Open Loop Position Control of Soft Continuum Arm Using Deep Reinforcement Learning

Sreeshankar Satheeshbabu[1], Naveen Kumar Uppalapati[1], Girish Chowdhary[2] and Girish Krishnan[3]

***Abstract*— Soft robots undergo large nonlinear spatial deformations due to both inherent actuation and external loading. The physics underlying these deformations is complex, and often requires intricate analytical and numerical models. The complexity of these models may render traditional model-based control difficult and unsuitable. Model-free methods offer an alternative for analyzing the behavior of such complex systems without the need for elaborate modeling techniques. In this paper, we present a model-free approach for open loop position control of a soft spatial continuum arm, based on deep reinforcement learning. The continuum arm is pneumatically actuated and attains a spatial workspace by a combination of unidirectional bending and bidirectional torsional deformation. We use Deep-Q Learning with experience replay to train the system in simulation. The efficacy and robustness of the control policy obtained from the system is validated both in simulation and on the continuum arm prototype for varying external loading conditions.**

I. INTRODUCTION

Soft robots have gained significant attention mainly due to their adaptability and safe operation especially involving humans [1]. Their attributes mainly stem from the compliant nature of their structural embodiment that is able to conform to different shapes and absorb rather than transmit impact loads. As a result soft robots may find applications in locomotion or navigation of unstructured terrain [2], human assistive or wearable devices [3] and manipulators and end effectors that handle delicate items [4]. However, the flexibility and softness of these robots leads to highly challenging control and modeling problems.

In this paper, we investigate the control of a unique soft continuum arm known as the BR^2 manipulator [5], [6]. Unlike several other existing soft manipulators, the BR^2 has a completely parallel architecture from an asymmetric combination of soft pneumatic building blocks known as FREEs (Fiber Reinforced Elastomeric Enclosures) that bend and rotate simultaneously, the combination of which yields a spatial deformation pattern as shown in Fig. 1. The advantage of this architecture is its ability to bend spatially to avoid obstacles, and use its whole arm towards grasping long and slender objects by spirally deforming around them, while still maintaining a parallel architecture. In contrast, existing state of the art designs [7], [8] use serial combination of symmetric building blocks [9] that may lead to complex valve routing and controls. The advantages associated with the design is promising for agricultural applications such as autonomous berry harvesting, weeding, and pruning. Our preliminary investigations [5], [6] of the manipulator reveal large nonlinear elastic deformation, with stiffness varying as a function of applied pressure. Furthermore, there exists elastic coupling between the different actuators, and sizable attenuation in the end effector workspace due to external loads or disturbances. This makes physics-based modeling and control of the manipulator extremely challenging.

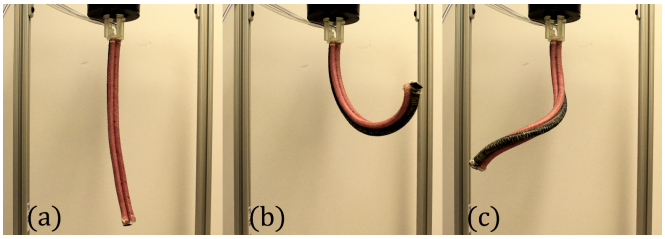

Fig. 1. Deformation modes available to a BR^2 soft manipulator. (a) Home position at zero pressure, (b) Bending FREE pressurized and (c) Bending and rotating FREE pressurized

Control of soft continuum arms have been widely explored through both model based and model free methods [10]. Traditional control methods are founded on first-principles based on mathematical models and use of analytical equations to capture the exact behavior within bounding assumptions. Escande et al. [11] used an exact model for forward kinematic calibration of compact bionic handling assistant (CBHA) based on constant curvature continuum robot theory. Godage et al. [12] employed modal shape functions to develop kinematics for multisection continuum arms. Renda et al. [13] used a geometrically exact steady-state model to capture the behavior of an octopus inspired soft manipulator. Mathematical models have also been used to estimate dynamic characteristics of soft systems as showcased by [14], [15], [16]. Other notable examples include the OctArm [17], Air-OCTOR [8] and DDU [18]. Mathematical models achieve great accuracy and stability as long as the the assumptions made while deriving their governing equations remain valid. As with other continuum manipulators [17], constant curvature assumptions lead to large tracking errors for the BR^2 continuum arm due to offsets from gravity loads.

