Grow-to-Shape Control of Variable Length Continuum Robots via Adaptive Visual Servoing

Abhinav Gandhi [1], Shou-Shan Chiang [1], Cagdas D. Onal [1], and Berk Calli[1]

***Abstract*— In this paper, we propose an adaptive eye-to-hand vision-based control methodology, which enables a closed-loop grow-to-shape capability for variable length continuum manipulators in 2D. Our method utilizes shape features of the continuum robot, i.e. module curvature and length, which are obtained from the image. Our adaptive control algorithm servos the robot to converge and track the desired values of these features in the image space without the need of a robot model. As a result the robot starts from a minimum length configuration and grows into a given desired shape, always staying on the course of the desired shape. We believe that this approach unlocks capabilities for variable length continuum robots by leveraging their actuation redundancy and avoiding obstacles while carrying out object manipulation or inspection tasks in cluttered and constrained environments. We perform experiments in simulations and on a real robot to assess the performance of our visual servoing algorithm. Our experiments demonstrate the controllers ability to accurately converge the current features to their references, for a variety of desired shapes in the image, while ensuring a smooth tracking response. We also present some proof of concept results demonstrating the effectiveness of this technique for controlling the robot in constrained environments. Markedly, this is the first successful demonstration for automatic grow-to-shape control using visual feedback for variable length continuum manipulators.**

I. Introduction

Continuum robots represent a paradigm shift in robotics research as well as its applications. Unlike traditional rigid-link robots that are composed of discrete joints, continuum manipulators are characterized by continuous flexible structures that mimic biological systems such as elephant trunks and octopus tentacles. This design approach allows them to conform to a wider range of shapes and configurations when compared to their rigid counterparts. Additionally, soft continuum manipulators offer inherent safety benefits and have reduced risks of injury or damage when interacting with humans or delicate objects compared to traditional rigid link manipulators. These advantages make continuum manipulators the perfect choice for applications in medical and assistive robotics [1]–[3], search and rescue operations [4], and safety inspections in constrained spaces [5]. Use of extensible robots in these applications can be advantageous as they can grow around obstacles to reach targets from various approach orientations. Accurate control of the entire shape of the manipulator's body is beneficial in such conditions. Modeling inaccuracies [6], [7] pose a major challenge in precisely controlling continuum manipulators but vision-based control techniques have been shown to be robust to common modeling inaccuracies in continuum robots as shown in our prior work [8] and in [9], [10].

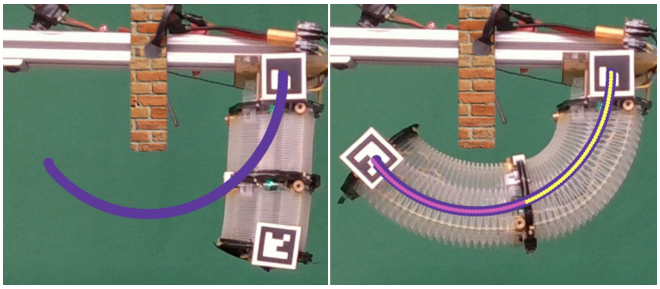


Fig. 1. **Left:** Shows the two module origami continuum arm at its minimum length configuration and the desired shape reference in the image, shown by the purple curve. **Right:** Shows the origami arm fully extended to match the desired shape reference in the image space. The two segments of the Piecewise Constant Curvature arcs estimating the shape of the robot are shown in yellow and pink. Notice how the manipulator bends around an obstacle to reach the target shape.

In this work, we present an adaptive vision-based control scheme that allows a variable length continuum manipulator grow to a desired shape, from its fully shrunk state to a goal configuration. While doing so, the robot closely follows the desired shape curve in the image space with minimal deviations, enabling to avoid obstacles while achiving the desired configuration. We believe that the ability to automatically grow the continuum arm, using purely visual features, while asserting control over its whole body configuration is an important skill that will unlock the ability to address challenges in object manipulation in cluttered and constrained environments. Fig. 1 shows an origami inspired continuum manipulator in its fully shrunk state and its desired final shape depicted by the purple curve. The robot grows along this desired shape and around an obstacle in the environment to reach a target. The estimated shape of the robot is represented by Piecewise Constant Curvature (PCC) arcs (shown in yellow and pink) that parameterize the shape of each module of the robot. Shape features of individual segments of the PCC arc are extracted and used to define an image space error vector between the current and desired shape of the robot. The robot's desired shape in the image is represented by a time varying PCC arc that grows in length until it achieves the final shape reference in the image. An adaptive visual servoing algorithm is developed to track the shape using the shape features. To the best of our knowledge, this paper presents the first successful demonstration for

*This paper was supported in part by the National Science Foundation under grant IIS-1900953, CMMI-1752195 and DGE-1922761.

