2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Learning to Control the Whole-body Shape of a Soft Robotic Arm in Unknown Situations

Zhiqiang Tang, Qianqian Wang, Daniela Rus, Cecilia Laschi

***Abstract*— Control of soft robots is considered one of the key elements in achieving their intelligence. However, it faces challenging problems such as nonlinear dynamics, highly deformable structures, and operation in unpredictable situations. Numerous methods have been proposed to overcome these challenges, but most of them focus on controlling only a small part of the soft robot's body, such as the end effector. Whole-body shape control is a problem that has not yet been fully explored, but it is critical for tasks that require whole-body path planning to navigate in confined or crowded spaces. In this study, we developed a convolutional neural network (CNN)-based approach for controlling the robot's whole-body shape. The key novelty of our approach is that it learns a purely image-driven CNN control policy with online adaptive capability. Our approach has three main components: (1) training an offline shape policy to offer basic actions, (2) building a shape model and updating it online to maintain accuracy, (3) conducting Bayesian optimization based on the basic action and shape model to obtain optimal performance. The presented approach is validated on a soft robotic arm and experimental results demonstrate that the soft arm can be controlled to achieve target shapes and adapt to different previously unknown situations. Meanwhile, our approach achieved better shape control performance than the state-of-the-art method. Overall, this work presents a feasible learning-based approach to the whole-body shape control problem and contributes to the development of soft robot intelligence from the control perspective.**

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

I. Introduction

Soft robots are characterized by their soft materials and compliant structures, which allow them to deform continuously and conform to unstructured spaces [1]. These characteristics make soft robots ideal for applications requiring safe and precise interactions, such as rehabilitation [2], [3], in-vivo drug delivery [4], minimally invasive surgery [5]. Despite their advantages, controlling soft robots is a significant challenge due to their high degrees of freedom, nonlinear dynamics, and susceptibility to external disturbances [6]. In the last decade, a number of control strategies have been developed to overcome these challenges [7], [8]. However, most existing control methods focus on specific parts of a soft robot, such as the end-effector or individual segments, while neglecting the control of the whole-body shape [9], [10]. This partial approach limits the potential of soft robots to fully leverage their unique deformability and versatility. The control of whole-body shape, which is crucial for coordinated and adaptive interactions with the environment, has not yet been thoroughly investigated.

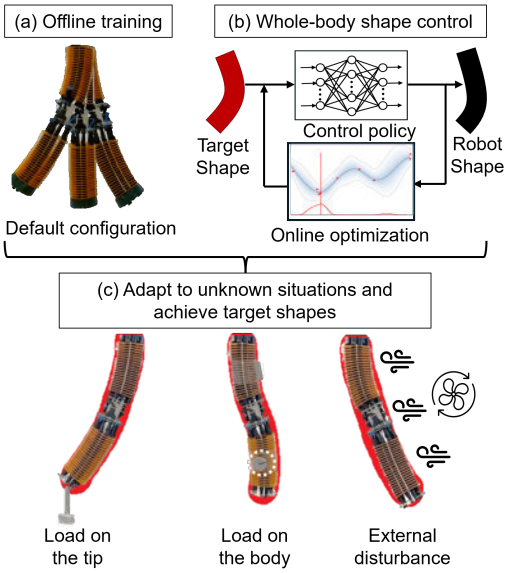


Fig. 1: The general idea and functionalities of our whole-body shape control approach. (a) Offline training is conducted on a soft robotic arm's default configuration. (b) The control policy drives the soft robotic arm to match the target shape and is optimized online to improve performance. (c) The soft robotic arm can be controlled to achieve different target shapes and adapt to previously unknown situations such as additional loads on the robot's tip and body, and random external disturbances.

Whole-body shape control holds significant promise for enhancing the capabilities of soft robots. This capability allows soft robots to navigate complex scenarios, manipulate objects with greater dexterity, and maintain robust performance across diverse applications. However, controlling the whole-body shape presents significant challenges because soft robots typically have infinitely many degrees of freedom due to their continuous and deformable structures.

This work was supported, in part, by the State Key Laboratory of Mechanical System and Vibration; in part, by the National Natural Science Foundation of China (52575652); in part, by the Start-up Research Fund of Southeast University (RF1028625147); in part, by National Key R&D Program of China (2025YFE0214200); in part, by National Research Foundation, Singapore, under Centre for Advanced Robotics Technology Innovation (CARTIN) program and Mens, Manus, and Machina project.

