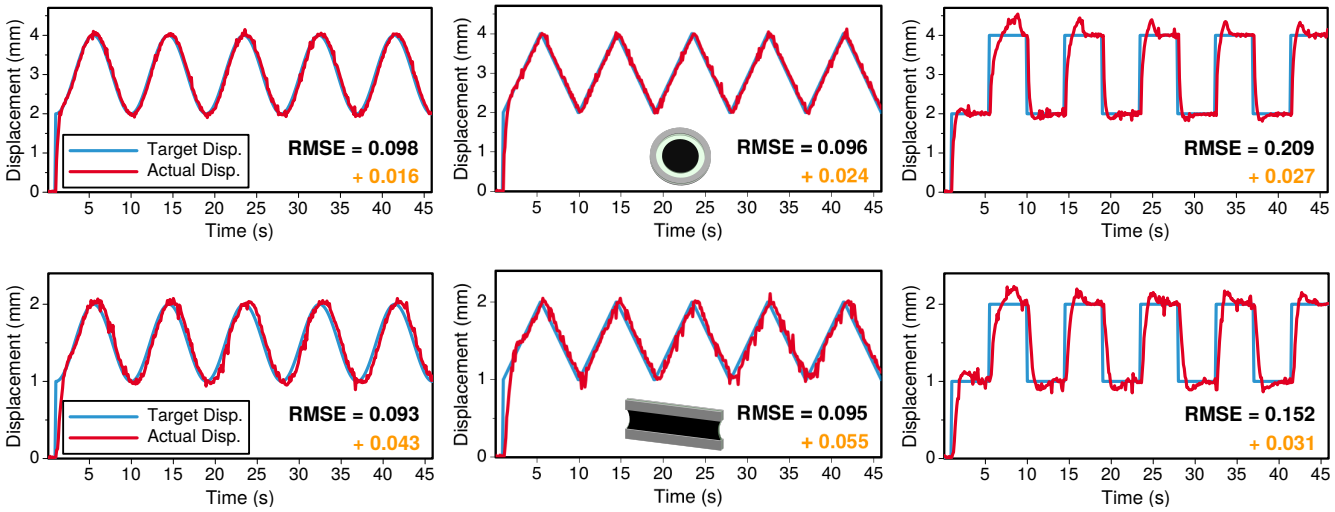
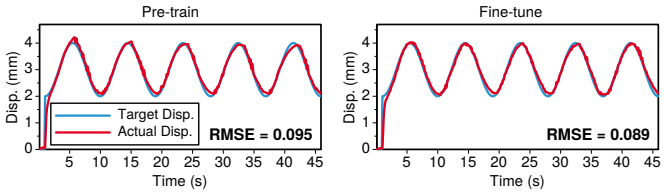


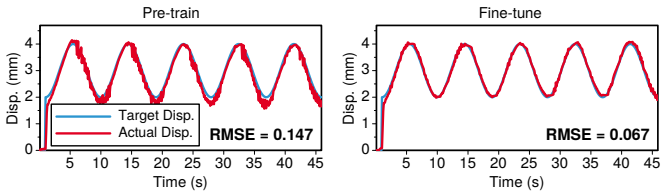
Fig. 9. Results using the PI controller to control the circular DEA (top row) and the rectangle DEA (bottom row). The number in orange shows the increase in RMSE for each trajectory compared with the proposed RL controller.

TABLE II: Test results (average RMSE across test trajectories) after material/structural changes.

Day1 DEA1($\lambda_p = 4$)	Change	pre-train	fine-tune
0.108	*Day2* DEA1($\lambda_p = 4$)	0.131	0.111
	Day1 DEA2($\lambda_p = 5$)	0.316	0.107



(a) *Day2* DEA1



(b) DEA2 ($\lambda_p = 5$)

Fig. 10. The change in (a) time-dependent material property or (b) pre-stretch ratio of a circular DEA affects the control performance. However, the RL controller is robust to such changes by using previously trained model. The controller can quickly adapt to the changes after fine-tuning for 10 epochs.

two sets of experiments using circular DEA to demonstrate the robustness of our method.

First, as shown in Figure 1, DEAs demonstrate a long-term time-dependent behavior caused by viscoelasticity. Therefore, it is important for the control method to be robust to such changes in material property. In Table II, we show that our model trained on Day 1 can quickly adapt to the same actuator on Day 2 (24 hours later). We first directly use the model pre-trained on Day 1 to control the actuator on Day 2, which achieves an acceptable average test RMSE of 0.131 (the test error on Day 1 is 0.108). Then we fine-tune the DEA for only 10 epochs. The error further reduces to 0.111. Figure 10(a) shows an example of sine wave for the pre-trained and fine-tuned model.

