Static Shape Control of Soft Continuum Robots using Deep Visual Inverse Kinematic Models

Elijah Almanzor[1], Fan Ye[1], Jialei Shi[2], Thomas George Thuruthel[3], Helge Wurdemann [2] and Fumiya Iida[1]

***Abstract*—Soft continuum robots are highly flexible and adaptable, making them ideal for unstructured environments such as the human body and agriculture. However, their high compliance and manoeuvrability make them difficult to model, sense, and control. Current control strategies focus on Cartesian space control of the end-effector, but few works have explored full-body control. This study presents a novel image-based deep learning approach for closed-loop kinematic shape control of soft continuum robots. The method combines a local inverse kinematics formulation in the image-space with deep convolutional neural networks for accurate shape control that is robust to feedback noise and mechanical changes in the continuum arm. The shape controller is fast and straightforward to implement; it takes only a few hours to generate training data, train the network, and deploy, requiring only a web camera for feedback. This method offers an intuitive and user-friendly way to control the robot's 3D shape and configuration through teleoperation using only 2D hand-drawn images of the desired target state without the need for further user instruction or consideration of the robot's kinematics.**

I. INTRODUCTION

Soft continuum robots bend continuously along their length via elastic deformation, making them highly flexible and adaptable [1]. They provide high dexterity and manoeuvrability over constrained unstructured spaces using fewer actuators and simple control strategies [2]. This makes them ideal for robotic exploration [3] and inspection [4], [5] in constrained environments, especially in medical applications [6]–[10]. An example of this type of robot is the STIFF-FLOP, designed for minimally invasive robotic surgeries (see Fig. 1). However, the high mechanical adaptability and dexterity of soft continuum robots present numerous challenges in their control.

Unlike the control of rigid robotics, the control of soft robots is non-trivial [11], [12]. Soft continuum arms exhibit large to infinitely many degrees of freedom with elastic materials that are highly non-linear, making accurate analytical modelling an arduous task. Significant variabilities in their design and actuation make the development of a general modelling framework difficult [13], [14].

Most controllers for soft continuum robots are based on some kinematics model. Kinematic controllers take on a steady-state assumption for the soft manipulator, where they are treated as quasi-static structures. The most common modelling technique used for soft continuum arms is the Constant Curvature (CC) model, where each section of the arm is assumed to bend with constant curvature. Hence, the kinematic shape of the manipulator can be represented by the arm length, curvature and its angle [15]–[20]. Piece-wise Constant Curvature (PCC) is an extension of CC, which treats the shape of each section itself as a series of finite curved links [21], [22]. These models, however, assume that the manipulator or its sections are uniform and symmetrical with negligible external loads or torsion [15]. More complex models with increased accuracy have also been proposed such as the Variable Constant Curvature [23], [24] (VCC), the Spring-Mass-Damper model [25], the Cosserat Rod [26], [27] and beam-theory models [16], [28] and Finite Element models (FEM) [29]–[31].

Once the forward kinematic model is developed, controllers can be achieved by inverting the model. For CC models, this can be done using several differential inverse kinematics (IK) based approaches [19], [20], [32]. Similar differential inverse kinematics-based controllers have also been developed for

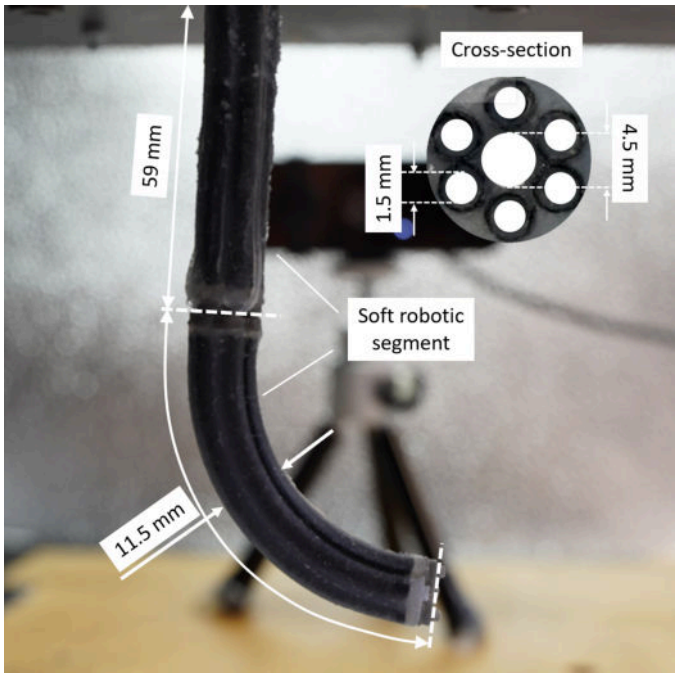


Fig. 1: Prototype of the soft continuum robot arm used in this paper emerged from the EU-project STIFF-FLOP. This version has two segments, that have an overall diameter of $11.5\,\mathrm{mm}$ and a length of $59\,\mathrm{mm}$. Actuation will result through pneumatic air pressurisation of three chamber pairs, each has a diameter of $1.5\,\mathrm{mm}$.

[1]The Bio-Inspired Robotics Lab, Department of Engineering, University of Cambridge, UK. [2]Department of Mechanical Engineering, University College London, UK. [3]Department of Computer Science, University College London, UK. email: eda26@cam.ac.uk; tg444@cam.ac.uk;fi224@cam.ac.uk

PCC [33], [34] and VCC [23] kinematic models. Alternatively, optimization-based methods can be used for solving the inverse problem [22]. Camarillo et al. [28] were able to achieve configuration open-loop control by direct inversion of the beam theory kinematics. FEM has demonstrated promise in accurately modelling and controlling soft systems [29]–[31]. Nonetheless, their application necessitates pre-existing computational models of the robot, which may not be readily accessible, and the need to fine-tune parameters, such as determining the optimal number of nodes, to maintain sufficient accuracy without sacrificing real-time control frequency.

Kapadia et al. [17], [18] developed the first dynamic closed-loop controller using a CC-based kinematic model and an Euler-Lagrangian dynamic model for a soft arm, capable of controlling the robot in task space. Another example which uses the CC model with dynamics is the work by Falkenhahn et al. [35]. They developed a dynamic controller which optimised a trajectory in terms of time and actuator jerk. They also developed a controller with feedback linearisation capable of fast trajectories that minimises dynamical errors [36]. Della Santina et. al [34] extended PCC-based point mass models and showed impedance control of the end-effector for interactions with an unstructured environment. Alqumsan et. al. [37] introduced a sliding mode dynamical controller for a simulated Cosserat Rod model. Spinelli et al. [38] utilised Model Predictive Control for the task space control of a pneumatic continuum robot, which utilised PCC with augmented Rigid-Body model assumptions.

These model-based controllers are, however, heavily reliant on the underlying analytical models, which are developed with numerous assumptions. These can cause large discrepancies between the model and the physical robot if factors such as material hysteresis, friction, mechanical asymmetry, torsion, fabrication imperfections and other external loads are not considered or compensated for [27], [39]. As a result, researchers have taken an interest in controllers that are hybrid, combining analytical models with learning-based approaches, as well as completely model-free controllers. These types of controllers utilize real-world data to build internal models and controllers, requiring minimal to no prior knowledge of the physical robot arm or the surrounding environment.

Regarding hybrid controllers, Braganza et al. [40] utilized a combination of a feedforward neural network and a nonlinear feedback component to create a control strategy that compensates for uncertain dynamics during trajectory tracking without requiring accurate knowledge of the continuum robot's dynamical model. Queißer et al. [41] combined feedback control with a feed-forward control that approximates the continuum arm's inverse dynamics under equilibrium. They used this technique for kinesthetic teaching of the robot's posture. Wang et al. [42] implemented a hybrid adaptive control approach that employs neural networks for learning the robot's inverse kinematics and online adaption of PID control parameters for path tracking. Bruder et al. [43] utilized Koopman-based Model Predictive Controllers (MPCs) to track trajectories when the manipulator has an unknown payload. Tang et al. [44] introduced an iterative learning Model Predictive Control (MPC) method for soft-bending actuators, using the iterative learning controller (ILC) to refine the model. Their results demonstrated that their approach outperformed ILC and MPC independently. Hanh et al. [45] developed a method which uses dynamic motion information to refine their FEM's visco-elastic parameters. Their approach enabled them to design an open-loop control strategy in simulation which was executed on a physical soft robot. However, their work requires the use of expensive motion capture systems.

For model-free controllers, Yip et al. [46] were one of the first to introduce a completely model-less controller. They achieved this through online empirical estimations of the robot's Jacobian. Although it allows the robot to navigate in unstructured environments, it is seemingly limited to the control of the robot's tip. Alambeigi et al. [47] estimated the Jacobian of a continuum arm using the optimisation Broyden update rule to manipulate a target point on a deformable material such that it corresponds to the desired point in their endoscopic camera's image space. Li et al. [22] employed an adaptive Kalman-filter controller for trajectory tracking. Fuzzy-logic controllers have also been explored [48].

Giorelli et al. [49], [50] were the first to implement feed-forward neural networks for learning a one-to-one IK mapping of the actuator space to the task space. Hence, their controller is incapable of tolerating redundant solutions for the same tip position in real soft continuum arms. Rolf et al. [51] proposed the use of goal babbling for obtaining movement samples, which can be used to bootstrap learnt IK solvers. In the process of generating the samples, a redundancy weighting scheme is applied to encourage smooth solutions for target points in the task space. One major issue with learning the IK is the non-uniqueness of IK solutions which are enclosed in a concave set, which makes learning global IK intractable. Thuruthel et al. [52] proposed a formulation which can achieve direct inversion of FK through linearisation at the current state. Sahoo et al. [53] expanded on this work by employing a meta-learning approach to reduce the training sample required for adapting the network to unknown tip-loading conditions. Distal learning is another method for inverting the kinematics of a redundant robot and has been used for soft robots by Melingui et al. [54]. Learning-based approaches have been notably effective for dynamic control of soft robots, where they have been employed for learning an accurate forward dynamics model [55]–[57] or for directly learning a closed-loop control policy using reinforcement learning [58]–[60]. Irrespective of the method used to control the soft continuum robot, be it model-based, model-free, or hybrid models, control approaches require expensive motion-tracking sensors to achieve closed-loop control [19], [43], [52].

A. Related Work

All the works mentioned above focused mainly on task space/end-effector control of the soft continuum arm. Using analytical models for controlling the robot's shape in configuration space, through joint space control, requires making

simplifying assumptions. As a result, the robot's actual shape may differ from that predicted by the model. Complete authority over the shape of the continuum robot, however, is critical in scenarios where minimal collisions or whole-arm path planning are necessary, such as in teleoperated endoscopy or minimally invasive surgery of the human body [61], [62].

Shape control of soft continuum robots is a straightforward problem when simple geometric models are used. Bajo et al. [63] showed that the use of both extrinsic and intrinsic sensory information can lead to better regulation and shape tracking performance, mitigating the effects of actuation coupling for a CC model. Wang et al. [39] extended the CC theory with the Kirchoff elastic rod theory for open-loop shape control of a continuum robot designed for in-situ aero-engine maintenance. Various works have implemented open-loop follow-the-leader techniques where desired arm configurations based on CC are traced by extendable robotic arms [64], [65]. Similarly, Santoso et al. [66] investigated the use of CC with a damped least-squares Jacobian for the closed-loop control of an origami-based continuum robot to grow into desired shape configurations.