[1]Abhinav Gandhi, Shou-Shan Chiang, Cadas D. Onal, and Berk Calli are with Department of Robotics Engineering, Worcester Polytechnic Institute, Worcester, MA 01609, USA agandhi2@wpi.edu

automatic grow-to-shape control using visual feedback for variable length continuum manipulators.

The closest work in literature that controls the shape of a variable length continuum manipulator is our prior work [8]. This work focuses on controlling the whole body shape of the robot from it's current shape to a desired shape, but it is unable to constrain the growing motion of the arm along the desired shape curve. Another recent work proposed in [11] uses a model-based image-based visual servoing method to control a soft robot that grows via tip extension. This formulation only focuses on controlling the end effector location of the robot using an eye-in-hand camera, but does not control the full body shape. In this case, motion of the robot end effector can change the shape of the robot, which may cause unintentional environmental interactions.

Scope and Assumptions of This Paper:

In this work, we focus on a planar implementation of grow to shape control for variable length continuum manipulators with adaptive Image-Based Visual Servoing (IBVS) in the eye-to-hand configuration. We use two modules of a novel variable length origami inspired continuum manipulator [12] to test the performance of our algorithm. The two modules of the robot grow along their reference shapes in the image one after the other, ie: only one module is allowed to grow at a time. The out of plane bending motion of the origami modules is constrained and we independently control four tendons of the manipulator, two on each module. This gives us four Degrees of Freedom (DoF) in two dimensions and thus configuration redundancy for a given end effector pose of the arm. The low level controllers of the origami arm use a coarsely tuned proportional control. ArUco markers are pasted at the base and the end effector of the robot to estimate their respective poses in the image. However, we do not use markers to specifically track shape features of the robot, instead fit PCC arcs to the robot's shape in the image. Further, we assume that the individual modules of the robot hold the PCC shape, which provides a close approximation under no-load conditions.

II. Background

This section discusses the origami inspired continuum arm used in our experiments, recaps the relevant constant curvature arc fitting models for shape approximation, and discusses related work in vision-based control of continuum robots.

A. Origami Continuum Arm

In this paper we use a modular origami-inspired cable-driven continuum manipulator. Each module of the origami continuum arm is capable of bending in 3-D and expanding in length [13]. The minimum length of an individual module is 80 mm and it can expand up to 200 mm in length. The modules are light weight and constructed using PET sheets that are folded in the Yoshimura crease pattern. Each module is driven by three tendons that are controlled individually by inexpensive actuators mounted at the base. This tendon driven actuation method, having actuators at the robot base, and not having proprioceptive sensing on the robots body are some common properties within many continuum arm designs, and the algorithms designed in this paper, being model-free, can easily be applied to other continuum manipulators with minor changes. In this work, we control two series connected origami modules as shown in Fig. 6.

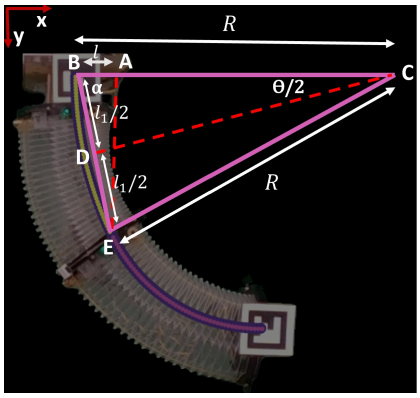


Fig. 2. Geometric method for fitting a constant curvature arc (shown in yellow) given the points **B** and **E** when the tangent to the arc at **B** is perpendicular to the x-axis (shown in red) of the image frame.