[1]Sreeshankar Satheeshbabu and Naveen Kumar Uppalapati are PhD candidates in the Industrial and Systems Engineering Department, University of Illinois at Urbana-Champaign, 104 S Mathews Ave, Urbana, IL 61801 sthshbb2@illinois.edu, uppalap2@illinois.edu

[2]Girish Chowdhary is an Assistant Professor in the Department of Agricultural and Biological Engineering and the Coordinated Science Lab (CSL), University of Illinois at Urbana-Champaign, 1304 W. Pennsylvania Urbana Illinois 61801 girishc@illinois.edu

[3]Girish Krishnan is an Assistant Professor in the Industrial and Systems Engineering Department, University of Illinois at Urbana-Champaign, 104 S Mathews Ave, Urbana, IL 61801 gkrishna@illinois.edu

A geometrically exact model that use Cosserat rods may be accurate for analysis, but is numerically complex, and unsuitable for model based control [19].

The realm of model free methods is dominated by machine learning techniques. Both supervised and reinforcement learning have been used to address control problems of varying levels of difficulty. Supervised learning has been used to fit artificial functions that maps inputs to outputs for estimating inverse kinematics and dynamics of robots in settings such as neuro-adaptive model-reference adaptive control [20], [21], [22]. In the context of soft robots, frameworks such as feed forward neural nets [23], [24], [25], and more recently recurrent neural nets [26], [27] have been used to characterize dynamic properties of different soft manipulators. While using neural networks makes the process of modeling simple, it is subject to issues like over-fitting and requires a good evaluation mechanism to guarantee stability of the control policy. Steps like two fold training and collection of data that define the work manifold effectively have to be carried out. This makes the process of training each robot different as the input-output mapping may vary for different prototypes. The same issue arises if the robots are subjected to external loads, which have been shown to impact the workspace of the BR^2 manipulator.

This paper presents promising results on the use of Reinforcement Learning (RL) for position control of the BR^2 soft manipulator. RL has been effectively used in the control of rigid-link robots as demonstrated in [28], [29]. The main benefit of RL as opposed to aforementioned neuro-adaptive control methodologies is that RL directly learns an optimal policy from experience. This precludes the need for a separate control strategy to choose optimal actions while transitioning between states. Control policies obtained using RL techniques are also more robust to external disturbances making them ideal for the BR^2 manipulator, whose workspace is dependent on external loads. RL implementations in the context of soft robots is relatively new and has focused on traditional Q-learning [30], [31]. Both implementations work on controlling planar 2D motions, and have a relatively small state-action space and use fixed steps during transitions. In the presented work, we extend this framework to a 3D setting rendered by the BR^2 continuum manipulator with highly nonlinear response characteristics between its actuation space (pneumatic pressure) and the task or the end effector space [10]. Additionally, we also incorporate the notion of adaptive steps to reduce the number of transitions to reach a target.

II. System Architecture

A. BR^2 Soft Robotic Manipulator Characterization

The BR^2 manipulator presented in [6], is a continuum robotic arm made from combining soft actuators namely Fiber Reinforced Elastomeric Enclosures (FREEs). FREEs are made of a hollow cylindrical tube reinforced with inextensible fibers wrapped in a helical shape on its outer surface. The tube is made of hyper-elastic materials such as latex rubber or silicone, thereby generating large strain. The angles at which the fibers are wrapped on the FREE determine its deformation behaviour upon being pressurized. Depending on number of fibers and their relative angles, FREEs can generate contraction, extension, rotation, bending, or a combination of these motions. For information regarding the fabrication of FREEs, refer to [32]. As the name suggests, the BR^2 has one bending and two rotating actuators that work in tandem to achieve complex 3D spatial motions. Fig.2 shows a pressurized BR^2 and half its workspace. The work space is symmetric about the robot sagittal plane. The workspace takes the form of a convex hull that is narrower at the bottom and widens at the top with localized shearing occurring intermittently. The two modes of actuation are coupled and this manifests as a non-linear response of the BR^2 end effector position to the FREE pressure inputs at different actuation ranges. For example, the rotational sweep of the manipulator at lower bending pressures is much smaller than at higher bending pressures.

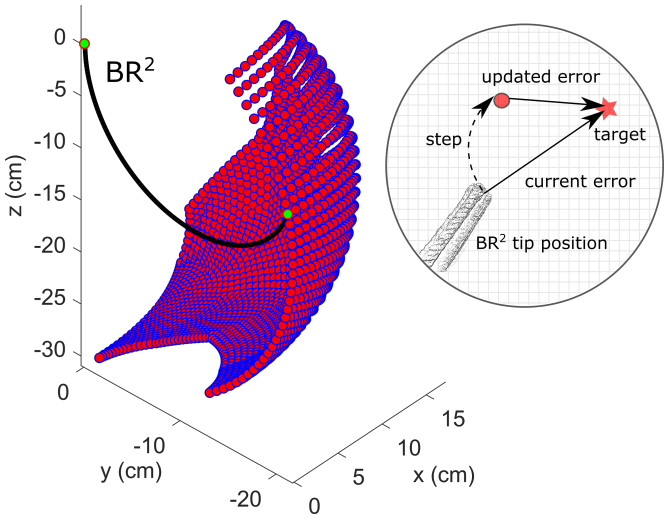

Fig. 2. Half-workspace of the BR^2 manipulator. Manifold corresponds to the tip positions at different pressures. Inset: MDP formulation of the control problem.