Zhiqiang Tang and Qianqian Wang are with School of Mechanical Engineering, Southeast University, China (email: zhiqiang.tang@seu.edu.cn, qqwang@seu.edu.cn)

Cecilia Laschi is with Department of Mechanical Engineering, National University of Singapore, Singapore (email: cecilia.laschi@nus.edu.sg)

Daniela Rus is with Computer Science and Artificial Intelligence Laboratory (CSAIL), Massachusetts Institute of Technology, USA (email: rus@csail.mit.edu)

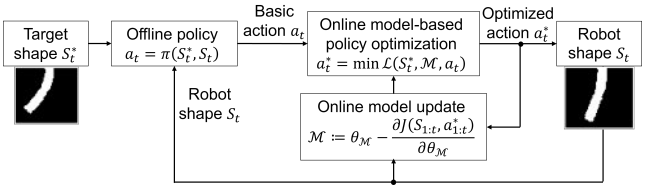


Fig. 2: Control diagram. A self-defined target shape is provided for the offline trained policy. Then the policy outputs a basic action. The basic action is further optimized to acquire better performance. Afterwards, the optimal action and robot shape are collected to refine the model. The process continues until the robot achieves the target shape.

This makes their configuration space extremely large and difficult to model or control using conventional techniques. Researchers in [11] proposed a visual servoing shape control method for a soft robot manipulator using Bézier curve features. Another study in [12] presented a learning-based shape controller using long-short-term-memory (LSTM) neural network. These studies focused on controlling the shape of the robot's backbone curve or contour not the entire body. On the other hand, a study in [13] used koopman operator to control the shape of soft robots, but it was tested in simulation only. Researchers in [14] developed a convolutional neural network (CNN) based inverse kinematic shape controller, but it was tested in static situations where no changes were made to the robot.

In this study, we developed an image-driven policy to control the whole-body shape of soft robots and the policy can be optimized online to adapt to preciously unknown situations, as shown in Fig.1. Our approach is built on three key components to ensure effective and adaptive whole-body shape control: (1) developing an offline shape policy that provides basic actions for online optimization to start with, (2) employing model-based Bayesian optimization to iteratively refine and obtain optimal actions for specific target shapes and situations, and (3) continuously updating the shape model in real-time to preserve accuracy and account for uncertain changes in the robot's configuration. The control diagram is shown in Fig.2. Major contributions of this study include: (1) developing an adaptive shape control policy that enables the soft robot to achieve target shapes in unknown situations; (2) presenting an image-based modeling technique that can output the high dimensional shape information; (3) investigating online learning and optimization methods to improve control and modeling accuracy.

II. Method

This section explains the principle of our approach which includes the control policy, shape model, and online optimization, as elaborated in the following subsections.

A. Control Policy

The control policy is designed to learn a mapping from the robot shape to the action, which is:

$$\boldsymbol{a}_t = \boldsymbol{\pi}(\boldsymbol{s}_{t+1}, \boldsymbol{s}_t) \tag{1}$$

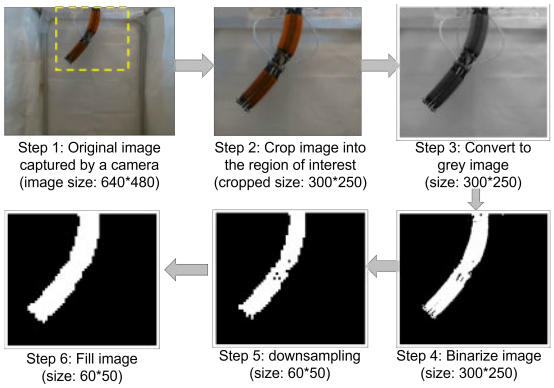


Fig. 3: Image processing. The original image is captured by a camera and cropped into the region of interest. Then the cropped image is converted to gray scale and binarized by Otsu thresholding.The binary image is downsampled and filled to get the final shape representation.