B. Shape Modeling of Continuum Arms

Continuum arms similar to the one we use in this paper are often modeled as constant curvature arcs. To model the entire shape of a multiple section continuum arm, Piecewise Constant Curvature (PCC) curves can be utilized. In this work, we model the shape of each module of the continuum arm as a constant curvature arc section of a PCC curve. The features of each arc section, arc length and curvature, are utilized as shape features in our image-based servo loop as discussed in Section III-C. We will now recap a geometric technique to fit constant curvature arcs to a module of the origami continuum robot. Let us consider a simplified case as shown in Fig. 2. The base of the arc at $\mathbf{B}(x_b, y_b)$ has a tangent that is perpendicular to the x-axis of the image frame. In this case, the center of the arc lies at $\mathbf{C}(x_c, y_c)$ where $x_c = x_b + R$, $y_c = y_b$. R is the radius of the curve as shown in Fig. 2. The straight line $\mathbf{BE}$ connecting the base and end $\mathbf{E}(x_e, y_e)$ of the arc has a length $l_1 = \sqrt{(x_b - x_e)^2 + (y_b - y_e)^2}$. Since the $\Delta\mathbf{BCE}$ is isosceles, the $\perp$ $\mathbf{CD}$ divides $\mathbf{BE}$ into halves. The length l can be computed by projecting $\mathbf{E}$ on $\mathbf{BC}$. We arrive at eq. 1 using $\Delta\mathbf{BAE}$ and $\Delta\mathbf{BCD}$. Eq. 2 is easily deduced from $\Delta\mathbf{BCE}$.

$$R = l_1^2/2l \tag{1}$$

$$\theta = 2(\pi/2 - arccos(l/l1)) \tag{2}$$

For the case, when the tangent to the base of the arc is not perpendicular to the x-axis of the image frame, a simple rotation can be applied to compute the values of the radius of the curve R and the central angle θ of the arc section.

C. Vision-based Control

Traditional Image-based Visual Servoing (IBVS) schemes [14], [15] utilize visual features that are observed either on the robot (eye-to-hand applications) or in the robot's environment (eye-in-hand applications) to control the pose of the manipulator's end effector. These techniques require knowledge of the robot Jacobian, camera calibration parameters, as well as sensor information such as joint encoder readings. In contrast to traditional IBVS, adaptive visual servoing methods require minimal knowledge of system parameters. They estimate system parameters online by utilizing a short series of excitation signals for system identification [16]–[18]. Since these methods are robust to modeling inaccuracies, they have also been applied to control the end effector pose of soft and continuum robots [10], [19], [20]. In [11] a model-based eye-in-hand IBVS approach is designed to control the tip extension of a continuum robot with pneumatic segments. Although this work allows to steer the robot around obstacles in the environment, and grow towards a target in the image, there is no explicit control over the shape of the robot's body. In this work, we present an eye-to-hand application of visual servoing, where an external camera observes and controls the shape of the robot. Adaptive IBVS formulations have been known to be useful when controlling systems with unknown or uncertain system models. In [21] an adaptive IBVS controller is used to servo a variable length continuum robot in a constrained environment. Constraints in the environment are used as passive contacts that apply bending forces on different regions of the continuum manipulator to conform it to the shape of the external environment but there is no explicit control over the shape of the robot. In contrast to controlling only the end effector pose of the robot in the image, we control the entire shape of the continuum manipulator in the image.

In [22] an adaptive visual servoing scheme is used to control the shape of a soft robotic manipulator. However, this work is formulated for in-extensible soft robots. Most recently, our work on whole body shape control of continuum robots [8] utilizes an adaptive scheme with an initial exploration phase to control a variable length continuum manipulator from one shape to another in the image. However, this work does not constrain the transient response of the system to grow along the desired shape curve in the image. In this work, we present an adaptive vision-based shape control algorithm that grows the variable length continuum manipulator along a desired shape curve in the image. To the best of our knowledge, these results on automatic shape growing of variable length continuum robots using visual feedback are the first of their kind in the literature.

III. Grow to Shape Adaptive Visual Servoing

In this paper, we grow a modular origami inspired variable length continuum manipulator [13] from its minimum length state to a desired shape while asserting control over the entire configuration of the robot in the image. Our work enables the robot to expand in length, one module at a time, in constrained or cluttered spaces without much deviation between its whole body shape and the desired shape curve in the image. In the rest of this paper, the term active module refers to the module of the robot that is growing to shape. We believe this work develops important shape-control techniques, for variable length continuum manipulators, that are required for carrying out object manipulation and navigation actions in cluttered and constrained environments.

This section discusses our approach to estimate the current shape of the active module in the image, shape feature extraction, formulation of the shape tracking problem in image space, and how we generate a dense set of valid intermediate growing shape references in the image without requiring snapshots of the robot at the reference shapes.