Bern et al. [67] created soft "plush" robots which they modelled using 2D FEM. To control the robots and move them to a desired 2D position, they developed a simple and intuitive method of dragging mesh nodes. They then extended this work to the locomotion control of a 3D soft cable-driven quadruped robot [68]. The feasibility of applying their method to any type of intricate soft robot comprising diverse and functional materials which give rise to highly irregular structures, and that employ different types of actuation such as pneumatics or magnetics [69], remains uncertain. Such robots would necessitate a more refined mesh, a greater number of nodes, and an appropriate computational model [70].

Ouyang et al. [71] developed a control approach which used shape correspondence to command the shape of the robot to a hand-drawn curve provided by the user. Their controller uses an online estimation of the Jacobian, with the aim of minimising the error between the desired curve approximated by cubic spline interpolation and feature points located on the robot. The shape of the robot is then approximated by PCC. Hand-drawn shapes are much more intuitive for users operating the continuum robot.

The development of completely 3D kinematic shape controllers with an intuitive interface would enhance the versatility of continuum robot arms by enabling direct control to more complex pose configurations [64], [66]. CC and PCC models approximate a large degree-of-freedom arm with a lower dimensional and discretised representation. Extending these control strategies to more complex geometric models like Cosserat Rod, however, is not trivial. Although Finite Element models show promise, they tend to grow in complexity when dealing with non-homogeneous structures and functional materials with various forms of actuation methods. Likewise, they also require pre-existing computational models. Inversion of fully 3D kinematic shape models scalable to any arbitrarily complex robots with appropriate representations of the target shapes is a challenge yet to be addressed in this field.

To the best of our knowledge, this work presents the first demonstration of a model-free shape controller for a soft continuum robot. We propose a deep visual inverse kinematic model for the shape control of a soft redundant continuum robotic manipulator. Based on steady-state assumptions and differential inverse kinematics, a unique methodology of representing the state of the soft robot as images are presented. The method offers relatively straightforward and fast learning of the differential IK without requiring *a priori* knowledge about the arm or its environment, such as shape, size and geometry, actuators, hysteresis, friction and internal loads, mechanical asymmetry, and torsion. The method takes only 3 hours to generate motor-babbling data (generation of motion through random actuation values), train the network, and deploy on a physical continuum robot.

Furthermore, minimal sensing is needed, requiring only a simple colour camera to learn the mapping between the actuator space and the proposed configuration space representation. The generalising ability of the deep visual model enables a more user-friendly and intuitive method of controlling the soft manipulator, simply by drawing the desired target shapes without any prior training on the dataset, or the need for further user instruction or consideration of the robot's kinematics.

The next section first explains the formulation of the learning algorithm in task space, followed by the description of the image-based extension. Section III details the simulated spring-mass-damper continuum arm on which the image-based method is trialled to show empirical evidence of the image-based method's efficacy, which is given in the section after. Section V then details the physical STIFF-FLOP manipulator and the experimental setup. This is followed by section VI showcasing the experimental results. The discussion and conclusions are given in section VII.

II. Model-Free Static Controller Theory

A. Learning Cartesian Kinematic Controllers

For soft robots, kinematic relationships can be formulated using steady-state models [72]. At steady-state conditions, the forward kinematics of a soft robot can be represented as a surjective function from the actuator space $\mathbf{q} \in R^n$ to the task space $\mathbf{x} \in R^m$.

$$\mathbf{x} = f(\mathbf{q}) \tag{1}$$

The actuator space $\mathbf{q}$ typically consists of the state of the actuators of the soft robot (e.g. pneumatic pressure, tendon forces, etc.). The task space $\mathbf{x}$ is typically represented by the Cartesian end-effector pose.

Learning the inverse of this forward mapping is not trivial because of the high redundancy in the system. Due to the high redundancy of a soft CR, there are infinite valid solutions to the inverse problem. Moreover, these infinite solutions do not form a convex set, making the direct learning of the inverse kinematics invalid [52], [73]. Direct inversion of the forward differential kinematics, however, can be done through

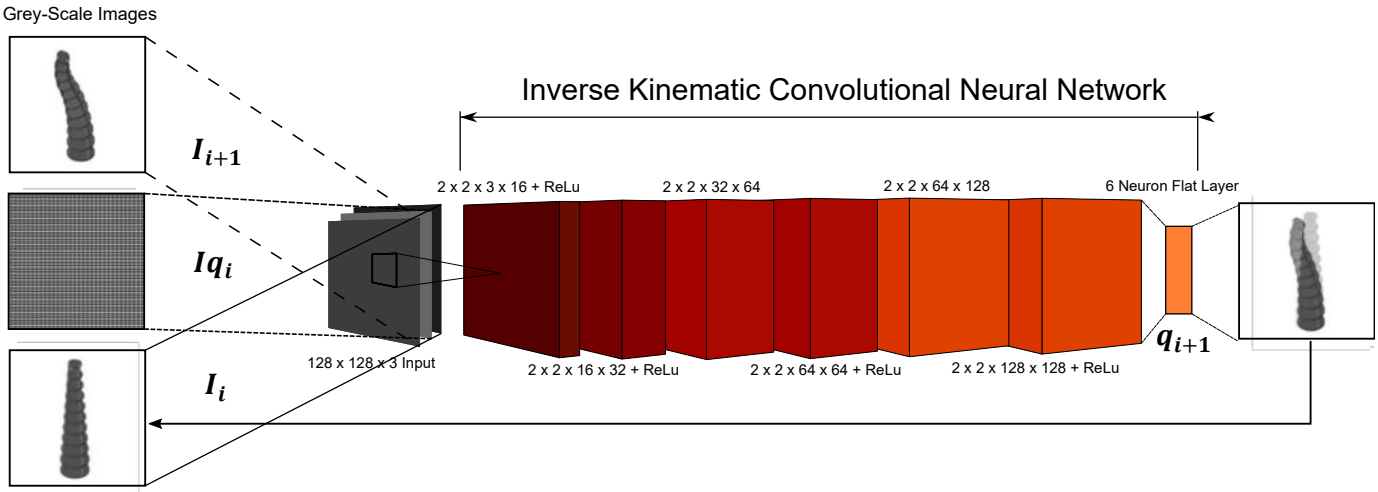


Fig. 2: The convolutional neural network-based kinematic shape controller. The deep network provides the control actions that iteratively take the robot to the target shape given the current shape and control action.

linearisation at some arbitrary feasible actuator state $\mathbf{q^0}$. The forward differential kinematics can be obtained by taking the derivative of the forward kinematics at this state ($\mathbf{q^0}$):

$$\dot{\mathbf{x}} = J(\mathbf{q^0})\dot{\mathbf{q}} \tag{2}$$

Where J is the Jacobian matrix that maps the actuator velocities $\dot{\mathbf{q}}$ to the end-effector velocities $\dot{\mathbf{x}}$ at state $\mathbf{q^0}$. Further discretization of Equation 2 using Taylor expansion (ignoring second order terms and above) allows for learning a local Inverse Kinematics [73].

$$\Delta\mathbf{x} \approx J(\mathbf{q^0})\Delta\mathbf{q} \tag{3}$$

To frame it as a learning problem, this expression can be expanded and rearranged as:

$$J(\mathbf{q_i})\mathbf{q_{i+1}} \approx \mathbf{x_{i+1}} - f(\mathbf{q_i}) + J(\mathbf{q_i})\mathbf{q_i} \tag{4}$$

$$\mathbf{q_{i+1}} = G(\mathbf{x_{i+1}} - \mathbf{x_i}) + \mathbf{q_i} \tag{5}$$

Allowing for the mapping $(\mathbf{x_{i+1}}, \mathbf{q_i}, \mathbf{x_i}) \rightarrow (\mathbf{q_i})$, where G is the inverse of the Jacobian matrix, $\mathbf{q_i}$ and $\mathbf{q_{i+1}}$ are the actuator states at the current and the next time steps respectively, and likewise for task space states $\mathbf{x_i}$ and $\mathbf{x_{i+1}}$. Training data can then be obtained through random actuator motion (motor babbling) ensuring spatial locality ($|\mathbf{q_{i+1}} - \mathbf{q_i}| < \mathcal{E}$) [52]. This data can then be learnt by a simple neural network. By providing the target points as $\mathbf{x_{i+1}}$ to the learnt network, the network outputs, $\mathbf{q_{i+1}}$ which brings the robot configuration closer to the target. Repeating the process with the updated $\mathbf{q_i}$ and $\mathbf{x_i}$ will eventually bring the Cartesian end-effector coordinates to the desired location, providing that the target location is not physically obstructed and is geometrically reachable, with a task-space tracking sensor that has reasonable accuracy. This is the learning equivalent of the resolved motion rate controller, which also makes the controller robust to inaccuracies in the learned representation of the Jacobian inverse [74].

B. Learning Kinematic Shape Controllers

The theory described above, which was developed for controlling the pose of an end-effector in Cartesian space, can also be applied to other types of task-space representations. Although control of the tip position is sufficient for many tasks, due to the redundancy of the system, the resulting shape of the soft robot is not within our control authority and is heavily influenced by the initial configuration and environmental constraints. This is problematic in scenarios where a certain specific shape is required in order to reduce environmental interactions, such as endoscopic operations [61], [62], [75]. Additionally, there are several tasks where shape control is more intuitive for the user, such as in Agritech [76] and other constrained environments [77], [78].

Due to the large degrees of freedom for a soft continuum robot, there is no single method for shape parametrization. Constant curvature models have largely been used for shape parametrization, but are based on several simplifying assumptions that parametrize the shape of a soft robot using low-dimensional representations. In this article, we propose the extension of the data-driven inverse kinematic controller (see Fig. 2) to take image states as a high-dimensional hyper-redundant task space representation $\mathbf{I}$. The static images are a function of the actuator state $\mathbf{q} \in R^n$ and the environmental conditions η.

$$\mathbf{I} = \mathbf{I}^q + \mathbf{I}^\eta \tag{6}$$

The region of the image that is dependent on the actuator variables obeys the static relationships as described in the previous section.

$$\mathbf{I}^q = f(\mathbf{q}) \tag{7}$$

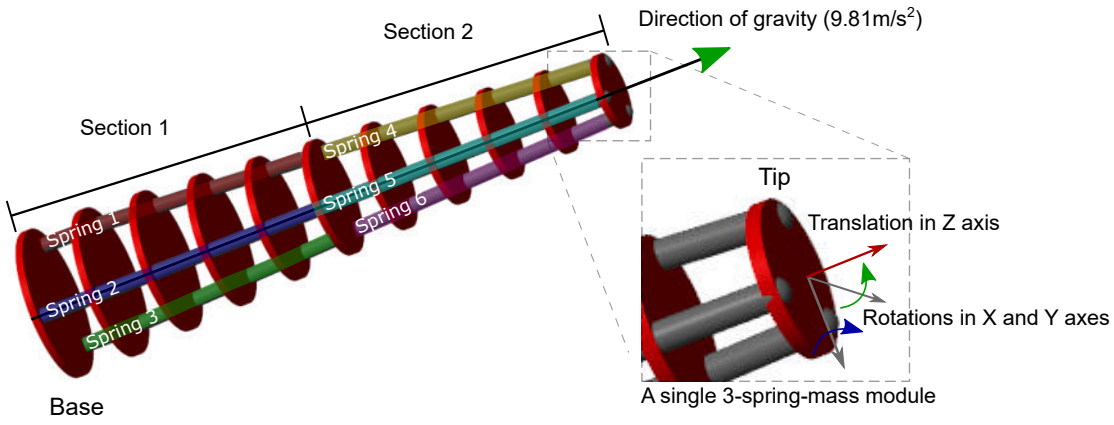


Fig. 3: Modular mass-spring-damper continuum arm used for validating the learning-based shape controller.

$$\dot{\mathbf{I}}^{\mathbf{q}} = J(\mathbf{q^0})\dot{\mathbf{q}} \tag{8}$$

$$\mathbf{q_{i+1}} = G(\mathbf{I^q}_{\mathbf{i+1}} - \mathbf{I^q}_{\mathbf{i}}) + \mathbf{q_i} \tag{9}$$

If $\mathbf{I^q}_{\mathbf{i+1}}$ and $\mathbf{I^q}_{\mathbf{i}}$ can be extracted from the observations of $\mathbf{I_{i+1}}$ and $\mathbf{I_i}$, then an image based kinematic controller can be developed. Assuming that a deep network can extract this information implicitly, the mapping $(\mathbf{I_{i+1}}, \mathbf{Iq_i}, \mathbf{I_i}) \rightarrow (\mathbf{q_{i+1}})$ can be sampled and learned as before. Note that $\mathbf{Iq_i}$ is the transformation of the vector $\mathbf{q_i}$ into a 2D image array to ease the concatenation of $\mathbf{I_{i+1}}, \mathbf{Iq_i}$ and $\mathbf{I_i}$ as network inputs. Varying lighting conditions, background noise and camera displacements, in theory, would not affect the mapping accuracy as long as $\mathbf{I^q}$ is visible. Occlusions, hence, would affect the accuracy of the mapping. Deep convolutional neural networks are used to learn the image-based differential inverse kinematics mapping. These networks are excellent at extracting image features from noisy images while exhibiting some shift and scale invariance [79]–[81].