where $\boldsymbol{s}_t$ is the robot shape at time instance t, $\boldsymbol{s}_{t+1}$ is the robot shape at the next time instance, $\boldsymbol{a}_t$ is the robot action at time instance t, $\boldsymbol{\pi}$ is the mapping function. The robot shape is represented as an image and the image processing is shown in Fig.3. The original image is finally converted into a binary image after operations such as cropping, grayscale, downsampling and filling. Using the image as the input, the control policy is built on a convolutional neural network (CNN) architecture and outputs the action, as shown in Fig.4. Offline data can be collected by performing random actions on the soft robotic arm, which include shape images $\boldsymbol{S}_T = [\boldsymbol{s}_1, \boldsymbol{s}_2, \cdots, \boldsymbol{s}_T]$ and corresponding actions $\boldsymbol{A}_T = [\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_T]$. The policy is trained by minimizing the behavior cloning loss function, which is:

$$\mathcal{L}_{policy} = \sum_{t=1}^{T-1} \|\boldsymbol{\pi}(\boldsymbol{s}_{t+1}, \boldsymbol{s}_t) - \boldsymbol{a}_t\|_2 \tag{2}$$

where $\|\|_2$ is the Euclidean norm. As suggested by the study [14], training data were split into 10% and 90% for training and testing, respectively. The CNN architecture was designed by incrementally adding convolutional layers with Rectified Linear Unit (ReLU) activation functions until the loss function value on the testing data reached a plateau. In this work, 100 data samples were collected and the network was trained for 200 epochs using the Adam optimizer with a learning rate of 0.01. The training process was performed in Python using TensorFlow package.

B. Shape Model

The shape model aims to predict the shape information based on the current shape and action, which is:

$$\boldsymbol{s}_{t+1} = \mathcal{M}(\boldsymbol{s}_t, \boldsymbol{I}\boldsymbol{a}_t) \tag{3}$$

where $\boldsymbol{I}\boldsymbol{a}_t$ is the action image. $\boldsymbol{I}\boldsymbol{a}_t$ is obtained by repeatedly copying the action vector $\boldsymbol{a}_t$ into a 2-D image array which has the same size as the shape image. This allows easy concatenation of the shape image and action as the network

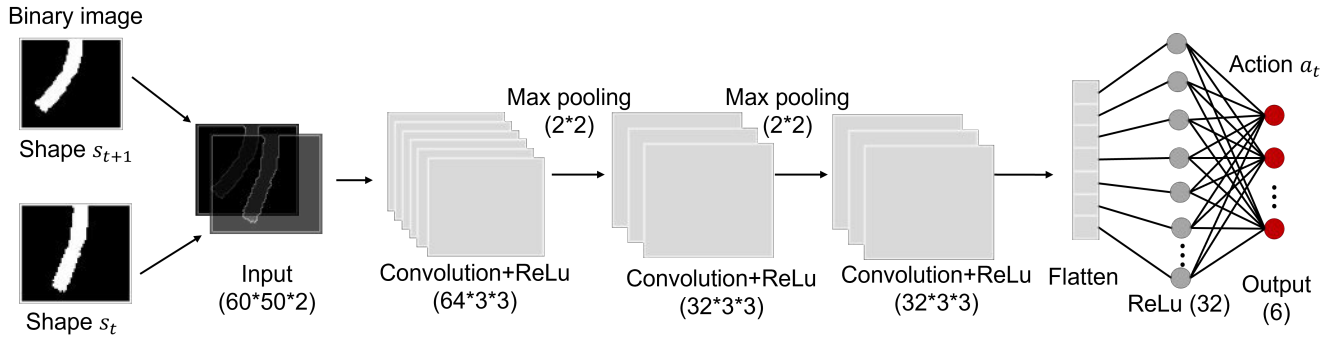


Fig. 4: Control policy architecture. The CNN-based controller takes the shape image as input and outputs the action. The policy contains 3 convolutional layers with ReLU activation functions and followed by a fully-connected network.

inputs. Similar to the design principle of the above control policy, the shape model is also constructed as a CNN architecture but with different inputs and outputs, as shown in Fig.6. The model is trained on the same offline data as the control policy, minimizing the shape prediction loss function:

$$\mathcal{L}_{model} = \sum_{t=1}^{T-1} \|\mathcal{M}(\boldsymbol{s}_t, \boldsymbol{I}\boldsymbol{a}_t) - \boldsymbol{s}_{t+1}\|_F \tag{4}$$

where $\|\|_F$ is the Frobenius norm. Moreover, the shape model will be updated online to keep accuracy. As suggested in the study [15], updating the last layer of the neural network can achieve good performance with minimal time. Thus, we update the last layer parameters by gradient descent using online data:

$$\boldsymbol{\theta}_{\mathcal{M}} \leftarrow \boldsymbol{\theta}_{\mathcal{M}} - \alpha \frac{\partial \mathcal{L}_{model}(\boldsymbol{s}_{t:t+H}, \boldsymbol{a}_{t:t+H})}{\partial \boldsymbol{\theta}_{\mathcal{M}}} \tag{5}$$

where $\boldsymbol{\theta}_{\mathcal{M}}$ is the last layer parameters, α is the online update rate (0.001 in this study), and H is the update data length counting from the latest data (10 in this study). The shape model is used for the following model-based online optimization.