A. Shape Estimation & Representation

In order to accurately control the shape of the robot in the image, we first need to estimate the active module's current shape in the image. We use Constant Curvature arcs as our parametric shape representation of the active module of our two module origami arm. It is well known that constant curvature arcs provide an accurate representation of continuum robot segments under no-load conditions [23]. To estimate the current shape of the active module, we need to know its base pose and tip position in the image as shown in Section II-B. As mentioned earlier, we assume that the base and end effector pose of the robot can be tracked in the image. However, the tip pose for the first module is unknown. This unknown tip pose is computed in the image by estimating the offset distance between the tip and the robot's end effector by manually selecting the tip in the image while the robot is in its initial (fully shrunk) state. Since this offset distance remains constant while the first module is growing, we only need to estimate it once. The tip position can be computed in real time using the values of the estimated offset distance and end effector pose. With the base pose and tip position of the active module its current shape in the image can be estimated, by fitting a constant curvature arc using the geometric methods recapped in Section II-B.

B. Shape Feature Extraction

In our vision-based control implementation we use constant curvature arc parameters, arc length denoted by s and curvature denoted by κ, as our shape features to servo the robot shape in the image. These are determined by using geometric equations of arcs: $s = R\theta$ and $\kappa = 1/R$, where R is the radius of curvature and θ is the center angle of the arc. The extracted shape features of the active module are arranged as follows in our image feature vector

$$\mathbf{f}_i = [s_i, \kappa_i]$$

where the subscript $i \in I$ and denotes the module number of the active module.

C. Image-Based Visual Servoing

The image space error $\mathbf{e}_i$, in eq. (3), is the difference between the current and desired shapes of the active module in the image. It is formulated as the difference between the

current shape features $\mathbf{f}_i$ and the desired shape features $\mathbf{f}_i^*$. The current shape features consist of arc parameters of the active module as explained above while the desired shape features consist of the arc parameters for the growing shape reference.

$$\mathbf{e}_i = \mathbf{f}_i - \mathbf{f}_i^* \tag{3}$$

The image space error is utilized in the image-based control loop [24] as shown in eq (4) to ensure that the current shape features track the desires shape features of the growing curve in the image.

$$\mathbf{v}_i = -\lambda \hat{J}_c^+ \mathbf{e}_i + \hat{J}_c^+ \dot{\mathbf{f}}_i^* \tag{4}$$

The combined Jacobian $\hat{J}_c$ is a product of the robot Jacobian J_r, that relates actuator velocities to arc parameter velocities in cartesian space, and an image Jacobian J_i that relates the cartesian space velocities of the arc parameters to image feature velocities. The combined Jacobian is first initialized with an initial guess and then updated online using the modified projection algorithm as described in section III-D.

D. Jacobian Adaptation & Reset

The initial value of the combined Jacobian $\hat{J}_c^+$ is updated at each iteration of the control loop by observing the system behavior for each set of velocity inputs to the actuators of the active module. An update term $\Delta \hat{J}_c^+(n)$ is computed by projecting and scaling the difference between the measured change in image features and the expected change in image features (based on the previous estimate of the combined Jacobian and the input velocities to the actuators) as shown in eq (5). Here, η is a scaling factor and is in the range $[0, 1]$ and $\mu > 0$ is a weighting factor. It is crucial to appropriately select values of μ and η to ensure smooth actuator inputs to the system.

$$\Delta \hat{J}_c^+(n) = \eta \frac{(\dot{\mathbf{f}}_i(n+1) - \hat{J}_c^+(n)\mathbf{v}_i(n))\mathbf{v}_i(n)^T}{\mu + \|\mathbf{v}_i(n)\|^2} \tag{5}$$

The update to each element $\hat{J}_{c_j}^+$ of the combined Jacobian matrix is performed, at each iteration of the control loop, as shown in eq (6). This method is commonly known as the modified projection algorithm [20].

$$\hat{J}_{c_j}^+(n+1) = \Delta \hat{J}_{c_j}^+(n) + \hat{J}_{c_j}^+(n) \tag{6}$$

To ensure a smooth transition between the active modules, we implement a Jacobian reset functionality. At the time instance where the active module switches to the next module, the estimated combined Jacobian is reset to its initial guess. This reset functionality ensures that the estimated combined Jacobian does not over-fit to the physical properties of either module as the properties for each module may vary due to wear and tear.