III. Simulation Environment

A. Simulation Arm

A simulation environment is used to test and validate the image-based kinematic shape controller. We created a simulated continuum arm in MATLAB & Simulink using the Simscape library.

The continuum arm comprises multiple individual modules. As shown in Fig. 3, each module is a 3-spring-mass structure, with three springs attached concentrically to a disk with mass. Each spring is a force-driven spring-damper object defined by an equilibrium position, and stiffness and damping coefficients. The stiffness and damping coefficients are set to relatively high values 10 kNm^{-1} and 10 Nsm^{-1} respectively, to minimise any dynamical behaviours. They are driven by external force inputs from a controller. Each disk has three degrees of freedom. A translation in Z axis and rotations about X and Y axis, relative to the previous disk. For our simulation, we mounted ten modules in total to form two sections containing five modules. For each section, consecutive spring receive the same control signal. For example, spring objects in Spring 1 receive the same control signals. This was done to conceptually match the real STIFF-FLOP arm shown in Section V-A. Our simulation robot was intentionally created with a distinct geometry, materials, and actuation mechanism compared to the STIFF-FLOP robot to demonstrate the image-based inverse kinematic solver's capability in controlling vastly different continuum arms. Gravity was configured to point from the base to the tip, to simulate a downward mounting arrangement. However, we deliberately kept the orientations between the robot and the camera to be different in simulation and in reality to demonstrate that the controller can effectively operate even when the camera and the robot have different relative orientations.

B. Simulation Experimental Setup

Training data were obtained through a quasi-static motor babbling method [73]. The vector $\mathbf{q_i}$ holds the individual actuation values for each of the six springs in the simulation robot. The actuation input applied to each of the six springs was limited to $150N$ to produce an arm configuration with sufficient complexity without causing instability. Random shape trajectories were generated by adding stochastic force values $\Delta\mathbf{q}$ to the current force inputs that satisfy $0 \leq \mathbf{q_i} \leq 150N$ every 5 seconds. A Sigmoid smooth-step function $S(t)$ (Equation 10) was multiplied to $\Delta\mathbf{q}$ to ensure smooth trajectories in the generated training data. This is done to ensure that the steady-state assumption used in deriving the theory remains valid.

$$S(t) = \frac{a}{1 + \exp\left(\frac{-\alpha(t-t_{50})}{t_r}\right)} \tag{10}$$

The parameters a, t_{50}, t_r are used to make the smooth-step function reach the $\Delta\mathbf{q}$ values over a time period of 3 seconds,

with $0 \leq t \leq 5$ seconds such that the newly generated random shape is reached when the value $S(t)$ is equal to 1. The values used for the parameters a, t_{50}, t_r are 1, 2, and 1, respectively. These were determined empirically. Saturation after 3 seconds was done to include training examples that stabilise to a final shape.

12000 seconds of motor babbling data was gathered. State images $\mathbf{I}$ were obtained at a rate of $1Hz$ via screen capture giving 12000 data samples. The image states were converted to grey-scale and down-sampled to 128×128 to speed up learning. Actuator image inputs $\mathbf{Iq_i}$ were obtained by repeatedly copying the current actuation signals $\mathbf{q_i}$ into a 2D image array. This allows us to concatenate the image state and actuator state easily. The next state $\mathbf{I_{i+1}}$, current state $\mathbf{I_i}$, and actuator inputs $\mathbf{Iq_i}$ were resized to a height and width of 128×128, followed by concatenation to form an $128 \times 128 \times 3$ input. The six-element vector $\mathbf{q_{i+1}}$ was kept as is for the data regression labels.

C. Model-Free Deep Visual Network Inverse Kinematic Solver

The controller uses a bespoke architecture to learn the mapping $(\mathbf{I_{i+1}}, \mathbf{Iq_i}, \mathbf{I_i}) \rightarrow (\mathbf{q_i})$ (see Fig. 2). The architecture was built systematically by incrementally adding convolutional and ReLu activations layers until the root mean squared error on a small 10% subset of the training data no longer improved. The same architecture is then used for real-world experiments. The network has six regression node outputs for controlling the six simulation springs $\mathbf{q_{i+1}}$. The control loop for the real arm is given in Fig. 11, which is similar to the simulation control block. The only difference is that the control outputs are fed into the simulation spring objects rather than through pneumatic regulators.

Training data was split into 10% and 90% for training and testing, respectively. The network was trained for 1500 epochs using the Adam optimizer with an initial learning rate of 0.005 at a learning drop rate of 0.99 which drops at every 100 epochs. The state images I_{i+1} and I_i were augmented with random speckle noise, random translations and rotations, and random occlusions to obstruct the visibility of the target and feedback states during training to make the controller more robust to noise. The data was not normalised and no overfitting was observed. For the experiments, the controller is run at $10Hz$, even though the training data was obtained at $1Hz$. It was observed that this strategy led to smoother trajectories and faster target convergence.

IV. Simulation Results

Four simulation experiments were undertaken. During the experiments, no online changes were made to the network weights.

The first experiment was to qualitatively verify the image-based IK controller for random various target shapes taken randomly from the validation dataset, shown in Fig. 4. During the experiments, the arm starts from a home configuration where it is initially straight and all the actuation values are set to zero. The controller requires multiple steps to converge to the target shape. Hence, for each case, the controller ran for 60 seconds to obtain the final resulting shape state. Note that the IK solutions provided by the controller are not unique. It is dependent on the initial configuration of the robot, and hence the controller is more robust to model inaccuracies.

The second experiment was to determine the robustness of the learned controller to feedback $\mathbf{I_i}$ translational and rotational noise to simulate the effect of the camera being displaced out of position and orientation after learning. Twenty-five more shapes were gathered to obtain an average performance of the controller. *The overlayed images for these remaining shapes are given in the supplementary material.* For the translation noise, the feedback images $\mathbf{I_i}$ were translated in the X and Y direction. Black pixels were used to pad the images after translation (see Fig. 5b for an example). Results are given in Fig. 6. For the rotational noise, the feedback images are rotated clockwise and counter-clockwise. As before, black pixels were used to pad the images after rotation (see Fig. 5c for an example). The results are given in 7. The state errors were obtained by the image pixel subtraction of the final resulting state $\mathbf{I_i}$ to the target state $\mathbf{I_{i+1}}$ images and averaging the absolute of the error image. Note that the non-augmented final feedback images were used to gather state errors.

Figures 6 and 7 demonstrate that the controller's accuracy is greater in the absence of noise, as evidenced by lower means and smaller standard deviations. Despite the increased severity of both types of noise, the fact that their standard deviations significantly overlap suggests that the controller's performance remains unchanged. However, the higher variances indicate that the performance of the controller with noise is also dependent on the target shape. Note that no direct intervention is applied to the network to reduce the error between the target and the feedback. The error reduction arises implicitly based on the learning architecture and the sample data used for learning.

The third experiment was to verify the robustness of the controller when the feedback images were partially occluded. For the same twenty-five shapes used in the third experiment, a black box is added to the feedback state $\mathbf{I_i}$ to simulate partial occlusion. Although the black box is placed randomly, the same pixel position was used for all twenty shapes (see Fig. 5d for an example of this occlusion). Results are given in Table I. From this table it can be seen that the standard deviations between no noise and occlusion noise overlap greatly, indicating that the control is robust even when the feedback is partially-occluded.

The fourth experiment was to verify the robustness of the controller to viscoelastic changes in the soft material properties of physical arms such as creep or stress relaxation. These can occur from usage over time due to periodic strains and stresses from actuation, which can induce permanent deformations [82]. Thus, softening the material over time. To model this behaviour, we tested the controller on simulations with reduced spring stiffness. The original simulation stiffness $10kNm^{-1}$ was reduced in increments of 5% up to 20%

Fig. 4: Examples of target (in red) and final resulting states (in cyan) using the proposed shape controller. Intersecting regions are shown in black. The images are taken directly from the convolutional network inputs of size 128×128.

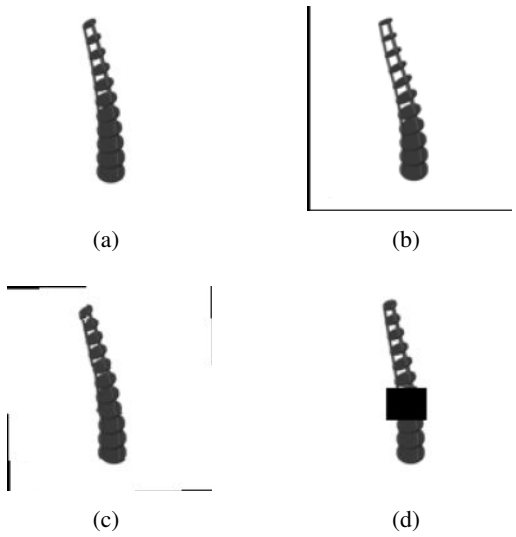


Fig. 5: Examples of the added noise for experiment two (with translational and rotational) and three (occlusion noise). (a) Desired target shape. (b) Final feedback state with translational noise in the X and Y directions. (c) Final feedback state with rotational noise. (d) Final feedback state with partial occlusion.

TABLE I: Average and standard deviations (σ) of the final state errors for twenty-five shapes without noise and with the occlusion noise (see Fig. 5d). The state errors are measured by the image subtraction between state I_i and I_{i+1} and averaging the absolute error.