C. Online Optimization

The objective of online optimization is to acquire optimal actions for the robot to cope with previously unknown situations. Bayesian optimization is one of the most efficient approaches to find optimum values of an objective function with a minimal number of function evaluations required [16]. The objective function is defined as:

$$g = \|\boldsymbol{s}_t^* - \mathcal{M}(\boldsymbol{s}_t, \boldsymbol{I}\boldsymbol{a}_t)\|_F \tag{6}$$

where $\boldsymbol{s}_t^*$ is the target shape and $\mathcal{M}$ is the above-mentioned shape model. The offline control policy offers a basic action $\boldsymbol{a}_t$ ($\boldsymbol{a}_t = \boldsymbol{\pi}(\boldsymbol{s}_t^*, \boldsymbol{s}_t)$) that can be a good starting point for online optimization. Based on the shape model, we generate virtual actions $\boldsymbol{V} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_T]$ and corresponding objective function values $\boldsymbol{G} = [g_1, g_2, \cdots, g_T]$. All the virtual actions are within the range $[0.5\boldsymbol{a}_t, 1.5\boldsymbol{a}_t]$. Then a surrogate function f, represented as a Gaussian process regression,

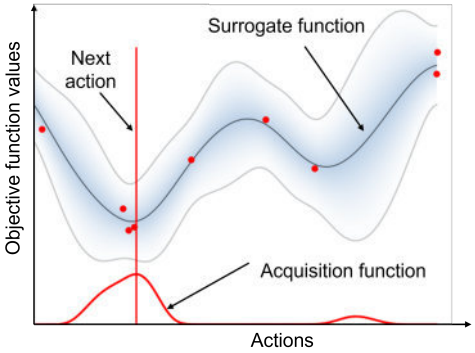


Fig. 5: The general idea of Bayesian optimization. Objective function values are generated by applying virtual actions to the shape model. Then the surrogate function captures the relationship between the objective function values and actions. Finally, the acquisition function proposes the next action close to the optimal value.

describes the relationship between the virtual actions and objective function values, which is:

$$\begin{gathered} f(\boldsymbol{v}) \sim \mathcal{GP}(\mu(\boldsymbol{v}), \sigma^2(\boldsymbol{v})) \\ \mu(\boldsymbol{v}) = \boldsymbol{k}^T \boldsymbol{K}^{-1} \boldsymbol{G} \quad \sigma^2(\boldsymbol{v}) = k(\boldsymbol{v}, \boldsymbol{v}) - \boldsymbol{k}^T \boldsymbol{K}^{-1} \boldsymbol{k} \end{gathered} \tag{7}$$

where k is a squared exponential kernel, $\boldsymbol{k}$ is kernel vector ($\boldsymbol{k}_i = k(\boldsymbol{v}, \boldsymbol{v}_i)$) and $\boldsymbol{K}$ is the kernel matrix ($\boldsymbol{K}_{ij} = k(\boldsymbol{v}_i, \boldsymbol{v}_j)$). The Gaussian process function f can be learned by evidence maximization [17].

Based on the surrogate function, an acquisition function is designed to guide the search for the optimum. We can select a control action that has a high probability near optimum by maximizing the acquisition function. Here, expected improvement (EI) is utilized as our acquisition function because EI balances the amount of improvement and the probability of improvement [18]. The expression of EI is given by:

$$\begin{aligned} \mathrm{EI}(\boldsymbol{v}) &= \int_{g^*}^{\infty} (f(\boldsymbol{v}) - g^*) \mathcal{N}\left(\mu(\boldsymbol{v}), \sigma^2(\boldsymbol{v})\right) df(\boldsymbol{v}) \\ &= (\mu(\boldsymbol{v}) - g^*) \Phi\left(\frac{\mu(\boldsymbol{v}) - g^*}{\sigma(\boldsymbol{v})}\right) + \sigma(\boldsymbol{v}) \phi\left(\frac{\mu(\boldsymbol{v}) - g^*}{\sigma(\boldsymbol{v})}\right) \end{aligned} \tag{8}$$