B. Markov Decision Process formulation

Towards developing a robust control strategy that can account for nonlinearities inherent in soft mechanical systems the following definitions are presented. The use of reinforcement learning on a robotic system requires it to be abstracted and represented as a Markov Decision Process (MDP). An MDP is characterized by states (s), actions (a) and rewards (r). Assuming the simplest form of representation, the BR^2 manipulator is abstracted as follows:

- State (s) : States are defined as the position of continuum arm with respect to the target. The workspace around manipulator's tip is discretized into a 3D grid with a resolution of 0.01 m. The discretized grid is centered on the target location to which the tip is directed to move. Therefore, the state of the BR^2 at any given pressure is a 3D vector describing the relative distance between the manipulator tip and target location and the current actuation pressures.

- Action (a) : Each of three FREEs in the BR^2 is capable of being pressurized and depressurized. However, as the workspace is symmetric about the manipulator sagittal plane, only one half of the BR^2 workspace is considered, which corresponds to the use of the bending and one rotating FREE. Any movement in the other half of the workspace can be achieved by implementing the same actions for the second rotating FREE. The system is capable of producing 12 actions corresponding to different magnitudes of pressurization and depressurization. The actions for the bending FREE are: ± 3.45, ± 6.89, ± 13.79 kPa and for the rotating FREE are ± 6.89, ± 13.79 and ± 27.57 kPa. This enables the system to choose an optimal sequence of action to reach the target in the fewest possible steps in a robust manner.
- Reward (r) : The reward quantifies the effect an action has on the manipulator's tip position with regard to the target. A reward system with an inductive bias is used to speed up the learning process. In our implementation, we use the L2 norm between the current tip position and target position to determine the reward and any action bringing the manipulator closer to the target returns a higher reward than an action moving it away from the target. The reward structure is described as follows:

$$r = \begin{cases} -2 + err_{prev} - err_{curr} + \zeta \\ -100, \ P > P_{max} \\ +100, \ err_{curr} \leq \varepsilon \end{cases} \quad (1)$$

The reward is structured to penalize every transition made by the system, forcing it to learn the most optimal path to the target. ζ is a tunable parameter used to penalize transitions in regions where the system could become unstable. In our implementation, this corresponds to the boundaries of the workspace. By biasing the system to avoid the trajectories along the boundaries, the policy generated becomes more robust to the warping of the workspace, a common occurrence for the BR^2 arm under external loading. Furthermore, the reward structure also penalizes any action that forces the pressure in the actuators to exceed the prescribed limits. This prevents the system from getting stuck at the outer boundaries of the workspace where any additional pressurization does not change the position of the manipulator. The system is rewarded a large positive value when a target has been reach within a threshold (ε).

C. Deep-Q Learning (DQN) Framework

Q-Learning is a model-free reinforcement learning (RL) technique that can identify an optimal action-selection policy for a given finite MDP. It is grounded on learning an action-value function which gives the expected utility for a given action when the system is at a particular state. A policy, Π, is a rule that the agent follows in selecting actions, given the state it is in. The value iteration update of the Q function follows the Bellman equation and is gives as:

$$Q_t(s_t, a_t) = Q_t(s_t, a_t) + \alpha(r_t + \gamma \ max_a \ Q(s_{t+1}, a) - Q_t(s_t, a_t)) \quad (2)$$

where s_t : is the state of the system at time t, a_t : the action it has taken to reach the new state s_{t+1}, r_t is the reward for taking action a_t, α is the learning rate, and γ is the discount factor. To account for large dimensionality, neural nets (NNs) are used to approximate the Q-functions and such frameworks are called Deep Q-Learning (DQN) networks [33]. In effect, the NN accepts states as inputs and outputs the quality associated with available actions that the system can perform and consequently, an action resulting in a larger value is considered superior. The loss function for a given state action pair is defined as the difference between the output obtained from the NN and the value obtained from the target Q-function ($\tilde{Q}$). The optimal policy is given by Eq.3.

$$\Pi(s) = max_a \tilde{Q}(s, a) \quad (3)$$

When applying RL in domains with very large state spaces, the experience obtained by the learning agent in interacting with the environment must be generalized. This is crucial for applications like position control that relies on a generalized control law. Towards this, experience replay is used to train the DQN. This ensures that the policy generated does not drive the system into a local minimum.