E. Reference Selection & Growing Reference Shapes

Image-based visual servo schemes compute an error between the current features and the reference features for the system to converge to its desired pose in the image. Conventionally, the current features are obtained from the current image of the manipulator that is acquired from an external camera looking at the robot while the desired features are obtained from a snapshot of the robot at its reference pose. In practical settings, it is not always possible to obtain an image of the robot in its desired configuration. This is an inherent limitation of image-based visual servoing formulations. *Our approach for shape reference feature generation does not require any snapshots of the robot in its desired configuration*. Instead, we generate reference shape features in the current image of the robot by selecting a reference end effector pose for the robot in the image. This is done by clicking a point in the current image to select the reference position and by dragging the mouse pointer to select the reference orientation. Next, a shape reference for the selected end effector pose is generated in the image by using an inverse kinematics (IK) solver (AMoRPH) [25], [26] that utilizes an analytical geometry approach [27]. This generated shape is the desired shape of the robot in the image. Reference shape features s and κ are computed as shown in Section III-B. Growing references are provided to the image-based visual servo loop by initializing the central angle of the PCC arc to a minimum value $\theta = \theta_{min}$. At each iteration of the control loop, θ is increased at a fixed rate until the reference arc achieves its final length in the image. The rate of increase of θ determines how rapidly the reference and thus the robot grows in the image. Note that certain curves in the workspace might not be tractable due to the robot's constraints i.e. maximum length and bending angle. In order to ensure the generated shapes are tractable, the hardware constraints need to be integrated to the AMoRPH solver.

In addition to generating kinematically feasible references, the IK solver discussed in [25] is also capable of generating collision free references. Grow to shape capability can be utilized for various applications such as reaching a target object in a cluttered environment and visual inspection in confined spaces. In such cases, our control algorithm would rely on a separate method, such as [25], to generate collision free reference shapes.

IV. Simulation

We test the grow to shape performance of our algorithm on a numerical simulation of two modules of the continuum arm. In this section, we present the simulation setup and discuss the results obtained.

A. Simulation Setup

A numerical simulation is developed using Python3 using the constant curvature model and virtual joints as described in [27]. The continuum modules in the simulation are developed to be similar to the origami arm discussed in [13]. A simulation camera is spawned using the pinhole projection model and a visualization of the camera image is created using OpenCV. A ROS package is implemented to integrate all the components of the simulation. We provide the package as open source here [28]. The robot kinematics implemented in the simulation are generalized and capable of spawning multiple modules connected in series. In our work we focus on controlling two modules of the robot in the image.

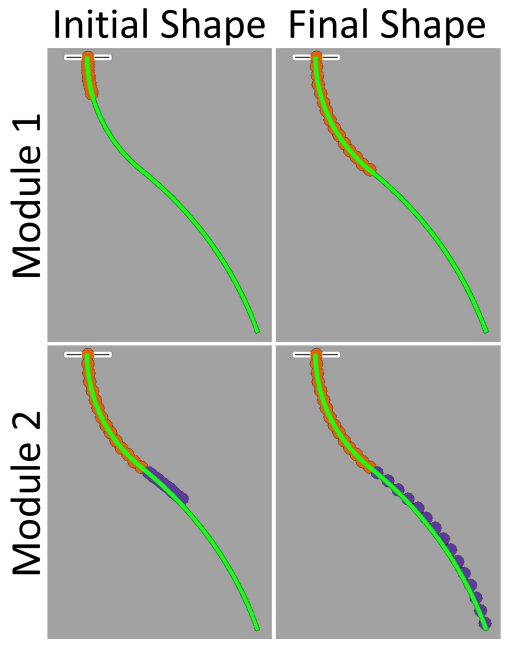


Fig. 3. Simulation experiment showing a two module variable length continuum arm expanding to the final reference shape curve shown in green. Module 1 of the continuum robot is shown in orange and module 2 is shown in purple.

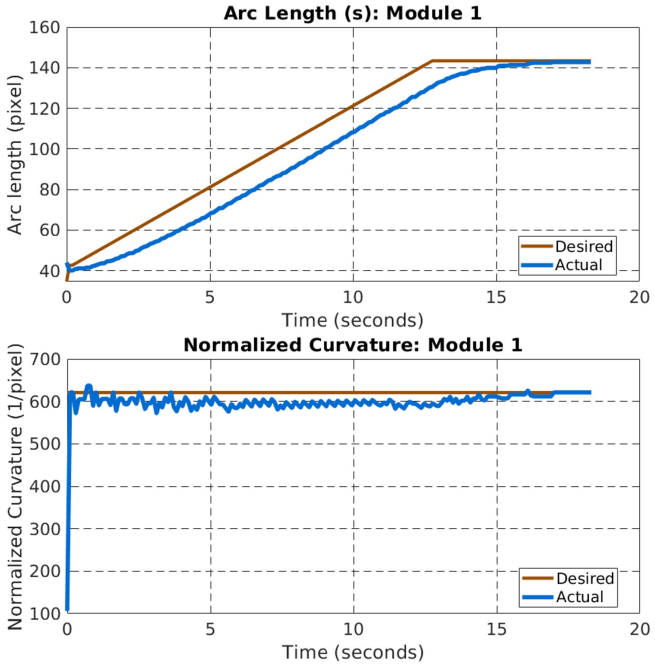


Fig. 4. Shape feature convergence plot for module 1 for the simulation experiment demonstrated in Fig. 3.