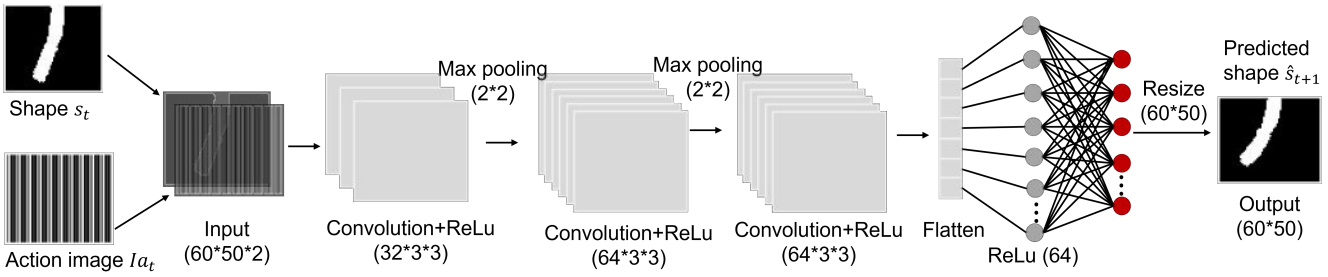


Fig. 6: Shape model architecture. The model takes the shape image and action as input and outputs the predicted shape information. The action image $\boldsymbol{Ia}_t$ is obtained by repeatedly copying the action vector $\boldsymbol{a}_t$ into a 2-D image array which has the same size as the shape image $\boldsymbol{s}_t$. The policy contains 3 convolutional layers with ReLU activation functions and followed by a fully-connected network. The network output is resized to be the same size of shape image.

Algorithm 1 Whole-body shape control of soft robots

1: **Initialize:**Perform T random actions on the soft robotic arm and record shape images $\boldsymbol{S}_T$ and corresponding actions $\boldsymbol{A}_T$.
2: Offline training the control policy (sec. II A) and the shape model (sec. II B)
3: **repeat**
4: Give a target shape $\boldsymbol{s}_t^*$.
5: Generate a basic action by the equation (1).
6: Optimize the action online by equations (sec. II C).
7: Apply the optimized action and record data.
8: Update the shape model online by the equation (5).
9: **until** Desired performance achieved

where g^* is the currently observed best value, Φ is normal cumulative function, and ϕ is normal density function.

The complete control workflow is shown in Algorithm 1. We first collect offline data by performing random actions on the soft robotic arm. Then we train the control policy and shape model. In the online situation, we provide a target shape for the control policy to generate a basic action. Next the basic action is optimized by model-based Bayesian optimization method. The optimized action is executed on the robot and online data is recorded to update the shape model. This process continues until the target shape is achieved.

III. Experiments

This section presents a experimental validation of the proposed control approach, using a soft robotic arm developed at Sant'Anna School of Advanced Studies [19]. The experimental setup is shown in Fig.7. The soft robotic arm is composed of two segments and each segment was driven by three cables. Each cable was controlled by a motor (HS-785HB Hitech Sail Winch Motor) and there were a total of six motors for the soft arm. A layer-by-layer plastic disk structure is inserted along the actuator length to constrain undesirable lateral and torsional movements. The distance between two consecutive layers is 10mm. The soft arm has a diameter of 60mm, a length of 375mm, and a weight of 220g. A camera (RealSense D435) was used to detect the soft arm's shape and sent the shape image to a computer.

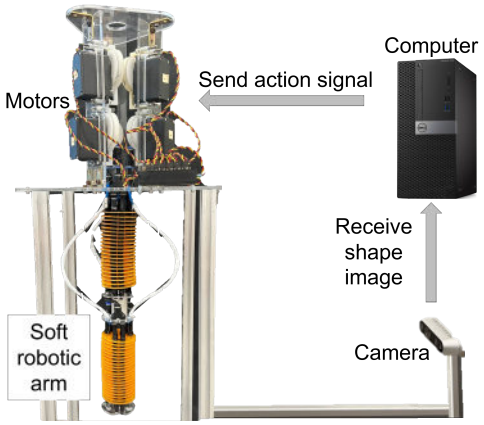
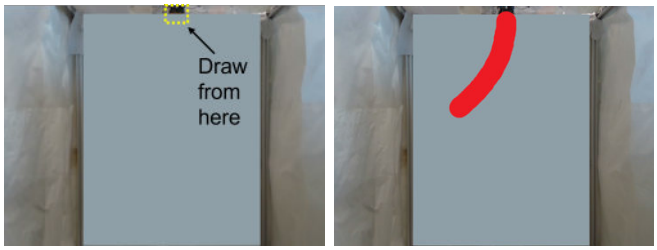


Fig. 7: Experimental setup. The soft robotic arm was driven by 6 cables and each cable was controlled by a motor. A camera detected the robot's shape and sent the shape image to the computer. The computer executed the control algorithm and sent action signals to the motors.