Algorithm 1 DQN Framework

```
for episode do
    Select source and target positions
    reached ← False , done ← False
    iter = 0
    Initialize memory
    while not done do
        action ← Q(state)
        next state, reward ← simulation output
        remember ← {s, a, r, s', done}
        iter += 1
        if iter > size(memory) then
            update exploration rate
            replay(batch size)
        end if
        if (err_curr < threshold) then
            done ← True
            reached ← True
        end if
        if (iter > iter_max or P > P_max) then
            done ← True
            reached ← False
        end if
    end while
end for
```

Algorithm 2 Experience replay

for *sample in memory*(*batch size*) **do**
  $\{s, a, r, s', done\} \leftarrow memory$
  if (*reached*) **then**
    $r \leftarrow 100$
  else
    if (*not done and not reached*) **then**
      $r \leftarrow -2 + \Delta Err + \zeta$
    else
      $r \leftarrow -100$
    end if
  end if
  $target \leftarrow r + \gamma * \max_{a'} Q(s')$
  $Q[state, action] \leftarrow target$
end for
Train DQN

III. Testing and Results

As mentioned previously, the strengths of an RL-based control system are two fold: (1) The system learns to choose an optimal sequence of actions and hence an optimal path to reach the target and (2) The system is robust to external disturbances. In this section we validate the system for its general efficacy in position control and robustness to disturbances through experiments in simulation and on a prototype of the BR^2 continuum arm.

A. Training

The prototype BR^2 has a length of 0.31 m with maximum operating pressures of 172.36 kPa in bending and 193.05 kPa in rotation. We use a DQN with 3 hidden layers having 512 nodes and *tanh* activation functions. An $\varepsilon-$greedy action selection strategy (ε = 0.1) is used while training. To expedite training we use the model developed in [6] to train the system on a simulation. The simulation is based on a Cosserat rod formulation [17], with the elasticity and precurvature parameters fit with experimental results [5]. It is important to note that the simulation is used as a data generator for one time data collection and the same could be done on a prototype in an automated manner. An Adam stochastic gradient optimizer (learning rate α=0.002) with mean-squared-error as the objective function is used to train the NN. In each episode, a tuple of points is randomly selected from the training dataset and the system is trained to transition between the two points using the ε-greedy policy. The episode is terminated in one of three ways: (1) the position error is below the threshold value, (2) number of steps exceed allowable limit and (3) the manipulator pressure exceeds the maximum rated pressure. The system is trained for 5000 episodes before evaluation. During evaluation, a greedy policy is used to select the best actions from the learned Q function to transition between states.

B. Results from simulation

Fig. 3 shows the path taken by the system for two sample trajectories with five waypoints each when the learned policy

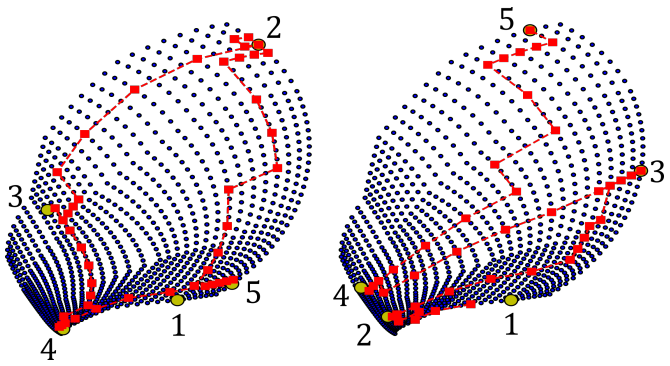

Fig. 3. Path taken (shown in red) by system for two sample trajectories (1-2-3-4-5)

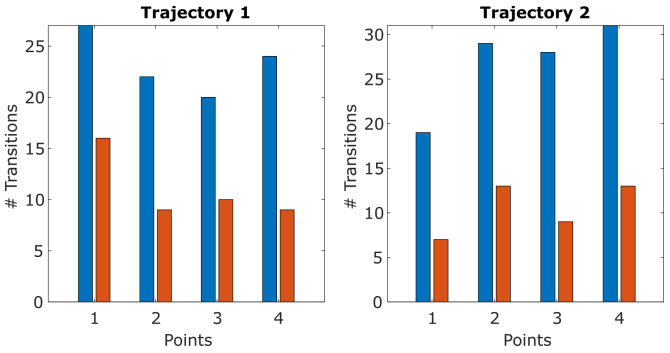

Fig. 4. Effect of adaptive steps during transitions. Blue: Fixed steps and Red: Adaptive steps

is evaluated. The first trajectory emphasizes on reaching extremal points that are more unlikely to be sampled during training while the second trajectory forces the system to sweep through the workspace laterally. While the position error for all the waypoints are within prescribed limits (0.01 m), two important observations can be made:

- When transitioning between waypoints, the system takes larger strides initially and switches to smaller stride as it gets closer to the target. If the separation between the waypoints is small, the system only uses smaller strides. As seen in Fig. 4, this ensures that fewer transitions are made while making sure that the target is not overshot.
- When a target is located at the edges of the reachable workspace the system learns to avoid taking actions that would exceed the prescribed pressure limits thereby preventing it from getting stuck at a local minimum. In effect, the system learns the boundaries of the reachable workspace.

Unlike conventional rigid manipulators, soft manipulators experience attenuation in their reachable workspace when loaded. Fig. 5 illustrates this effect where the work space is observed to translate and distort (or shear) when a tip load is applied. To validate the robustness of the learned policy against such external loads we ran experiments with the BR^2 loaded at the tip. This draws parallels from traditional pick and place tasks in robotic manipulation. We ran experiments to evaluate the ability to generalize of the learned RL policy

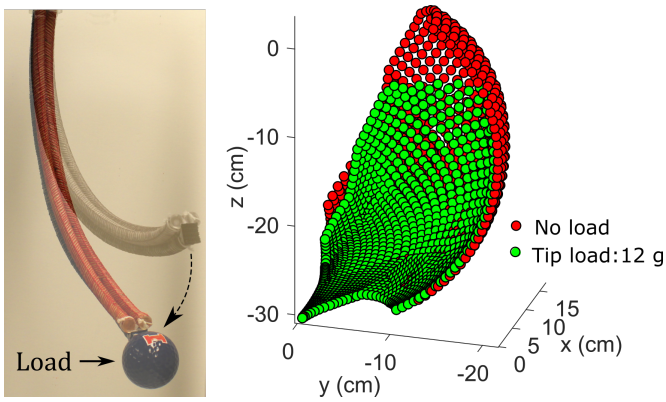

Fig. 5. Attenuation of the BR^2 workspace with load

TABLE I
EFFECT OF EXTERNAL LOAD ON CONTROL POLICY ACCURACY

Loads (g)	# Trajectories		
	Error $\leq$ 1.5 cm	1.5 cm $\leq$ Error <3.0 cm	3.0 cm $\leq$ Error
0	9730	249	21
6	9719	260	21
9	9588	367	45
12	9592	343	65
18	9460	376	164

to different end loads. the system is trained with no external load and the learned RL policy is evaluated on different external loads. To obtain statistically significant results, 10,000 random trajectories from the reachable work spaces corresponding to the different tip loads are sampled and classified as being successful or unsuccessful. The threshold for success corresponds to a position error < 0.015 m ($\approx$ radius of the end effector). Table I specifies the effect of four loads (6 g, 9 g, 12 g, and 18 g) on the accuracy of the system by detailing the number of unsuccessful attempts and the margin of error for each attempt. The accuracy is observed to decrease (from 97.3% to 94.6%) with increase in tip load and this is directly correlated to the extent by which the workspace warps. For a load of 18 g the number of trajectories with an error > 0.03 m ($\approx$ diameter of the end effector) increases rapidly and hence this region can be considered the limit above which the system may fail to perform effectively.

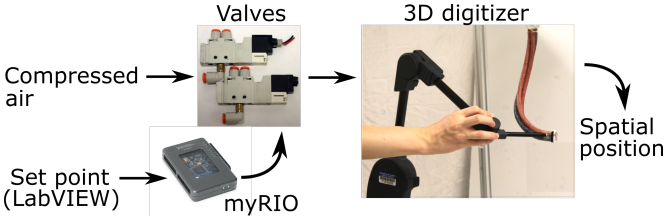

Fig. 6. Validation of the system on the BR^2 prototype

C. Results from physical prototype

To validate the results with a physical prototype of the BR^2 we use the setup shown in Fig. 6. The BR^2 is connected to a pneumatic pressure source which is controlled using a pressure regulator (SMC ITV0050-2UN). A LabVIEW interface integrated with a myRIO-1900 (National Instruments) controls the input voltage to the pressure regulator and hence the configuration of the BR^2. The tip position is captured using a 3D digitizer from MicroScribe. A single trajectory having four waypoints is analyzed for three loading cases. The trajectory as reported in the simulation is replicated on the prototype with no feedback and therefore we expect the observed errors to be of the same order of magnitude as those reported in [6]. Two important observations highlighted in the purview of this test are:

- The variation in input pressure required by the BR^2 to reach the same waypoints for different loads.
- The effectiveness of the system in attempting to reach points in the unreachable workspace.