The motion of the modules is constrained to a plane and we control 2 degrees of freedom(DoF) in each module. Altogether, our simulation setup has 4 DoF in a plane and is a redundant system. In Fig. 3, the images obtained from the simulation camera are visualized. The different colored segments (orange and purple) denote the different modules of the continuum arm that are connected in series and suspended from a horizontal base shown in white.

B. Simulation Results: Grow to Shape

The proposed adaptive visual servoing algorithm is tested on various growing shape references in the image using our numerical simulation. Snapshots for a trial where the robot automatically expands to an "S-shape" reference on the right side of its workspace are provided in Fig. 3. The first row of the figure demonstrates module 1 (shown in orange) expanding to its reference shape (shown in blue). In the second row, module 2 (shown in purple) expands from its minimum length to the desired shape (shown in blue). The feature convergence plots for module 1 and module 2 is shown in Fig. 4 and Fig. 5 respectively. A small-steady state tracking error is observed for both the features for both modules. However, all the features converge to their respective reference values with minimal overshoot.

V. Experimental Results

A. Experimental Setup

We use two series connected modules of the origami arm [12], [13] to test the performance of our proposed grow-to-shape algorithm. Each module is controlled using 3 actuators that are connected to flexible tendons. In our work, we couple two of the tendons in each module in software thereby individually controlling two tendons in each module of the manipulator. This constrains the motion of the manipulator to a plane. An image of our experimental setup is shown in Fig. 6. The manipulator is mounted, facing downward, to a horizontal bar. An Intel Realsense d435i camera is used in an eye-to-hand setting to obtain real-time high resolution (1280 X 720 pixels) RGB images of the manipulator. Two ArUco markers are pasted at the base and end effector of the robot to obtain their respective poses in the image. The manipulator is powered using a bench top power supply set to 6V. At the beginning of each experiment, both modules of the robot are set to their minimum length. A static transform between the tip of the first module and the robot's end effector is estimated by selecting the tip position of the first module in the image. The combined Jacobian is initialized with an intuitive guess of the system parameters and is adapted online to the specific system parameters for the active module and camera parameters. The control frequency for our experiments is set to $10Hz$. The rest of this section presents the experiments conducted to test the performance of our algorithm and discusses their outcomes.

B. Grow to Shape Validation

We experimentally demonstrate the capability of our algorithm to grow the continuum manipulator along desired reference shapes in the image. Experiments are conducted for 8 randomly selected reference poses of the end effector in different parts of the robot's workspace. Shape references for each of the selected poses are generated using the methods discussed in Section III-E. For each of these experiments,

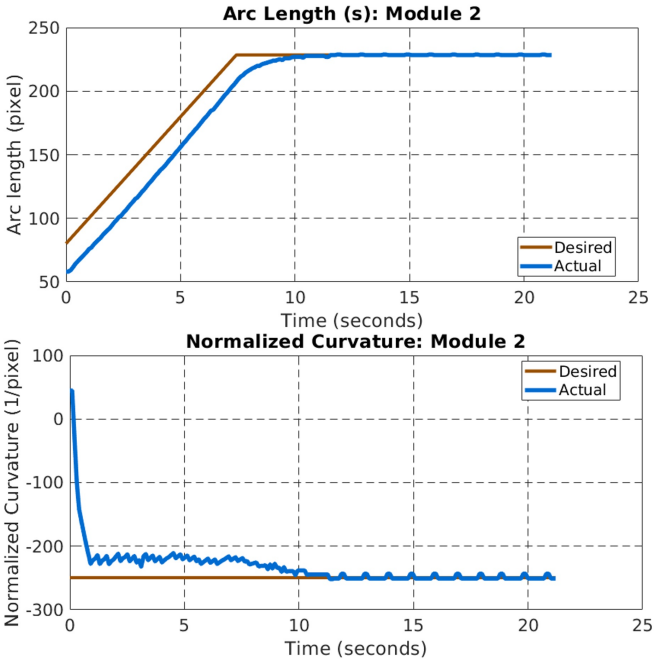


Fig. 5. Shape feature convergence plot for module 2 for the simulation experiment demonstrated in Fig. 3.