The computer processed the image information, executed the control algorithm, and sent the action signal to the motors.

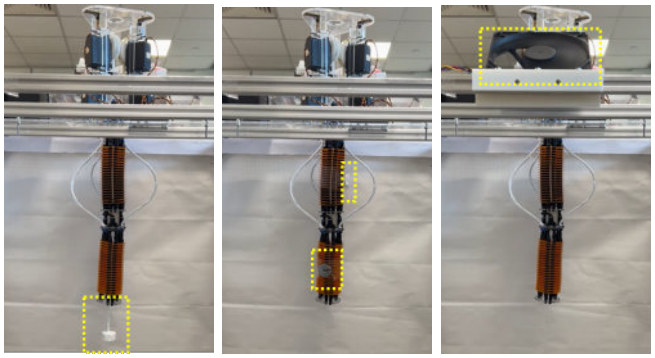
In the offline situation, we performed random actions on the soft robotic arm and collected 100 data samples including shape images and corresponding actions. The offline data were used to train the control policy and shape model. In the online situation, a target shape was given by hand-drawing a shape in a background canvas, as shown in Fig.8. We used a similar method to the study [14], where a background canvas was loaded into the Microsoft paint software and a target shape was created using the line tool. Its image size was then adjusted to be the same as the robot shape image.

We tested our approach in three different online situations which were previously unknown to the offline training, as shown in Fig.9. The three situations include: (1) we put a 100g load on the robot's tip to affect its tip motions. The additional tip load was almost half the robot's self-weight (220g) and had uncertain swaying during robot motions. (2) we attached a 50g load to both segments of the soft arm. The load position was randomly selected to cause uncertain body weight distribution. (3) A fan was put alongside the soft



(a) Background canvas (b) Draw a target shape

Fig. 8: Hand-drawn target shape. (a) A background canvas was loaded into the Microsoft paint software. (b) A target shape was created using the line tool.



(a) Load on the tip (b) Load on the body (c) Wind disturbance

Fig. 9: Online tested three situations which are unknown to the offline training. (a) Place a 100g load was put on the soft arm's tip. (b) Place a 50g load on each segment of the soft arm. (c) External wind causes motion disturbances.

arm. The fan speed (FS) can be adjusted to generate different wind disturbances. Meanwhile, we compared our approach with the deep visual inverse kinematic method presented in the study [14]. All the test conditions, including the offline data and hardware setup, were the same for our approach and the inverse kinematic method to have a fair comparison. The shape matching accuracy is defined as the follows:

$$accuracy = 1 - \frac{\sum_{i=1}^{I}\sum_{j=1}^{J}|\boldsymbol{S}_{ij}^{*} - \boldsymbol{S}_{ij}|}{I * J} \tag{9}$$

where $\boldsymbol{S}_{ij}^{*}$ and $\boldsymbol{S}_{ij}$ are the pixel value of the binarized target shape image and robot shape image, respectively. I and J are the image width and height, respectively.

The experimental results are shown in Fig.10-12, where various target shapes were created. These target shapes were not observed in the offline data, which indicated that our control policy can be generalized to shapes outside of the training set. The whole-body shape of the soft robotic arm can be controlled by our approach to achieve target shapes in previously unknown situations with high accuracy (larger than 90%). In contrast, the inverse kinematic method showed inferior performance and had limited ability to adapt to online unknown situations. Table I lists the shape matching accuracy. We can see that the best accuracy of our approach is 0.98 while the inverse kinematic is 0.91. The worst

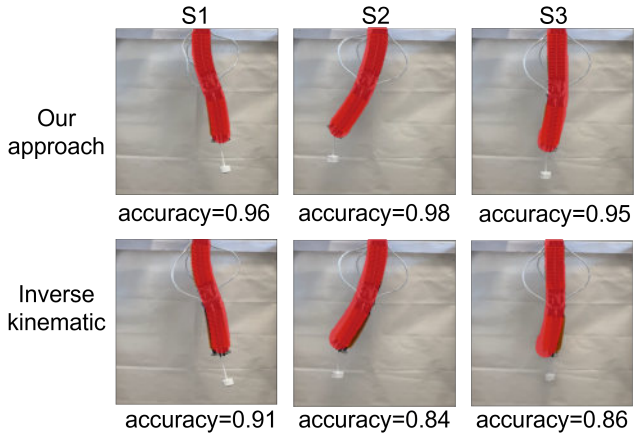


Fig. 10: Shape control performance when placing an extra 100g load on the tip of the soft robotic arm.