Table II compares the errors obtained from the simulation and experiments. The errors from the simulation indicate a robust system reaching the waypoints with a mean error of 0.48 cm ($\approx$ 15% of the arm diameter) having a standard deviation of 0.41 cm in the reachable workspace. Results from the prototype are subject to factors like pre-curvature at zero input pressures (Fig.1(a)) and hysteresis during cycles of pressurization and depressurization. We report a mean error of 3.05 cm ($\approx$ the arm diameter) having a standard deviation of 0.97 cm for points in the reachable workspace. This aligns with the errors reported in [6] between simulation and experimental results. To compare the error associated with points in the unreachable workspace, the corresponding nearest point in the reachable workspace for each point is used as a reference to calculate the error. The results are tabulated in Table III. For simplicity the unreachable points are labeled i$-$iv from Table II. The points are tracked robustly in simulation with the maximum variation in error of 0.31 cm ($\approx$ 10% of the arm diameter). In case of the prototype, the maximum variation is 2.98 cm ($\approx$ the arm diameter).

It is important to note that the implementation of such an open loop system with learned control policy offers a foundation upon which a more robust closed loop systems with feedback can be developed. This in turn offers a new regime for complex soft manipulators where the system is initially trained on a simulation to learn a robust control policy, which can then be transferred to a physical prototype integrated with a feedback mechanism such as those discussed in [34].

IV. CONCLUSION

This works showcases the use of deep reinforcement learning as a viable approach for generalized position control for a novel pneumatic continuum arm known as the BR^2 that deforms by combining spatial bending and torsion. Through a valid training and testing scheme, the efficacy of the approach can be attested to with an accuracy >94% even for

TABLE II

SYSTEM VALIDATION AGAINST BR^2 PROTOTYPE. UNREACHABLE POINTS ARE INDICATED USING $*$

waypoints		1		2		3		4	
	Load (g)	Pressure (kPa)	Error (cm)	Pressure (kPa)	Error (cm)	Pressure (kPa)	Error (cm)	Pressure (kPa)	Error (cm)
Simulation	0	(48.26,55.15)	0.0	(151.68,0)	0.0	(158.58,103.42)	0.0	(124.1,124.1)	0.0
	6	(55.15,34.47)	0.37	(158.58,0)	0.98	(158.58,75.84)	1.05	(131.0,103.42)	0.63
	9	(55.15,34.47)	0.92	(158.58,0)	*i	(162.03,62.05)	*ii	(134.45,96.52)	0.74
	12	(62.05,34.47)	0.25	(165.47,0)	*iii	(165.47,62.05)	*iv	(134.45,89.63)	0.87
Prototype	0	(48.26,55.15)	3.35	(151.68,0)	1.20	(158.58,103.42)	1.62	(124.1,124.1)	2.98
	6	(55.15,34.47)	3.46	(158.58,0)	2.84	(158.58,75.84)	4.87	(131.0,103.42)	3.87
	9	(55.15,34.47)	2.72	(158.58,0)	*i	(162.03,62.05)	*ii	(134.45,96.52)	3.68
	12	(62.05,34.47)	3.01	(165.47,0)	*iii	(165.47,62.05)	*iv	(134.45,89.63)	3.05

TABLE III

ERROR COMPARISON FOR UNREACHABLE POINTS IN EXPERIMENTS WITH BR^2 PROTOTYPE

Points	i	ii	iii	iv
Distance to closest point in reachable workspace (cm)	1.65	1.31	2.18	1.65
Distance to BR^2 tip position from simulation (cm)	1.65	1.60	2.48	1.79
Distance to BR^2 tip position from prototype (cm)	2.10	4.29	0.90	3.05