As introduced in Section III-A, the prestretch ratio $\lambda_p = 4$ is an important structural design parameter that significantly affects the voltage-induced behavior of a DEA [17]. We show that our method can transfer knowledge learned from a previous DEA, to a new DEA with different λ_p. In this experiment, we fabricate a second circular DEA with $\lambda_p = 5$, which we denote as DEA2. We first use the model pre-trained on DEA1 to directly control DEA2, which has an average test RMSE of 0.316, as shown in Table II. After fine-tuning the pre-trained model on DEA2 for only 10 epochs, the error quickly reduces to 0.107. The learning speed is much faster comparing to training a model on DEA2 from scratch. We show a sine wave example in Figure 10(b).

V. Conclusion

In this paper, we propose a model-free method, based on deep reinforcement learning, for control of DEAs. Our method can achieve dynamic feedback control of DEAs, considering their time-dependent behavior caused by material viscoelasticity. We conduct experiments on two common DEAs (with circular and rectangle configurations) to demonstrate the advantages of the proposed method, such as accurate control and robustness to changes in material property and structure of DEAs.

This work validates the great potential of RL based method to control soft actuators/robots. There are several limitations of the current method, which we intend to address in future work. We will aim to reduce the training time, and improve the controller's performance at higher frequencies. Furthermore, we intend to control soft actuators of more complex structures and accomplish real-world tasks in unstructured environments. For example, building a soft mobile robot which can learn to avoid obstacles.

REFERENCES

[1] C. Majidi, "Soft robotics: a perspective—current trends and prospects for the future," *Soft Robotics*, vol. 1, no. 1, pp. 5–11, 2014.

[2] D. Rus and M. T. Tolley, "Design, fabrication and control of soft robots," *Nature*, vol. 521, no. 7553, p. 467, 2015.

[3] J. Bishop-Moser and S. Kota, "Design and modeling of generalized fiber-reinforced pneumatic soft actuators," *IEEE Trans. Robot*, vol. 31, no. 3, pp. 536–545, 2015.

[4] P. Polygerinos, Z. Wang, J. T. Overvelde, K. C. Galloway, R. J. Wood, K. Bertoldi, and C. J. Walsh, "Modeling of soft fiber-reinforced bending actuators," *IEEE Trans. Robot*, vol. 31, no. 3, pp. 778–789, 2015.

[5] H.-T. Lin, G. G. Leisk, and B. Trimmer, "Goqbot: a caterpillar-inspired soft-bodied rolling robot," *Bioinspiration Biomimetics*, vol. 6, no. 2, p. 026007, 2011.

[6] B. Mazzolai, L. Margheri, M. Cianchetti, P. Dario, and C. Laschi, "Soft-robotic arm inspired by the octopus: Ii. from artificial requirements to innovative technological solutions," *Bioinspiration Biomimetics*, vol. 7, no. 2, p. 025005, 2012.

[7] I. Must, F. Kaasik, I. Põldsalu, L. Mihkels, U. Johanson, A. Punning, and A. Aabloo, "Ionic and capacitive artificial muscle for biomimetic soft robotics," *Advanced Engineering Materials*, vol. 17, no. 1, pp. 84–94, 2015.

[8] P. Brochu and Q. Pei, "Advances in dielectric elastomers for actuators and artificial muscles," *Macromolecular rapid communications*, vol. 31, no. 1, pp. 10–36, 2010.

[9] G. Rizzello, D. Naso, A. York, and S. Seelecke, "Modeling, identification, and control of a dielectric electro-active polymer positioning system," *IEEE Trans. Contr. Sys. Techn.*, vol. 23, no. 2, pp. 632–643, 2015.

[10] G.-Y. Gu, J. Zhu, L.-M. Zhu, and X. Zhu, "A survey on dielectric elastomer actuators for soft robots," *Bioinspiration Biomimetics*, vol. 12, no. 1, p. 011003, 2017.

[11] H. Godaba, J. Li, Y. Wang, and J. Zhu, "A soft jellyfish robot driven by a dielectric elastomer actuator," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 624–631, 2016.

[12] C. T. Nguyen, H. Phung, H. Jung, U. Kim, T. D. Nguyen, J. Park, H. Moon, J. C. Koo, and H. R. Choi, "Printable monolithic hexapod robot driven by soft actuator," in *ICRA*, 2015, pp. 4484–4489.

[13] Y. Wang and J. Zhu, "Artificial muscles for jaw movements," *Extreme Mechanics Letters*, vol. 6, pp. 88–95, 2016.

[14] J. Bergström and M. Boyce, "Constitutive modeling of the large strain time-dependent behavior of elastomers," *Journal of the Mechanics and Physics of Solids*, vol. 46, no. 5, pp. 931–954, 1998.