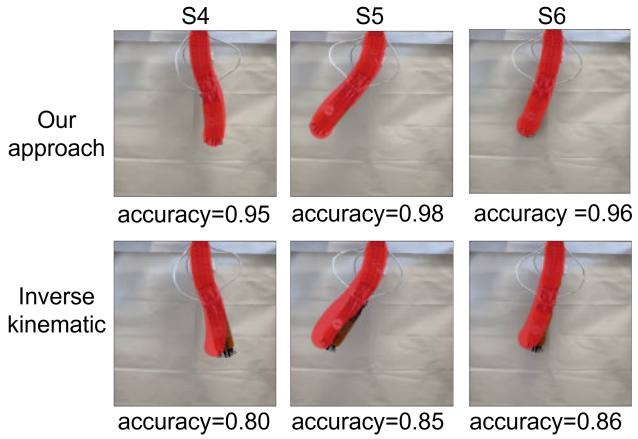


Fig. 11: Shape control performance when placing extra 50g loads on each segment of the soft robotic arm.

accuracy of our approach is 0.94 while the inverse kinematic is 0.67. The average performance over all the target shapes of our approach is 0.96±0.02 while the inverse kinematic is 0.82±0.07. This comparison shows our approach can achieve better performance than the inverse kinematic method. A supplementary video is also provided to show the real-time shape control performance.

The inverse kinematic method was only trained offline without online adaptation and thus limited its performance in unknown situations. Our approach adopted an online optimization strategy to acquire better actions than the offline policy. To make the online optimization efficient, we used the latest online data to update only the last layer parameter of the shape model. Meanwhile, we employed the offline policy to provide a good starting point and reduced searching range for the online optimization. Therefore, the soft robotic arm can adapt to unknown situations by our approach.

IV. Conclusion

In this work we developed a whole-body shape control approach for a soft robotic arm to achieve target shapes in

TABLE I: Shape control accuracy comparison

	S1	S2	S3	S4	S5	S6	S7	S8	S9	average performance
Our approach	0.96	0.98	0.95	0.95	**0.98**	0.96	0.97	0.98	**0.94**	0.96±0.02
Inverse kinematic	**0.91**	0.84	0.86	0.8	0.85	0.86	0.76	**0.67**	0.89	0.82±0.07

The average performance is the mean and standard deviation of all the 9 shapes (from S1 to S9).

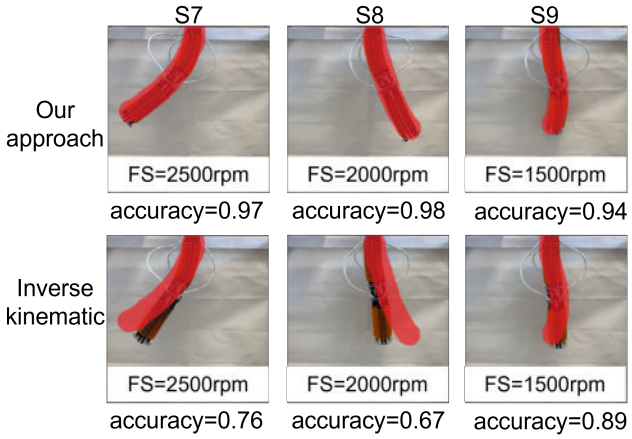


Fig. 12: Shape control performance when the soft robotic arm experienced unknown disturbances caused by different wind speeds. FS: fan speed. rpm: revolutions per minute.

previously unknown situations. This approach consisted of three key components that included a convolutional neural network (CNN)-based control policy, an image-based shape model, and a model-based online optimization technique. The CNN-based control policy was trained offline, providing basic actions and reducing the search range for the online optimization. The shape model can output the high-dimensional shape information and was updated online to maintain accuracy. Based on the trained policy and shape model, Bayesian optimization was employed to acquire optimal actions. Experiments on a soft robotic arm showed that our approach can control the soft arm to achieve different shapes with high accuracy and adapt to various unknown situations. Meanwhile, comparisons with another state-of-the-art shape control method showed that our approach had better performance and adaptive capability.

In our future work, we will extend the approach from 2D image to 3D morphology by incorporating point cloud data such as in the study [20]. Controlling the 3D whole shape needs to solve the computational complexity and deal with uncertainties of point cloud data. Moreover, we will generalize the approach to other soft robots and control larger shape deformations than the existing soft arm. Overall, this work presents a feasible whole-body shape control approach for soft robots, allowing online adaptation to unknown situations.