	Without Noise	***With Occlussion Noise***
Average State Error	$2.40, \sigma \pm 1.00$	$2.66, \sigma \pm 1.38$

reduction. Note that $10kNm^{-1}$ is the stiffness used to gather the motor babbling data. The same target shape was used for all simulation models with varying stiffness using one of the shapes in Fig. 4. For this particular target shape configuration, only springs 1, 2 and 6 need to be actuated. From Fig. 8, it can be seen that the image-based controller can compensate for the lowered stiffness values by generally reducing the required spring forces in order to achieve the desired target shape without requiring any further re-training. Interestingly,

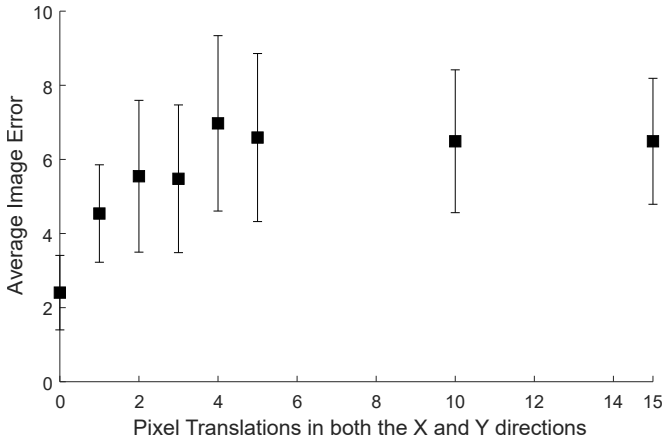


Fig. 6: Average final state errors for twenty-five shapes as the image state feedback is translated in the X and Y directions. The error bars are the standard deviations. The state errors are measured by the image subtraction between state I_i and I_{i+1} and averaging the absolute error.

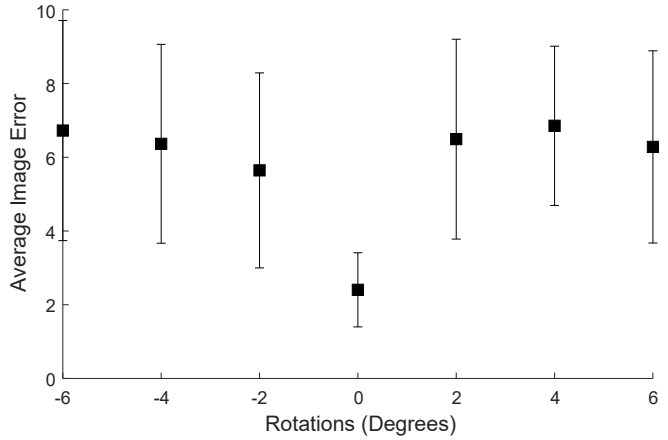


Fig. 7: Average final state errors for twenty-five shapes as the image state feedback is rotated up to positive (counter-clockwise) and negative (clockwise) 6 degrees. The error bars are the standard deviations. The state errors are measured by the image subtraction between state I_i and I_{i+1} and averaging the absolute error.

from Table II, it can be seen that slightly lower stiffness for the springs resulted in error reductions of -13.8% and -6.7% for the 5% and 10% stiffness reductions. The performance of the controller decreased rapidly at the 20% stiffness reduction increasing the final state error by 81%. This behaviour is unlikely to be indicative of the controller's performance on other shapes. Nevertheless, it shows the ability of the controller to accommodate changes in the model's mechanical properties (see the upper section of Fig. 8).

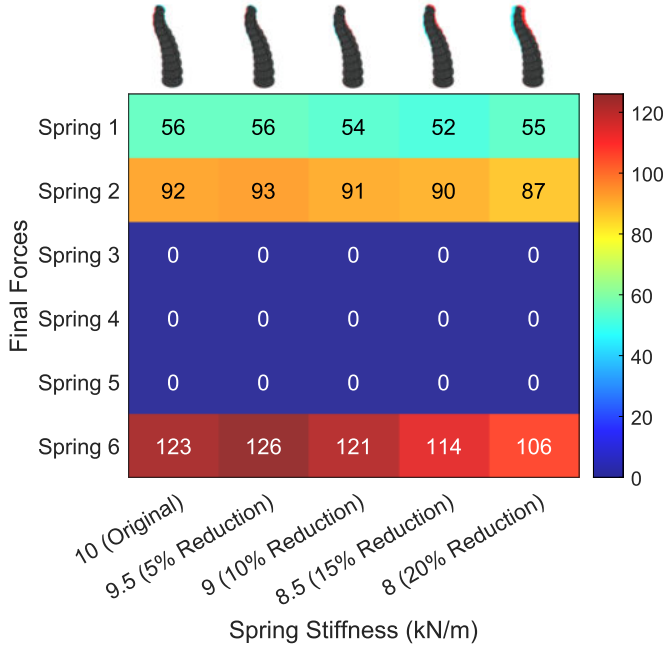


Fig. 8: Final state reached by the controller (top) as the simulation springs stiffness is decreased to model material creep. The state errors are measured by the image subtraction between state I_i and I_{i+1} and averaging the absolute error. Percentage differences in the final state errors relative to the original spring stiffness are given in II.

TABLE II: Percentage difference of the final state errors between decreased stiffness models and the original stiffness. The final state error for each stiffness is measured by the image subtraction between state I_i and I_{i+1} and averaging the absolute error.

	Simulation Model Stiffness $(\mathbf{kNm^{-1}})$			
	9.5	9	8.5	8
Percentage Error (%)	-13.8	-6.7	12.9	81.1

V. Physical Environment

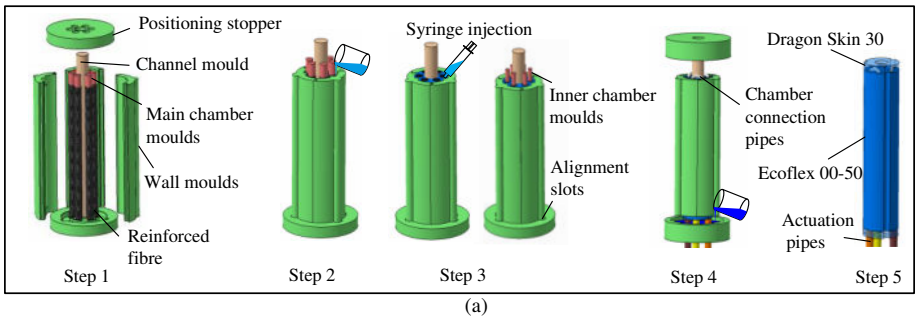
A. STIFF-FLOP Continuum Robot

Our proposed static shape controller is implemented on a miniaturised soft, pneumatically actuated manipulator to validate and demonstrate its efficacy. The fundamental design and manufacturing process (as shown in Fig. 9) has been introduced in an EU FP7 project called STIFF-FLOP [83]–[85]. Hence, the soft robot in this paper is referred to as the STIFF-FLOP manipulator, a cylindrical robotic device made of silicone (Ecoflex 00-50 Supersoft, SmoothOn), with six fully fibre-reinforced chambers. Two adjacent chambers are internally connected together via $1\,\mathrm{mm}$ silicone pipes and actuated as one chamber pair. The moulds are 3D-printed using Tough2000 resin (Formlabs Form 3). Following a five-step fabrication process (as illustrated in Fig. 9(a)), the final robotic manipulator has a diameter of $11.5\,\mathrm{mm}$. A central working channel with a $4.5\,\mathrm{mm}$ diameter is preserved for feeding through instruments, e.g., to conduct surgical tasks. Details on the dimension of the robot can be found in Fig. 9(b). Two or more manipulators can be connected in series via 3D-printed connection plates, as shown in Fig. 9(c).

B. Experimental Setup

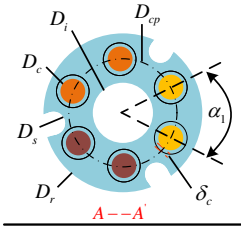
The STIFF-FLOP continuum arm is mounted upside down on a table platform (see Fig. 10). Six SMC regulator valves rated with a maximum pneumatic output of $0.5MPa$ are used to actuate the robot's air chambers. An Arduino Mega is interfaced with six MCP4725 DACs through a multiplexer for providing the $0-10V$ analogue voltage control inputs required by the valves. A Lenovo Webcam with a resolution of 1920×1080 is used to capture the image states of the robot. A workstation with a RTX 3070 graphics unit and an Intel i7 processor is used for data processing and learning. The whole platform is placed inside a photo-booth for better lighting conditions.

Similar to the simulation arm, training data was obtained through the described quasi-static motor babbling algorithm. The training data was not augmented to speed up the learning progress. The maximum control voltage was set to $3V$ for all chambers which correspond to $150KPa$, the chamber's pressure limit. As before, random shape trajectories were generated by adding stochastic pressure values $\mathbf{\Delta q}$ multiplied by the Sigmoid function (Equation 10) to the current pressure value $\mathbf{q_i}$. The parameters a, t_{50}, t_r all had the value of 1. These were also determined empirically. A new trajectory is generated every 5 seconds. Due to the multiplexing of the DACs, the fastest control frequency allowable is at $2.7Hz$. The STIFF-FLOP has a tendency to bend more in its lower module compared to the upper section due to the influence of gravity on the whole assembly. The upper module also houses the three actuation pipes of the lower section which makes it stiffer. Similar to the simulation goal-babbling, state images $\mathbf{I}$ were therefore obtained at a rate of $2.7Hz$ and were subsequently down-sampled to $0.27Hz$ for the training data, resulting in approximately 1200 training images. It took approximately an hour and 30 minutes to gather the whole training data including the setup time. The training took approximately 2 hours using the workstation. The network takes an average of 0.002 seconds to process the image inputs. The state images were cropped, grey-scaled, and resized to 128×128 followed by concatenation to form the individual training samples. As done previously with the simulation



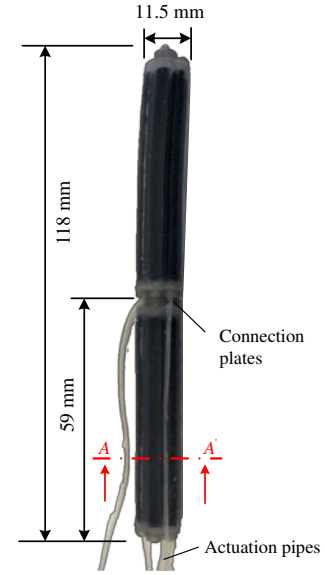
(a)

CROSS-SECTIONAL DIMENSION OF THE SOFT ROBOT



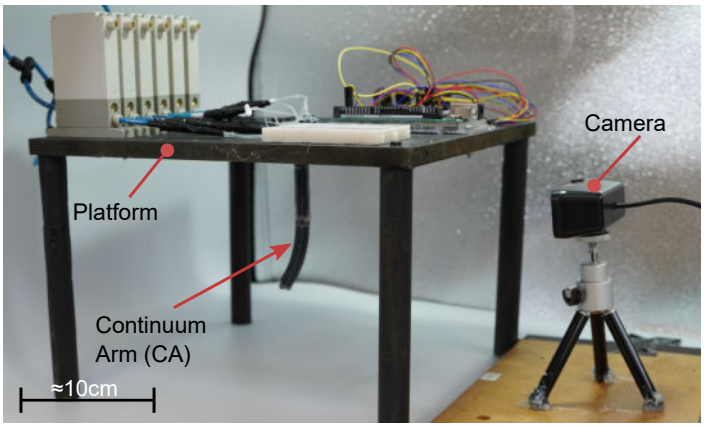
Symbol	Description	Value	Unit
D_r	Diameter of the soft robot.	11.5	[mm]
D_i	Diameter of the central channel.	4.5	[mm]
D_c	Diameter of the actuation chambers.	1.5	[mm]
D_{cp}	Diameter of the chamber position.	8.0	[mm]
D_s	Diameter of the slots.	1.5	[mm]
α_1	Angle between two adjacent chambers.	60.0	[deg]
δ_c	Wall thickness of the chambers.	0.5	[mm]

(b)



(c)