[15] K. M. Schmoller and A. R. Bausch, "Similar nonlinear mechanical responses in hard and soft materials," *Nature materials*, vol. 12, no. 4, p. 278, 2013.

[16] G.-Y. Gu, U. Gupta, J. Zhu, L.-M. Zhu, and X. Zhu, "Modeling of viscoelastic electromechanical behavior in a soft dielectric elastomer actuator," *IEEE Trans. Robot*, vol. 33, no. 5, pp. 1263–1271, 2017.

[17] M. Kollosche, J. Zhu, Z. Suo, and G. Kofod, "Complex interplay of nonlinear processes in dielectric elastomers," *Physical Review E*, vol. 85, no. 5, p. 051801, 2012.

[18] J.-S. Plante and S. Dubowsky, "On the performance mechanisms of dielectric elastomer actuators," *Sensors and Actuators A: Physical*, vol. 137, no. 1, pp. 96–109, 2007.

[19] A. York, J. Dunn, and S. Seelecke, "Experimental characterization of the hysteretic and rate-dependent electromechanical behavior of dielectric electro-active polymer actuators," *Smart Materials and Structures*, vol. 19, no. 9, p. 094014, 2010.

[20] G. Berselli, R. Vertechy, M. Babič, and V. Parenti Castelli, "Dynamic modeling and experimental evaluation of a constant-force dielectric elastomer actuator," *Journal of Intelligent Material Systems and Structures*, vol. 24, no. 6, pp. 779–791, 2013.

[21] M. Hodgins, G. Rizzello, D. Naso, A. York, and S. Seelecke, "An electro-mechanically coupled model for the dynamic behavior of a dielectric electro-active polymer actuator," *Smart Materials and Structures*, vol. 23, no. 10, p. 104006, 2014.

[22] S. Xie, P. Ramson, D. Graaf, E. Calius, and I. Anderson, "An adaptive control system for dielectric elastomers," in *Proc. IEEE Int. Conf. Ind. Technol*, 2005, pp. 335–340.

[23] K. Yun and W.-j. Kim, "Microscale position control of an electroactive polymer using an anti-windup scheme," *Smart materials and structures*, vol. 15, no. 4, p. 924, 2006.

[24] G. Rizzello, D. Naso, B. Turchiano, and S. Seelecke, "Robust position control of dielectric elastomer actuators based on lmi optimization." *IEEE Trans. Contr. Sys. Techn.*, vol. 24, no. 6, pp. 1909–1921, 2016.

[25] R. W. Jones and R. Sarban, "Inverse grey-box model-based control of a dielectric elastomer actuator," *Smart Materials and Structures*, vol. 21, no. 7, p. 075019, 2012.

[26] R. Sarban and R. W. Jones, "Physical model-based active vibration control using a dielectric elastomer actuator," *Journal of Intelligent Material Systems and Structures*, vol. 23, no. 4, pp. 473–483, 2012.

[27] E. D. Wilson, T. Assaf, M. J. Pearson, J. M. Rossiter, S. R. Anderson, and J. Porrill, "Bioinspired adaptive control for artificial muscles," in *Conference on Biomimetic and Biohybrid Systems*, 2013, pp. 311–322.

[28] C. M. Druitt and G. Alici, "Intelligent control of electroactive polymer actuators based on fuzzy and neurofuzzy methodologies," *IEEE/ASME Transactions on Mechatronics*, vol. 19, no. 6, pp. 1951–1962, 2014.

[29] 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," in *International Conference on Learning Representations*, 2016.

[30] S. Gu, T. Lillicrap, I. Sutskever, and S. Levine, "Continuous deep q-learning with model-based acceleration," in *International Conference on Machine Learning*, 2016, pp. 2829–2838.

[31] 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.

[32] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[33] S. J. A. Koh, T. Li, J. Zhou, X. Zhao, W. Hong, J. Zhu, and Z. Suo, "Mechanisms of large actuation strain in dielectric elastomers," *Journal of Polymer Science Part B: Polymer Physics*, vol. 49, no. 7, pp. 504–515, 2011.

[34] L. Qin, Y. Tang, U. Gupta, and J. Zhu, "A soft robot capable of 2d mobility and self-sensing for obstacle detection and avoidance," *Smart Materials and Structures*, vol. 27, no. 4, p. 045017, 2018.

[35] G. Hinton, N. Srivastava, and K. Swersky, "Lecture 6d-a separate, adaptive learning rate for each connection," *Slides of Lecture Neural Networks for Machine Learning*, 2012.

[36] K. H. Ang, G. Chong, and Y. Li, "PID control system analysis, design, and technology," *IEEE Trans. Contr. Sys. Techn.*, vol. 13, no. 4, pp. 559–576, 2005.