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] P. Polygerinos, Z. Wang, K. C. Galloway, R. J. Wood, and C. J. Walsh, "Soft robotic glove for combined assistance and at-home rehabilitation," *Robotics and Autonomous Systems*, vol. 73, pp. 135–143, 2015.

[3] Z. Q. Tang, H. L. Heung, X. Q. Shi, K. Y. Tong, and Z. Li, "Probabilistic model-based learning control of a soft pneumatic glove for hand rehabilitation," *IEEE Transactions on Biomedical Engineering*, vol. 69, no. 2, pp. 1016–1028, 2021.

[4] Q. Wang, N. Xiang, J. Lang, B. Wang, D. Jin, and L. Zhang, "Reconfigurable liquid-bodied miniature machines: magnetic control and microrobotic applications," *Advanced Intelligent Systems*, vol. 6, no. 2, p. 2300108, 2024.

[5] M. Runciman, A. Darzi, and G. P. Mylonas, "Soft robotics in minimally invasive surgery," *Soft robotics*, vol. 6, no. 4, pp. 423–443, 2019.

[6] 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, 2018.

[7] Z. Tang, L. Tian, W. Xin, Q. Wang, D. Rus, and C. Laschi, "A general soft robotic controller inspired by neuronal structural and plastic synapses that adapts to diverse arms, tasks, and perturbations," *Science Advances*, vol. 12, no. 2, p. eaea3712, 2026.

[8] C. Della Santina, C. Duriez, and D. Rus, "Model-based control of soft robots: A survey of the state of the art and open challenges," *IEEE Control Systems Magazine*, vol. 43, no. 3, pp. 30–65, 2023.

[9] D. Bruder, D. Bombara, and R. J. Wood, "A koopman-based residual modeling approach for the control of a soft robot arm," *The International Journal of Robotics Research*, p. 02783649241272114, 2024.

[10] Z. Q. Tang, H. L. Heung, K. Y. Tong, and Z. Li, "Model-based online learning and adaptive control for a "human-wearable soft robot" integrated system," *The International Journal of Robotics Research*, vol. 40, no. 1, pp. 256–276, 2021.

[11] F. Xu, Y. Zhang, J. Sun, and H. Wang, "Adaptive visual servoing shape control of a soft robot manipulator using bezier curve features," *IEEE/ASME Transactions on Mechatronics*, vol. 28, no. 2, pp. 945–955, 2022.

[12] Y. Shen, J. Zhang, Y. Yuan, F. Zhang, and H. Ding, "Online learning based shape control for a soft manipulator based on spatial features feedback," *IEEE Robotics and Automation Letters*, 2024.

[13] A. Singh, J. Sun, and J. Zhao, "Controlling the shape of soft robots using the koopman operator," in *2023 American Control Conference (ACC)*. IEEE, 2023, pp. 153–158.

[14] E. Almanzor, F. Ye, J. Shi, T. G. Thuruthel, H. A. Wurdemann, and F. Iida, "Static shape control of soft continuum robots using deep visual inverse kinematic models," *IEEE Transactions on Robotics*, vol. 39, no. 4, pp. 2973–2988, 2023.

[15] M. O'Connell, G. Shi, X. Shi, K. Azizzadenesheli, A. Anandkumar, Y. Yue, and S.-J. Chung, "Neural-fly enables rapid learning for agile flight in strong winds," *Science Robotics*, vol. 7, no. 66, p. eabm6597, 2022.

[16] R. Garnett, *Bayesian optimization*. Cambridge University Press, 2023.

[17] C. E. Rasmussen, "Gaussian processes in machine learning," in *Summer school on machine learning*. Springer, 2003, pp. 63–71.

[18] J. Son, S. Gupta, and G. Tan, "Bayesian optimization in high dimensional input space," in *Proceedings of the 9th EAI International Conference on Simulation Tools and Techniques*, 2016, pp. 18–27.

[19] Y. Ansari, M. Manti, E. Falotico, Y. Mollard, M. Cianchetti, and C. Laschi, "Towards the development of a soft manipulator as an assistive robot for personal care of elderly people," *International Journal of Advanced Robotic Systems*, vol. 14, no. 2, p. 1729881416687132, 2017.

[20] B. Chen, R. Kwiatkowski, C. Vondrick, and H. Lipson, "Fully body visual self-modeling of robot morphologies," *Science Robotics*, vol. 7, no. 68, p. eabn1944, 2022.