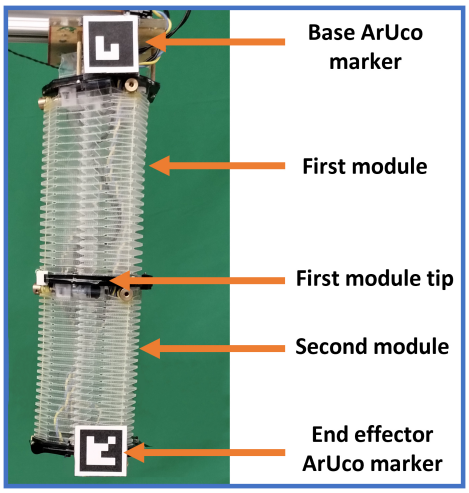


Fig. 6. Experimental setup showing the two module origami continuum manipulator.

the final shape of the robot along with the reference shape is shown in Fig. 7. In all the experiments, the final shape of the robot closely mimics the provided reference shape in the image. Evidence of this can be seen in Fig. 8 that shows the steady-state shape feature errors recorded for each of the modules in these experiments. The shape feature plots for the experiment demonstrated in Fig. 7f are shown in Fig. 9 and Fig. 10. They demonstrate our controller's ability to closely track the growing shape reference for the active module and converge to the final shape with minimal steady state error in the image.

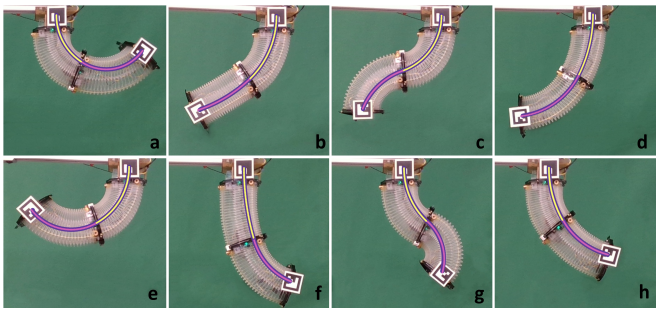


Fig. 7. Eight different experiments validating grow-to-shape performance to different shape references in the image. Desired shape references are depicted with purple curves, and the final shape achieved by module 1 and module 2 is shown by the yellow and pink curves respectively.

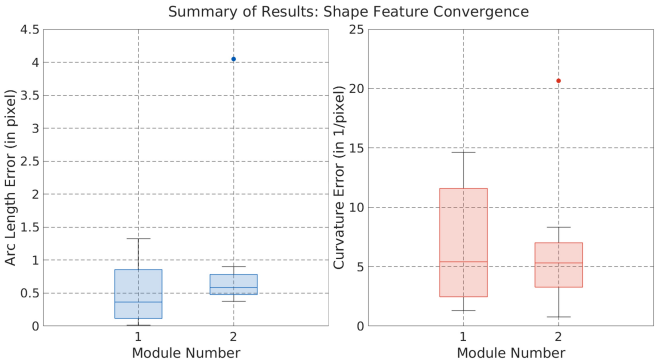


Fig. 8. Box plot showing the mean and standard deviation of shape feature errors recorded for each module at steady-state.

C. Controller Performance Comparison

The response of the image-based controller is compared for varying speeds of the growing reference curve in the image. We use the experiment shown in Fig.7f as our baseline of comparison. The controller parameters, for the baseline trial, are tuned such that the overshoot in shape feature errors is always below 7%. The shape feature plots shown in Fig.11 are for 2 trials of the same experiment where the growing speed of the reference is varied. For each experiment, the shape features converge to their reference values with minimal steady state error. Since the speed of the growing reference is increased, we observe a slightly higher overshoot (within 10%) for these trials. These results demonstrate our controllers ability to track shape references with faster growing speeds in the image.

D. Proof of Concept: Obstacle Avoidance

In Fig. 12, we demonstrate proof of concept experiments to show that our controller can be applied to cluttered or constrained environments. Note, we do not explicitly implement obstacle avoidance as shown in [25]. Instead, we visually select reference shapes that are free of collisions. Our controller ensures the growing references are tracked closely to avoid collisions (as seen in the supplementary video) in the environment while still ensuring convergence to the final shape with a smooth transient response.