Fig. 9: Details of the miniaturised soft robot referred to as STIFF-FLOP manipulator. (a) The five-step fabrication process. Step 1: 3D-printed moulds are assembled, with the fibre densely wrapped around the main chamber moulds; Step 2: Ecoflex 00-50 is poured to the mould assembly to make the main body of the robot; Step 3: Ecoflex 00-50 is injected to the chambers after the removal of the main chamber moulds, smaller moulds are inserted to make the inner layers of the chambers; Step 4: the bottom and top sides are sealed using Dragon Skin 30, after adding the actuation pipes and connecting two adjacent chambers; Step 5: the moulds are taken apart to complete the fabrication. (b) The dimensions of the final robot prototype. Three slots are reserved for positioning the actuation pipes. (c) A two-segment robotic manipulator connected in series



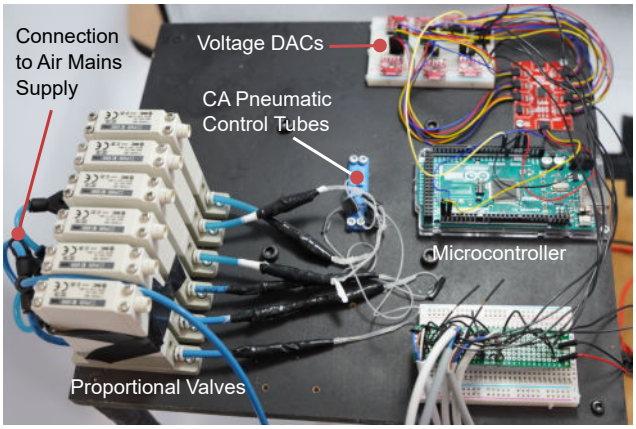


Fig. 10: Experimental setup for controlling the STIFF-FLOP continuum arm.

setup, the output of the network, which corresponds to the six actuator inputs for the next time step $\mathbf{q_{i+1}}$, is used to update $\mathbf{q_i}$. This is then transformed into a 2D matrix of size 128×128 (through repeating values) and inserted as a layer between states $\mathbf{I_i}$ and $\mathbf{I_{i+1}}$ making it feasible to input it back into the convolutional network controller.

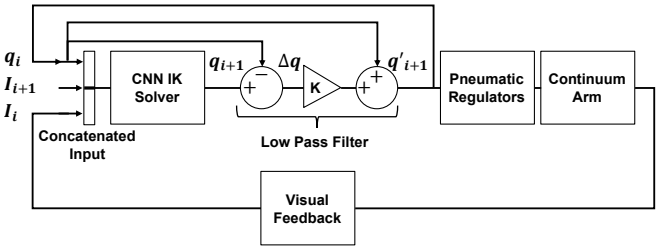


Fig. 11: Control diagram of the closed-loop kinematic shape controller.

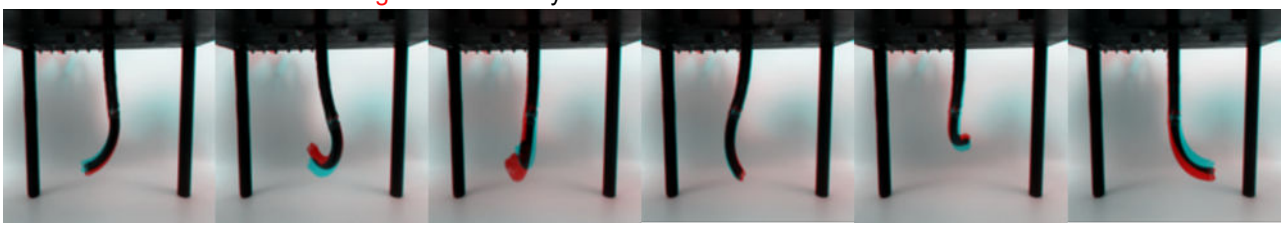


Fig. 12: Examples of the target (in red) and final resulting states (in cyan) using the proposed shape controller on the STIFF-FLOP arm. Intersecting regions are shown in black. The images are taken directly from the convolutional network inputs of size 128×128. *See the supplementary video for the robot trajectories.*

The same network architecture and training parameters were used, however, the network was only trained for 500 epochs due to the lower number of training samples. The complete control diagram of the experimental setup with the proposed CNN IK solver is shown in Fig. 11. A low pass filter is used on the output of the IK solver, where the difference between the predicted force $\mathbf{q_{i+1}}$ and the current force $\mathbf{q_i}$ is multiplied by a small gain value, which is then subsequently added to $\mathbf{q_{i+1}}$ to form $\mathbf{q'_{i+1}}$. The low pass filter was used to ensure that the arm remains quasi-static during movement. The current state $\mathbf{I_i}$ is obtained via the camera feedback. $\mathbf{q'_{i+1}}$ is also fed back as the current force inputs for the next iteration.

VI. Experimental Results

To test and validate the performance of the image-based IK solver on the real STIFF-FLOP arm, four experiments were conducted. The first was to qualitatively validate the controller for six random trajectories taken from the validation dataset. Similar to the simulation experiments, the arm starts in a straight home configuration where all the pressure actuation input are set to $0V$. Fig. 12 shows the ability of the solver to generalise IK solutions and reach arbitrary target shapes when starting from the home position. The CNN IK controller is able to accurately determine the correct pose of the end-effector, such as pointing towards or away from the camera, using only the low-resolution 2D input images and the position of the grey tip of the robot (see Fig. 9). This grey tip essentially acts as a feature point. Non-uniqueness of the 3D space projection to the 2D image plane can occur when the robot is bending directly away from the camera (where the tip is fully occluded by the body), however, this issue can be addressed by using additional feature points and multiple camera views in future work.

Fig. 13 shows the average error for the six target shapes over time. The errors were obtained by using Otsu-thresholding [86] on $\mathbf{I_{i+1}}$ and $\mathbf{I_i}$ to obtain their masks, and subtracting the difference (see Fig. 15b). This was done to remove the effects of noise caused by the flicker of the camera and the slight variances in the lighting conditions. These errors were then normalised and averaged. This metric, however, does have limitations. As this metric uses image masks, in some cases such as when the robot is facing directly towards the camera, non-unique 3D projections can also occur. However, the six target shapes shown in Fig. 12 do retain their uniqueness when projecting from 3D to 2D, even after obtaining the image mask. For instance, a posture bending towards the camera will appear larger. This makes the masking metric suitable for evaluating the performance of the controllers on the real robot, and using multiple camera views will enhance its robustness. In the future, work will be done to further improve this metric by incorporating 3D shape sensing technologies.

From Fig. 13, it can be seen that on average, the error quickly converges at around 10 seconds or 27 time-steps. The relatively high error standard deviations compared to the repeatability test in the following experiment (see Fig. 15c) shows that the accuracy of the controller is dependent on the desired target shape. The starting average mask error between the robot's straight position and the target shape was 0.0354. At the final step, the average error value dropped to 0.0133, which represents a 63% decrease in error. This demonstrates that the control system was able to successfully match the target, resulting in a reduction in error as seen in Fig. 12. It is important to reiterate that error reduction arises implicitly based on the learning formulation. However, the error does not reach zero, as there are minute differences between the resulting final and the desired target states, as seen from Fig. 12. Factors such as the lighting condition, slight changes in the table position with respect to the camera, sub-optimal learning, non-linearities in the physical material properties such as hysteresis, imperfect air sealing, and the non-uniqueness of the solutions can contribute to the observed error.

Unlike the simulation environment, the real-world data is filled with noise and variabilities, even with the arm being contained in a semi-closed system. An example of this is the differing average lighting intensities between the target and feedback images for the six shapes given in Fig. 12, which can be observed from the histogram in Fig. 14. From this figure, it can be seen that the feedback states are generally darker than the desired target states. From the *supplementary video*, these variabilities are even more pronounced. Camera flicker was found to occur due to the frequency difference between the camera and the light source. Additionally, the supplementary video also shows that there are changes in the background shadows between the target and the camera feedback states

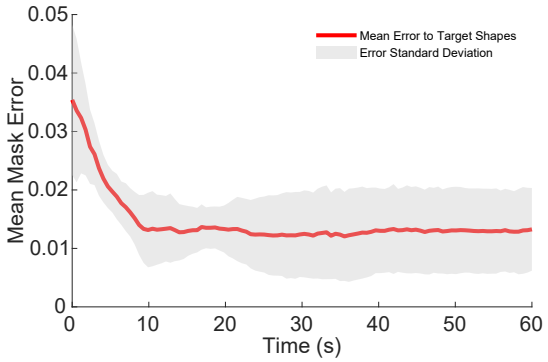


Fig. 13: Average mask error between the target and the feedback states for the six target shapes given in Fig. 12 over the duration of 60 seconds. Mask errors were obtained by the image subtraction between I_i (feedback) and I_{i+1} (target) states after Otsu-thresholding, followed by normalisation and averaging of the absolute image errors. On average, the controller converges to the final shape after 10 seconds.

due to misalignments. It is important that the controller is able to tolerate changes in the lighting condition and background states, and this is indeed observed in our experimental results.

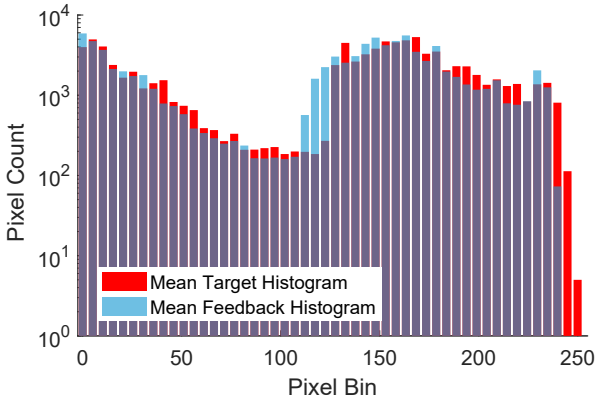


Fig. 14: Histogram of the average target and final feedback images for the six targets in Fig. 12. Pixel values of the target and final feedback states were grouped into sub-divisions of 5 resulting in 51 bins. These bins were averaged accordingly to the six target shapes. On average, the final feedback states are darker than the target states.

The second experiment was to validate the repeatability of the controller. Using one of the target shapes from Fig. 12, given in Fig. 15a, 10 repeatability tests were performed. Errors between the current state $\mathbf{I_i}$ and target state $\mathbf{I_{i+1}}$ were obtained using the Otsu-masking subtraction method described before (see Fig. 15b). Fig. 15c shows that although there are slight variances as it moves towards the target, the arm is able to converge to a low final error with reasonable precision, as indicated by the small standard deviations in error. This is also evident in Fig. 15d, which highlights the convergence of the valve control voltages to their final values. The low standard deviations of the final states control voltages after 60 seconds as given in Table III also proves the repeatability of the controller. Higher variability before convergence shows that the trajectory taken by the soft arm is not unique, even though the starting condition is the same. For this particular shape, it takes approximately 30 seconds which corresponds to 81 time-steps to reach the final shape. Three times longer than average. This is likely due to the non-unique solutions generated by the controller. The oscillations in the mean error are likely due to the imperfect pneumatic sealing, variation and lighting as well as the other factors mentioned previously.