large external loads. The system has two important features, which are (a) its ability to choose appropriate actions to reduce the number of transitions and (b) its considerable robustness to external loads including self weight. We also validate the system for its ability to reach the best possible configuration in cases where the target lies in the unreachable work space. The results clearly indicate that reinforcement learning can emerge as a viable technique for learning control policies for complex continuum manipulators which are otherwise hard to model and control using methods based on first principles alone. Future work involves incorporating the system with sensors to create a framework with real time feedback of the BR^2's tip position. Closed loop control, coupled with online policy adaptation based on the feedback, could significantly improve the tracking performance. In addition, we also plan to extend the system for more complex manipulators with greater degrees of freedom. Examples in this regard include mounting the BR^2 on a rotating base or on a moving robot, or connecting multiple BR^2 segments in series for greater dexterity. Lastly, this work will establish a baseline for exploring other RL paradigms that offer a more continuous mode for control, such as Deep Deterministic Policy Gradients (DDPG) and Proximal Policy optimization (PPO). It additionally can also provide a good case study for emerging work in simulation-to-real learning and transfer learning paradigms, such as Target Apprentice based Transfer Learning or Meta Learning.

ACKNOWLEDGEMENT

This work was supported by NSF CMMI-1454276, and in part by the NIFA-NSF joint National Robotics Initiative program.

REFERENCES

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

[2] R. F. Shepherd, A. A. Stokes, R. M. D. Nunes, and G. M. Whitesides, "Soft machines that are resistant to puncture and that self seal," *Advanced Materials*, vol. 25, no. 46, pp. 6709–6713, 2013.

[3] G. Singh, C. Xiao, E. T. Hsiao-Wecksler, and G. Krishnan, "Design and analysis of coiled fiber reinforced soft pneumatic actuator," *Bioinspiration & Biomimetics*, feb 2018. [Online]. Available: http://iopscience.iop.org/article/10.1088/1748-3190/aab19c

[4] F. Ilievski, A. D. Mazzeo, R. F. Shepherd, X. Chen, and G. M. Whitesides, "Soft Robotics for Chemists," *Angewandte Chemie International Edition*, vol. 50, no. 8, pp. 1890–1895, feb 2011. [Online]. Available: http://doi.wiley.com/10.1002/anie.201006464

[5] N. K. Uppalapati, G. Singh, and G. Krishnan, "Parameter estimation and modeling of a pneumatic continuum manipulator with asymmetric building blocks," in *2018 IEEE International Conference on Soft Robotics (RoboSoft)*. IEEE, apr 2018, pp. 528–533. [Online]. Available: https://ieeexplore.ieee.org/document/8405380/

[6] N. K. Uppalapati and G. Krishnan, "Design of soft continuum manipulators using parallel asymmetric combination of fiber reinforced elastomers," Aug 2018. [Online]. Available: engrxiv.org/z892e

[7] S. Neppalli, B. Jones, W. McMahan, V. Chitrakaran, I. Walker, M. Pritts, M. Csencsits, C. Rahn, and M. Grissom, "OctArm-A soft robotic manipulator," in *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*. IEEE, 2007, p. 2569.

[8] W. McMahan, B. A. Jones, I. D. Walker *et al.*, "Design and implementation of a multi-section continuum robot: Air-octor." in *IROS*, 2005, pp. 2578–2585.

[9] D. Trivedi, D. Lesutis, and C. D. Rahn, "Dexterity and workspace analysis of two soft robotic manipulators," in *ASME 2010 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers, 2010, pp. 1389–1398.

[10] T. George Thuruthel, Y. Ansari, E. Falotico, and C. Laschi, "Control Strategies for Soft Robotic Manipulators: A Survey," *Soft Robotics*, vol. 5, no. 2, pp. 149–163, apr 2018. [Online]. Available: http://www.liebertpub.com/doi/10.1089/soro.2017.0007

[11] C. Escande, T. Chettibi, R. Merzouki, V. Coelen, and P. M. Pathak, "Kinematic calibration of a multisection bionic manipulator," *IEEE/ASME transactions on mechatronics*, vol. 20, no. 2, pp. 663–674, 2015.

[12] I. S. Godage, G. A. Medrano-Cerda, D. T. Branson, E. Guglielmino, and D. G. Caldwell, "Modal kinematics for multisection continuum arms," *Bioinspiration & biomimetics*, vol. 10, no. 3, p. 035002, 2015.

[13] F. Renda, M. Cianchetti, M. Giorelli, A. Arienti, and C. Laschi, "A 3d steady-state model of a tendon-driven continuum soft manipulator inspired by the octopus arm," *Bioinspiration & biomimetics*, vol. 7, no. 2, p. 025006, 2012.

[14] F. Renda, M. Giorelli, M. Calisti, M. Cianchetti, and C. Laschi, "Dynamic model of a multibending soft robot arm driven by cables," *IEEE Transactions on Robotics*, vol. 30, no. 5, pp. 1109–1122, 2014.