TABLE III: Means and Standard Deviations of Final Pneumatic Voltages for the Repeatability Test in Fig. 15d

Valve	*Voltage Mean with Standard Deviation (V)*
Valve 1	2.44 ± 0.05
Valve 2	2.51 ± 0.03
Valve 3	1.70 ± 0.04
Valve 4	2.61 ± 0.16
Valve 5	0.69 ± 0.13
Valve 6	2.39 ± 0.18

The third experiment was to determine the effects of adding translation noise to the feedback $\mathbf{I_i}$ of the solver. In order to have full accurate control of the translation, image feedback states were shifted to the right rather than moving the camera itself. The feedback images are also cropped and padded by white 255 pixels as they are translated to the right. From Fig. 16, despite the loss in accuracy as the feedback states are translated, the controller remains capable of discovering a solution that captures the overall form of the target. Note that no data augmentation was done to the training data of the physical system to make the controller robust to these translations, and the emergent behaviour arises from the learning architecture and data structure.

The fourth and final experiment was to validate the generalisability and applicability of the controller. Here, we provide hand-drawn targets to the controller rather than actual images of the robot. Fig. 17 shows how the hand-dawn target shapes were made on top of the background image. First, a template without the arm was created in order to match the image environment. This template was then loaded into Microsoft paint. Random target shapes were then created using the line tools. These images are then given target to the robot. No retraining is done for the experiment.

Qualitative results for the six random hand-drawn target trajectories are given in Fig. 18. Even though the drawn targets are user-defined without consideration of the robot kinematics, the controller was still able to achieve qualitatively similar configurations. We have deliberately kept the hand-drawn target images ambiguous, omitting tip-like feature points, to test the generalizability of the network. The solver was also able to automatically determine a suitable pose for the robot, despite not being explicitly given in the target states. For example, in the third image in Fig. 18, the given target state is

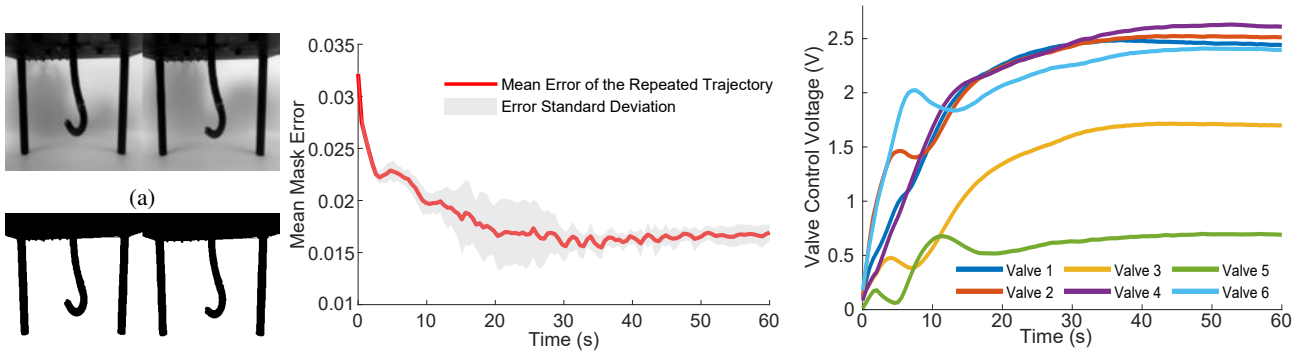


Fig. 15: Repeatability of the IK CNN solver for an example target shape. (a) Target state (left) and resulting final state (right). (b) Shape state masks for calculating the errors. (c) Average mask errors of 10 repeatability tests over 60 seconds. (d) Average control voltage of the 10 repeatability tests for the six proportional regulator valves. The final voltage means and standard deviations are given in Table III.

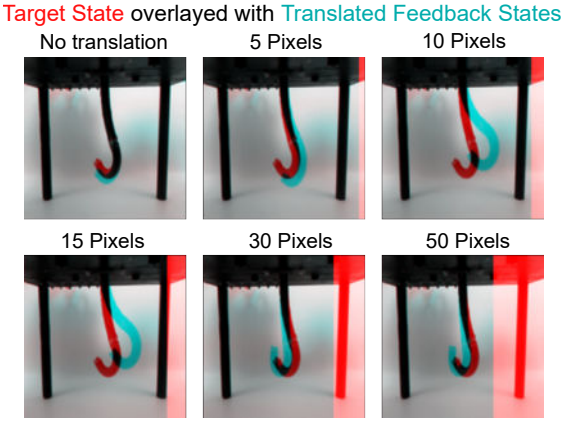


Fig. 16: Resulting final shape states for an example trajectory after translating the image feedback states I_i to the right in the X direction.

a short line. As there were no feature points, the controller was able to reasonably generalise this to a shape that is pointing away from the camera. Similar to the other experiments, the controller was also able to reduce the error between the target and feedback states without any intervention and was also able to tolerate varying thicknesses in the drawn target shapes. Overall, this shows the generalizability of the controller due to the deep learning architecture. This also provides the user with an intuitive way to control the shape of the soft robot.

VII. DISCUSSION AND CONCLUSIONS

In this article, we introduce an image-based deep-learning kinematic controller for continuum robots. The controller is able to realise desired arbitrary target shapes given only image inputs with high accuracy and minimal errors. Thus, giving direct authority over not only the robot's tip but also its shape and configuration. We have also shown a simple yet elegant technique of teleoperating the soft continuum arm, where 2-D hand-drawn images are used to control the robot's 3-D configuration. This is much more user-friendly and intuitive compared to analytical methods which require plenty of parameters to define the arc and configuration for each of the robot's section [15], [16], [16], [17], [23], [26]. Hence, due to the generalizability of deep networks, the controller is able to realise an appropriate pose given a hand-drawn target without any user guidance or manual consideration of the robot's kinematics.

Our control system has demonstrated its versatility on a range of continuum robotic manipulators, regardless of their actuation method, materials, geometry, kinematics, and the number of degrees of freedom, as evidenced by our simulation and physical experiments with vastly different robots. This indicates that the system can be effectively applied to any type of such manipulator, as long as the network architecture and training data size are appropriate, and the target shape images are kinematically feasible. Our method is also applicable to robotic manipulators with lower degrees such as robotic fingers or manipulators, provided they are relatively easy to draw. Similarly, our controller is also able to operate regardless of the relative orientation between the robot and camera as shown by our results. Furthermore, our kinematic controller has a simple learning process, as demonstrated in our results. In our work, it took approximately 3 hours to generate samples, train and deploy the controller on the STIFF-FLOP manipulator. We verified the efficacy and robustness of the image-based controller in both simulations and in reality. In the simulation, it was seen that the deep CNN network is able to tolerate noise in the image feedback such that the desired shape is reached when trained with data augmentation techniques such as random translations, rotations and partial obstructions. More importantly, it was able to compensate for drastic changes in mechanical properties for up to 15% reduction in stiffness, whilst maintaining the accuracy of the resulting final shape to the desired target shape.

In the physical setup, the controller was shown to tolerate

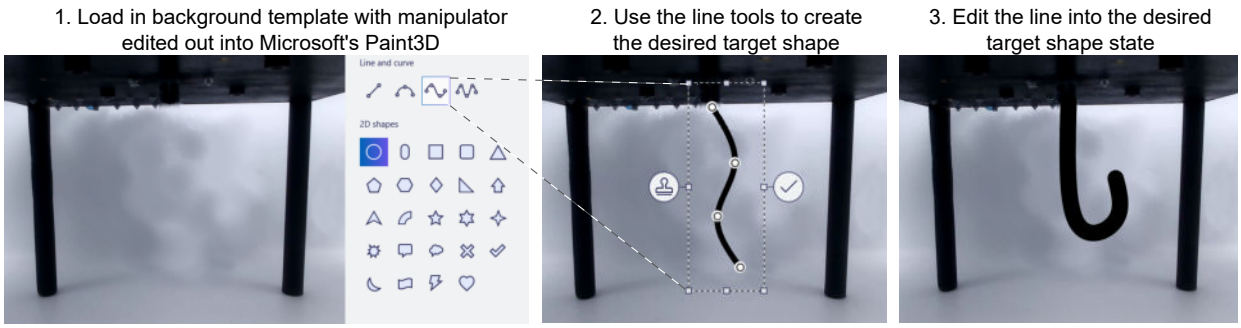


Fig. 17: Steps for the hand-drawn target shapes.

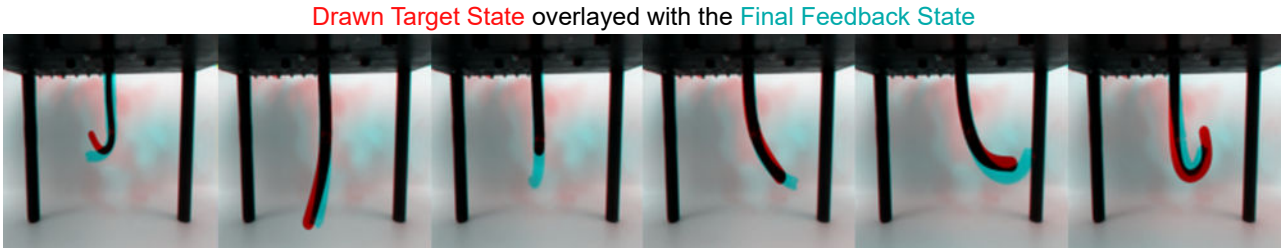


Fig. 18: Examples of the drawn target (in red) and final resulting states (in cyan) from the CNN controller using the STIFF-FLOP arm.

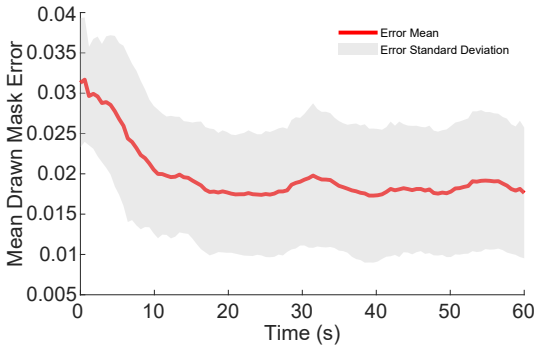


Fig. 19: Average mask error between the hand-drawn targets and the feedback states for the six target shapes given in Fig. 18 over the duration of 60 seconds. Mask errors were obtained by the image subtraction between $\mathbf{I_i}$ (feedback) and $\mathbf{I_{i+1}}$ (target) states after Otsu-thresholding, followed by normalisation and averaging of the absolute image errors. On average, the controller converges to the final shape after 10 seconds.

noise and variabilities such as inconsistent lighting conditions and translational noise in the image feedback inputs even without any training data augmentation. Based on our simulation experiments, however, we have shown that data augmentation can increase the performance of the controller under the presence of noise. Future work will hence conduct a more thorough analysis of the optimal level of data augmentation necessary to achieve robustness of physical systems to feedback noise. The controller also requires minimal sensors, needing only an inexpensive and off-the-shelf camera without requiring any calibration. These are major advantages over sensors used in current work such as optical markers for end-effector tracking which are expensive and require calibrations [52]. As the camera is an external form of sensing, it removes the need for additional fabrication steps, manual labour, and the challenges of integrating sensors, particularly in the case of soft robots. The same applies to the process of adding sensors to a pre-existing robot. Our controller also eliminates the requirement for additional steps and algorithms to process embedded sensing.

Model-based prior works have relied on simplifications and assumptions such as sectional constant curvature, pure symmetry in the structure, and minimal to no torsion in the exhibited actuated shapes. These assumptions severely limit the possible shape configurations and the scalability to more complicated continuum arms, hence affecting the accuracy of shape controllers based on such models [71]. Feedback control in these cases also involves an additional stage, where visual data has to be segmented and parametrized for comparison to the analytical model. The advantage of our technique is the whole use of image-state representations of both the task and the configuration space. This allows for high-dimensional and hyper-redundant descriptions of the arm which encodes all of these inherent yet implicit structural, mechanical and material properties of the arm, which is wholly controllable in 2D. Therefore, there is also no need to formulate specific mappings of the actuator space to the joint space that is typical of model-based controllers [15].

Counter-intuitively, the mapping between the shape configuration of the arm and the IK solutions is also not unique. This is greatly affected by the initial starting configuration, the material and structural properties, the noise in the image data, and even temporal dependence. For example, the same target shape is reachable from infinitely many directions. The ability to tolerate this redundancy is another major advantage of our local inverse kinematic formulation compared to other model-free global IK techniques such as goal babbling [51] which learn particular and specific solutions to the IK problem [49]. As shown in Fig. 15c, there are slight fluctuations in the trajectory leading to convergence for the same target shape, occurring between 10 to 30 seconds. This is due to two physical factors. The pneumatic tubes had slight leaks, which resulted in unexpected and random vibrations when the pressure was varied. Additionally, the STIFF-FLOP is a small silicone-based soft robotic system. Movement is a dynamic process due to the body's mass, combined with the imperfections in the pneumatic system, causing the robot to act like a 3D pendulum. Despite these fluctuations, our controller was able to bring the robot to the desired shape and hold it once it stabilized, which was consistent with our simulation results where the robot was able to handle changes in mechanical stiffness (refer to Fig. 8). A direct solution to this issue of oscillation would be to reduce the low pass filter gain and the control frequency, at the expense of speed to convergence.

Enhancements to the shape control accuracy, speed and robustness can be made to both the control system and the arm design. One way to improve is by gathering more training data through image augmentation and a diverse range of backgrounds and foregrounds typically used in computer vision. Additionally, using higher-resolution state images and optimizing the network architecture and hyper-parameters will increase the controller's resistance to noise and decrease steady-state errors. The use of well-established CNN architectures like ResNet [87], which have demonstrated effective performance in noisy environments, will also be taken into consideration. The control frequency is currently bottle-necked by the multiplexed DACs. The network takes 2 ms to process the image inputs, thus faster control frequencies can be achieved by replacing DACs with a better stand-alone Digital-Analog board. With faster control frequencies, our work can be extended to dynamic shape control by combining image state representations and convolutional networks with recurrent neural networks as was done in prior work for continuum soft robots [55]. The issue of non-unique projections will also be addressed using additional feature points as well as multiple camera views in future work. Data augmentation would not address the issue of full occlusions, which are common in medical scenarios such as minimally invasive surgery or endoscopy. A direct avenue of future research could be to leverage CT scans previously performed on medical continuum robots [88]. Alternatively, embedded sensors within the arm that permit 3D shape estimation can also be incorporated.

ACKNOWLEDGMENT

This work was supported by the SHERO project, a Future and Emerging Technologies (FET) programme of the European Commission (grant agreement ID 828818), AgriFoRwArdS Centre for Doctoral Training programme under the UKRI grant [EP/S023917/1], the Jersey Farmers Union, the Springboard Award of the Academy of Medical Sciences (grant number: SBF003-1109), the Engineering and Physical Sciences Research Council (grant numbers: EP/R037795/1, EP/S014039/1 and EP/V01062X/1), the Royal Academy of Engineering (grant number: IAPP18-19\264), the UCL Dean's Prize, UCL Mechanical Engineering, and the China Scholarship Council (CSC). For the purpose of open access, the author has applied a Creative Commons Attribution (CC BY) licence to any Author Accepted Manuscript version arising.

REFERENCES

[1] G. Robinson and J. B. C. Davies, "Continuum robots-a state of the art," in *Proceedings 1999 IEEE international conference on robotics and automation (Cat. No. 99CH36288C)*, vol. 4. IEEE, 1999, pp. 2849–2854.

[2] S. Kolachalama and S. Lakshmanan, "Continuum robots for manipulation applications: a survey," *Journal of Robotics*, vol. 2020, 2020.

[3] E. W. Hawkes, L. H. Blumenschein, J. D. Greer, and A. M. Okamura, "A soft robot that navigates its environment through growth," *Science Robotics*, vol. 2, no. 8, p. eaan3028, 2017.

[4] J. M. M. Tur and W. Garthwaite, "Robotic devices for water main in-pipe inspection: A survey," *Journal of Field Robotics*, vol. 4, no. 27, pp. 491–508, 2010.

[5] X. Dong, D. Axinte, D. Palmer, S. Cobos, M. Raffles, A. Rabani, and J. Kell, "Development of a slender continuum robotic system for on-wing inspection/repair of gas turbine engines," *Robotics and computer-integrated manufacturing*, vol. 44, pp. 218–229, 2017.

[6] J. Burgner-Kahrs, D. C. Rucker, and H. Choset, "Continuum robots for medical applications: A survey," *IEEE Transactions on Robotics*, vol. 31, no. 6, pp. 1261–1280, 2015.

[7] A. Arezzo, Y. Mintz, M. Allaix, S. Arolfo, M. Bonino, G. Gerboni, M. Brancadoro, M. Cianchetti, A. Menciassi, H. Wurdemann, Y. Noh, K. Althoefer, J. Fras, J. Glowka, Z. Nawrat, G. Cassidy, R. Walker, and M. Morino, "Total mesorectal excision using a soft and flexible robotic arm: a feasibility study in cadaver models," *Surgical Endoscopy*, vol. 31, no. 1, pp. 264–273, 2017.

[8] C. Li, X. Gu, X. Xiao, C. M. Lim, and H. Ren, "Flexible robot with variable stiffness in transoral surgery," *IEEE/ASME Transactions on Mechatronics*, vol. 25, no. 1, pp. 1–10, 2019.

[9] S.-A. Abad, A. Arezzo, S. Homer-Vanniasinkam, and H. A. Wurdemann, "Chapter 4 - soft robotic systems for endoscopic interventions," in *Endorobotics*, L. Manfredi, Ed. Academic Press, 2022, pp. 61–93. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128217504000049

[10] K.-W. Kwok, H. Wurdemann, A. Arezzo, A. Menciassi, and K. Althoefer, "Soft robot-assisted minimally invasive surgery and interventions: Advances and outlook," *Proceedings of the IEEE*, vol. 110, no. 7, pp. 871–892, 2022.

[11] A. Shiva, S. H. Sadati, Y. Noh, J. Fraś, A. Ataka, H. Würdemann, H. Hauser, I. D. Walker, T. Nanayakkara, and K. Althoefer, "Elasticity versus hyperelasticity considerations in quasistatic modeling of a soft finger-like robotic appendage for real-time position and force estimation," *Soft Robotics*, vol. 6, no. 2, pp. 228–249, 2019, pMID: 30702390. [Online]. Available: https://doi.org/10.1089/soro.2018.0060

[12] A. Shariati, J. Shi, S. Spurgeon, and H. A. Wurdemann, "Dynamic modelling and visco-elastic parameter identification of a fibre-reinforced soft fluidic elastomer manipulator," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 661–667.

[13] 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, pMID: 29297756. [Online]. Available: https://doi.org/10.1089/soro.2017.0007

[14] 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," *arXiv preprint arXiv:2110.01358*, 2021.

[15] I. Robert J. Webster and B. A. Jones, "Design and kinematic modeling of constant curvature continuum robots: A review," *The International Journal of Robotics Research*, vol. 29, no. 13, pp. 1661–1683, 2010.

[16] D. Camarillo, C. Carlson, and J. Salisbury, "Task-space control of continuum manipulators with coupled tendon drive," vol. 54, 01 2008, pp. 271–280.

[17] A. Kapadia and I. D. Walker, "Task-space control of extensible continuum manipulators," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, pp. 1087–1092.

[18] A. D. Kapadia, K. E. Fry, and I. D. Walker, "Empirical investigation of closed-loop control of extensible continuum manipulators," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 329–335.

[19] R. S. Penning, J. Jung, J. A. Borgstadt, N. J. Ferrier, and M. R. Zinn, "Towards closed loop control of a continuum robotic manipulator for medical applications," in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 4822–4827.

[20] R. S. Penning, J. Jung, N. J. Ferrier, and M. R. Zinn, "An evaluation of closed-loop control options for continuum manipulators," in *2012 IEEE International Conference on Robotics and Automation*, 2012, pp. 5392–5397.

[21] M. W. Hannan and I. D. Walker, "Kinematics and the implementation of an elephant's trunk manipulator and other continuum style robots," *Journal of Robotic Systems*, vol. 20, no. 2, pp. 45–63, 2003. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.10070

[22] M. Li, R. Kang, S. Geng, and E. Guglielmino, "Design and control of a tendon-driven continuum robot," *Transactions of the Institute of Measurement and Control*, vol. 40, no. 11, pp. 3263–3272, 2018. [Online]. Available: https://doi.org/10.1177/0142331216685607

[23] T. Mahl, A. E. Mayer, A. Hildebrandt, and O. Sawodny, "A variable curvature modeling approach for kinematic control of continuum manipulators," in *2013 American Control Conference*, 2013, pp. 4945–4950.

[24] T. Mahl, A. Hildebrandt, and O. Sawodny, "A variable curvature continuum kinematics for kinematic control of the bionic handling assistant," *IEEE Transactions on Robotics*, vol. 30, no. 4, pp. 935–949, 2014.

[25] T. Zheng, D. T. Branson, R. Kang, M. Cianchetti, E. Guglielmino, M. Follador, G. A. Medrano-Cerda, I. S. Godage, and D. G. Caldwell, "Dynamic continuum arm model for use with underwater robotic manipulators inspired by octopus vulgaris," in *2012 IEEE International Conference on Robotics and Automation*, 2012, pp. 5289–5294.

[26] 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 &amp Biomimetics*, vol. 7, no. 2, p. 025006, may 2012. [Online]. Available: https://doi.org/10.1088/1748-3182/7/2/025006

[27] 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.

[28] D. B. Camarillo, C. R. Carlson, and J. K. Salisbury, "Configuration tracking for continuum manipulators with coupled tendon drive," *IEEE Transactions on Robotics*, vol. 25, no. 4, pp. 798–808, 2009.

[29] C. Duriez, "Control of elastic soft robots based on real-time finite element method," in *2013 IEEE International Conference on Robotics and Automation*, 2013, pp. 3982–3987.

[30] O. Goury and C. Duriez, "Fast, generic, and reliable control and simulation of soft robots using model order reduction," *IEEE Transactions on Robotics*, vol. 34, no. 6, pp. 1565–1576, 2018.

[31] T. M. Bieze, F. Largilliere, A. Kruszewski, Z. Zhang, R. Merzouki, and C. Duriez, "Finite element method-based kinematics and closed-loop control of soft, continuum manipulators," *Soft Robotics*, vol. 5, no. 3, pp. 348–364, 2018, pMID: 29658827. [Online]. Available: https://doi.org/10.1089/soro.2017.0079

[32] Y. Bailly and Y. Amirat, "Modeling and control of a hybrid continuum active catheter for aortic aneurysm treatment," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, 2005, pp. 924–929.

[33] A. D. Marchese, K. Komorowski, C. D. Onal, and D. Rus, "Design and control of a soft and continuously deformable 2d robotic manipulation system," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 2189–2196.

[34] C. Della Santina, R. K. Katzschmann, A. Biechi, and D. Rus, "Dynamic control of soft robots interacting with the environment," in *2018 IEEE International Conference on Soft Robotics (RoboSoft)*. IEEE, 2018, pp. 46–53.

[35] V. Falkenhahn, A. Hildebrandt, and O. Sawodny, "Trajectory optimization of pneumatically actuated, redundant continuum manipulators," in *2014 American Control Conference*, 2014, pp. 4008–4013.