[15] A. D. Marchese, R. Tedrake, and D. Rus, "Dynamics and trajectory optimization for a soft spatial fluidic elastomer manipulator," *The International Journal of Robotics Research*, vol. 35, no. 8, pp. 1000–1019, 2016. [Online]. Available: http://ijr.sagepub.com/content/35/8/1000.abstract

[16] F. Renda, F. Boyer, J. Dias, and L. Seneviratne, "Discrete cosserat approach for multi-section soft robots dynamics," *arXiv preprint arXiv:1702.03660*, 2017.

[17] D. Trivedi, A. Lotfi, and C. D. Rahn, "Geometrically exact models for soft robotic manipulators," *IEEE Transactions on Robotics*, vol. 24, no. 4, pp. 773–780, 2008.

[18] N. Simaan, K. Xu, W. Wei, A. Kapoor, P. Kazanzides, R. Taylor, and P. Flint, "Design and integration of a telerobotic system for minimally invasive surgery of the throat," *The International journal of robotics research*, vol. 28, no. 9, pp. 1134–1153, 2009.

[19] S. M. H. Sadati, S. E. Naghibi, A. Shiva, I. D. Walker, K. Althoefer, and T. Nanayakkara, "Mechanics of Continuum Manipulators, a Comparative Study of Five Methods with Experiments," in *Conference Towards Autonomous Robotic Systems*, 2017, pp. 686–702. [Online]. Available: http://link.springer.com/10.1007/978-3-319-64107-2_56

[20] H. A. Kingravi, G. Chowdhary, P. A. Vela, and E. N. Johnson, "Reproducing kernel hilbert space approach for the online update of radial bases in neuro-adaptive control," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 7, pp. 1130–1141, 2012.

[21] L. Behera, S. Chaudhury, and M. Gopal, "Neuro-adaptive hybrid controller for robot-manipulator tracking control," *IEE Proceedings-Control Theory and Applications*, vol. 143, no. 3, pp. 270–275, 1996.

[22] S. Khemaissia and A. S. Morris, "Neuro-adaptive control of robotic manipulators," *Robotica*, vol. 11, no. 5, pp. 465–473, 1993.

[23] M. Giorelli, F. Renda, M. Calisti, A. Arienti, G. Ferri, and C. Laschi, "Neural network and jacobian method for solving the inverse statics of a cable-driven soft arm with nonconstant curvature," *IEEE Transactions on Robotics*, vol. 31, no. 4, pp. 823–834, 2015.

[24] M. Rolf, J. J. Steil, and M. Gienger, "Online goal babbling for rapid bootstrapping of inverse models in high dimensions," in *Development and Learning (ICDL), 2011 IEEE International Conference on*, vol. 2. IEEE, 2011, pp. 1–8.

[25] A. Melingui, O. Lakhal, B. Daachi, J. B. Mbede, and R. Merzouki, "Adaptive neural network control of a compact bionic handling arm," *IEEE/ASME Transactions on Mechatronics*, vol. 20, no. 6, pp. 2862–2875, 2015.

[26] T. G. Thuruthel, E. Falotico, M. Manti, and C. Laschi, "Stable open loop control of soft robotic manipulators," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 1292–1298, 2018.

[27] T. G. Thuruthel, E. Falotico, F. Renda, and C. Laschi, "Learning dynamic models for open loop predictive control of soft robotic manipulators," *Bioinspiration & biomimetics*, vol. 12, no. 6, p. 066003, 2017.

[28] N. Birdwell, S. Livingston, and I. Elhanany, "Reinforcement learning in sensor-guided aibo robots," *Univ. Tennesse, Knoxville, TN, USA, Tech. Rep*, 2007.

[29] S. Gu, E. Holly, T. Lillicrap, and S. Levine, "Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 3389–3396.

[30] X. You, Y. Zhang, X. Chen, X. Liu, Z. Wang, H. Jiang, and X. Chen, "Model-free control for soft manipulators based on reinforcement learning," in *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*. IEEE, 2017, pp. 2909–2915.

[31] H. Zhang, R. Cao, S. Zilberstein, F. Wu, and X. Chen, "Toward effective soft robot control via reinforcement learning," in *International Conference on Intelligent Robotics and Applications*. Springer, 2017, pp. 173–184.

[32] G. Singh and G. Krishnan, "A constrained maximization formulation to analyze deformation of fiber reinforced elastomeric actuators," *Smart Materials and Structures*, vol. 26, no. 6, p. 065024, 2017.

[33] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, p. 529, 2015.

[34] G. Joshi and G. Chowdhary, "Cross-domain transfer in reinforcement learning using target apprentice," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 7525–7532.