[36] V. Falkenhahn2, A. Hildebrandt, R. Neumann, and O. Sawodny, "Model-based feedforward position control of constant curvature continuum robots using feedback linearization," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 762–767.

[37] A. A. Alqumsan, S. Khoo, and M. Norton, "Robust control of continuum robots using cosserat rod theory," *Mechanism and Machine Theory*, vol. 131, pp. 48–61, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0094114X18311777

[38] F. A. Spinelli and R. K. Katzschmann, "A unified and modular model predictive control framework for soft continuum manipulators under internal and external constraints," 2022. [Online]. Available: https://arxiv.org/abs/2204.13710

[39] M. Wang, X. Dong, W. Ba, A. Mohammad, D. Axinte, and A. Norton, "Design, modelling and validation of a novel extra slender continuum robot for in-situ inspection and repair in aeroengine," *Robotics and Computer-Integrated Manufacturing*, vol. 67, p. 102054, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0736584520302659

[40] D. Braganza, D. M. Dawson, I. D. Walker, and N. Nath, "A neural network controller for continuum robots," *IEEE Transactions on Robotics*, vol. 23, no. 6, pp. 1270–1277, 2007.

[41] J. F. Queißer, K. Neumann, M. Rolf, R. F. Reinhart, and J. J. Steil, "An active compliant control mode for interaction with a pneumatic soft robot," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 573–579.

[42] Z. Wang, T. Wang, B. Zhao, Y. He, Y. Hu, B. Li, P. Zhang, and M. Q.-H. Meng, "Hybrid adaptive control strategy for continuum surgical robot under external load," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1407–1414, 2021.

[43] D. Bruder, X. Fu, R. B. Gillespie, C. D. Remy, and R. Vasudevan, "Koopman-based control of a soft continuum manipulator under variable loading conditions," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 6852–6859, 2021.

[44] Z. Q. Tang, H. L. Heung, K. Y. Tong, and Z. Li, "A novel iterative learning model predictive control method for soft bending actuators," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 4004–4010.

[45] D. Hahn, P. Banzet, J. M. Bern, and S. Coros, "Real2sim: Visco-elastic parameter estimation from dynamic motion," *ACM Trans. Graph.*, vol. 38, no. 6, nov 2019. [Online]. Available: https://doi.org/10.1145/3355089.3356548

[46] M. C. Yip and D. B. Camarillo, "Model-less feedback control of continuum manipulators in constrained environments," *IEEE Transactions on Robotics*, vol. 30, no. 4, pp. 880–889, 2014.

[47] F. Alambeigi, Z. Wang, Y. Liu, and R. Taylor, "A versatile data-driven framework for model-independent control of continuum manipulators interacting with obstructed environments with unknown geometry and stiffness," 05 2020.

[48] W. Ba, X. Dong, A. Mohammad, M. Wang, D. Axinte, and A. Norton, "Design and validation of a novel fuzzy-logic-based static feedback controller for tendon-driven continuum robots," *IEEE/ASME Transactions on Mechatronics*, vol. 26, no. 6, pp. 3010–3021, 2021.

[49] M. Giorelli, F. Renda, G. Ferri, and C. Laschi, "A feed-forward neural network learning the inverse kinetics of a soft cable-driven manipulator moving in three-dimensional space," *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5033–5039, 2013.

[50] M. Giorelli, F. Renda, M. Calisti, A. Arienti, G. Ferri, and C. Laschi, "Learning the inverse kinetics of an octopus-like manipulator in three-dimensional space," *Bioinspiration & Biomimetics*, vol. 10, no. 3, p. 035006, may 2015. [Online]. Available: https://doi.org/10.1088%2F1748-3190%2F10%2F3%2F035006

[51] M. Rolf and J. Steil, "Goal babbling: a new concept for early sensorimotor exploration," 11 2012.

[52] T. George Thuruthel, E. Falotico, M. Manti, A. Pratesi, M. Cianchetti, and C. Laschi, "Learning closed loop kinematic controllers for continuum manipulators in unstructured environments," *Soft robotics*, vol. 4, no. 3, pp. 285–296, 2017.

[53] A. R. Sahoo and P. Chakraborty, "A study on position control of a continuum arm using maml (model-agnostic meta-learning) for adapting different loading conditions," *IEEE Access*, vol. 10, pp. 14 980–14 992, 2022.

[54] 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.

[55] 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.

[56] K. Chin, T. Hellebrekers, and C. Majidi, "Machine learning for soft robotic sensing and control," *Advanced Intelligent Systems*, vol. 2, no. 6, p. 1900171, 2020.

[57] M. T. Gillespie, C. M. Best, E. C. Townsend, D. Wingate, and M. D. Killpack, "Learning nonlinear dynamic models of soft robots for model predictive control with neural networks," in *2018 IEEE International Conference on Soft Robotics (RoboSoft)*. IEEE, 2018, pp. 39–45.

[58] R. Morimoto, S. Nishikawa, R. Niiyama, and Y. Kuniyoshi, "Model-free reinforcement learning with ensemble for a soft continuum robot arm," in *2021 IEEE 4th International Conference on Soft Robotics (RoboSoft)*, 2021, pp. 141–148.

[59] T. George Thuruthel, E. Falotico, F. Renda, and C. Laschi, "Model-based reinforcement learning for closed-loop dynamic control of soft robotic manipulators," *IEEE Transactions on Robotics*, vol. PP, pp. 1–11, 11 2018.

[60] S. Satheeshbabu, N. K. Uppalapati, G. Chowdhary, and G. Krishnan, "Open loop position control of soft continuum arm using deep reinforcement learning," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 5133–5139.

[61] T. da Veiga, J. H. Chandler, P. Lloyd, G. Pittiglio, N. J. Wilkinson, A. K. Hoshiar, R. A. Harris, and P. Valdastri, "Challenges of continuum robots in clinical context: a review," *Progress in Biomedical Engineering*, vol. 2, no. 3, p. 032003, aug 2020. [Online]. Available: https://doi.org/10.1088/2516-1091/ab9f41

[62] P. E. Dupont, N. Simaan, H. Choset, and C. Rucker, "Continuum robots for medical interventions," *Proceedings of the IEEE*, vol. 110, no. 7, pp. 847–870, 2022.

[63] A. Bajo, R. E. Goldman, and N. Simaan, "Configuration and joint feedback for enhanced performance of multi-segment continuum robots," in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 2905–2912.

[64] M. Neumann and J. Burgner-Kahrs, "Considerations for follow-the-leader motion of extensible tendon-driven continuum robots," 05 2016, pp. 917–923.

[65] E. Amanov, T.-D. Nguyen, and J. Burgner-Kahrs, "Tendon-driven continuum robots with extensible sections—a model-based evaluation of path-following motions," *The International Journal of Robotics Research*, vol. 40, no. 1, pp. 7–23, 2021.

[66] J. Santoso and C. D. Onal, "An origami continuum robot capable of precise motion through torsionally stiff body and smooth inverse kinematics," *Soft Robotics*, vol. 8, no. 4, pp. 371–386, 2021, pMID: 32721270. [Online]. Available: https://doi.org/10.1089/soro.2020.0026

[67] J. M. Bern, G. Kumagai, and S. Coros, "Fabrication, modeling, and control of plush robots," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 3739–3746.

[68] J. M. Bern, P. Banzet, R. Poranne, and S. Coros, "Trajectory optimization for cable-driven soft robot locomotion," *Robotics: Science and Systems XV*, 2019.

[69] G. Pittiglio, P. Lloyd, T. da Veiga, O. Onaizah, C. Pompili, J. H. Chandler, and P. Valdastri, "Patient-specific magnetic catheters for atraumatic autonomous endoscopy," *Soft Robotics*, vol. 9, no. 6, pp. 1120–1133, 2022, pMID: 35312350. [Online]. Available: https://doi.org/10.1089/soro.2021.0090

[70] J. Davy, P. Lloyd, J. H. Chandler, and P. Valdastri, "A framework for simulation of magnetic soft robots using the material point method," *IEEE Robotics and Automation Letters*, vol. 8, no. 6, pp. 3470–3477, 2023.

[71] B. Ouyang, Y. Liu, H.-Y. Tam, and D. Sun, "Design of an interactive control system for a multisection continuum robot," *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 5, pp. 2379–2389, 2018.

[72] 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.

[73] A. D'Souza, S. Vijayakumar, and S. Schaal, "Learning inverse kinematics," in *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No. 01CH37180)*, vol. 1. IEEE, 2001, pp. 298–303.

[74] D. E. Whitney, "Resolved motion rate control of manipulators and human prostheses," *IEEE Transactions on man-machine systems*, vol. 10, no. 2, pp. 47–53, 1969.

[75] J. Burgner-Kahrs, D. Rucker, and H. Choset, "Continuum robots for medical applications: A survey," *IEEE Transactions on Robotics*, vol. 31, pp. 1–20, 11 2015.

[76] A. Yeshmukhametov, K. Koganezawa, Y. Yamamoto, Z. Buribayev, Z. Mukhtar, and Y. Amirgaliyev, "Development of continuum robot arm and gripper for harvesting cherry tomatoes," *Applied Sciences*, vol. 12, no. 14, 2022. [Online]. Available: https://www.mdpi.com/2076-3417/12/14/6922

[77] E. W. Hawkes, L. H. Blumenschein, J. D. Greer, and A. M. Okamura, "A soft robot that navigates its environment through growth," *Science Robotics*, vol. 2, no. 8, p. eaan3028, 2017. [Online]. Available: https://www.science.org/doi/abs/10.1126/scirobotics.aan3028

[78] ——, "A soft robot that navigates its environment through growth," *Science Robotics*, vol. 2, no. 8, p. eaan3028, 2017. [Online]. Available: https://www.science.org/doi/abs/10.1126/scirobotics.aan3028

[79] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[80] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[81] K. Fukushima and S. Miyake, "Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition," in *Competition and cooperation in neural nets*. Springer, 1982, pp. 267–285.

[82] J. B. Pascual-Francisco, L. I. Farfan-Cabrera, and O. Susarrey-Huerta, "Characterization of tension set behavior of a silicone rubber at different loads and temperatures via digital image correlation," *Polymer Testing*, vol. 81, p. 106226, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0142941819316344

[83] J. Fraś, J. Czarnowski, M. Maciaś, J. Główka, M. Cianchetti, and A. Menciassi, "New STIFF-FLOP module construction idea for improved actuation and sensing," in *IEEE International Conference on Robotics and Automation*, 2015, pp. 2901–2906.

[84] H. Abidi, G. Gerboni, M. Brancadoro, J. Fras, A. Diodato, M. Cianchetti, H. Wurdemann, K. Althoefer, and A. Menciassi, "Highly dexterous 2-module soft robot for intra-organ navigation in minimally invasive surgery," *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 14, no. 1, p. e1875, 2018.

[85] J. Shi, W. Gaozhang, and H. A. Wurdemann, "Design and characterisation of cross-sectional geometries for soft robotic manipulators with fibre-reinforced chambers," in *2022 IEEE 5th International Conference on Soft Robotics (RoboSoft)*, 2022, pp. 125–131.

[86] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[87] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015. [Online]. Available: http://arxiv.org/abs/1512.03385

[88] E. J. Lobaton, J. Fu, L. G. Torres, and R. Alterovitz, "Continuous shape estimation of continuum robots using x-ray images," in *2013 IEEE International Conference on Robotics and Automation*, 2013, pp. 725–732.