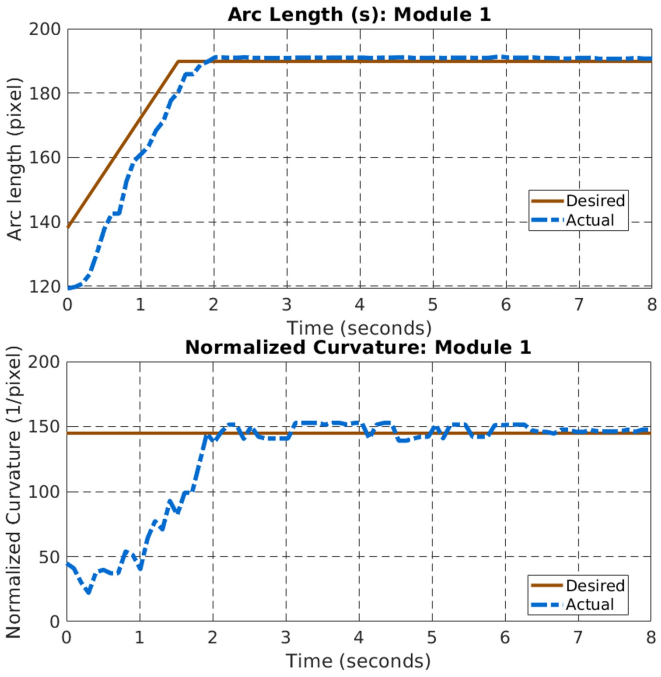


Fig. 9. Feature convergence plot for shape features (s_1, κ_1) of module 1 for the experiment shown in Fig. 7d.

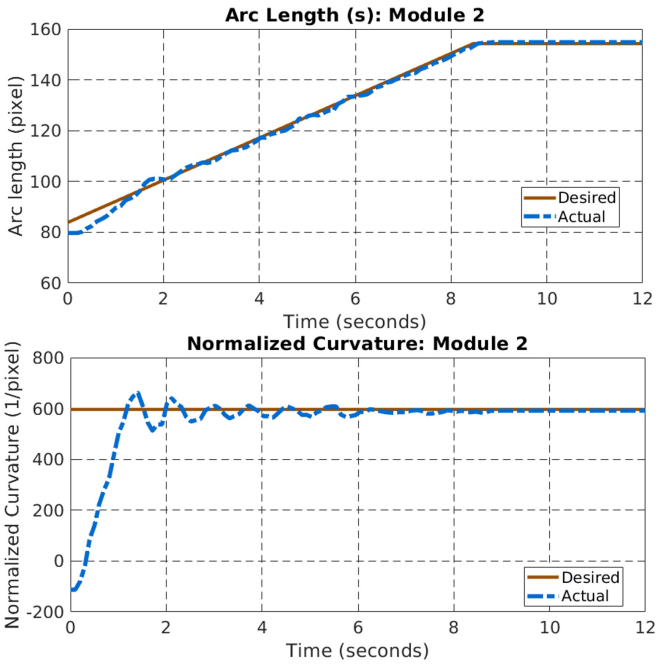


Fig. 10. Feature convergence plot for shape features (s_2, κ_2) of module 2 for the experiment shown in Fig. 7d.

E. Limitation

Our proposed grow-to-shape formulation is currently limited to 2-D. An approach to extending the capabilities of this algorithm to control the shape of the robot in 3-D is to use information from depth sensors to fit Piecewise Constant Curvature curves. Indeed, there will be challenges of noisy depth data that will need to be handled. Extending these capabilities to 3-D would enable us to use this work to address practical problems in various domains such as object manipulation, inspection, and assistive robotics.

VI. Conclusion & Future work

In this paper, we develop an adaptive vision-based shape controller that allows us to utilize the configuration redundancies of a novel variable length continuum robot while ensuring that the robot tracks desired shape curves in the image space as it expands from its minimum length to the desired shape. We first obtain a parametric representation of the manipulator's shape in the image by fitting piecewise constant curvature arcs. The arc features of each piecewise segment are used as image features for visual servoing. Notably, we do not use markers or fiducials on the entire robot's body to compute our image features. We only use markers to track the base and end effector pose of the robot in the image and this information is not used in the control loop. An adaptive visual servoing controller that does not require any initial excitation velocities is applied to control each module of the manipulator such that it tracks a growing reference shape in the image. Further, our controller does not use any information from the robot's sensors, we only utilize information about the robot's morphology in the image. Experiments are performed on a numerical simulation as well as a two module origami arm hardware. The experimental results demonstrate that the manipulator is able to accurately track the growing shape references in the image using our approach.

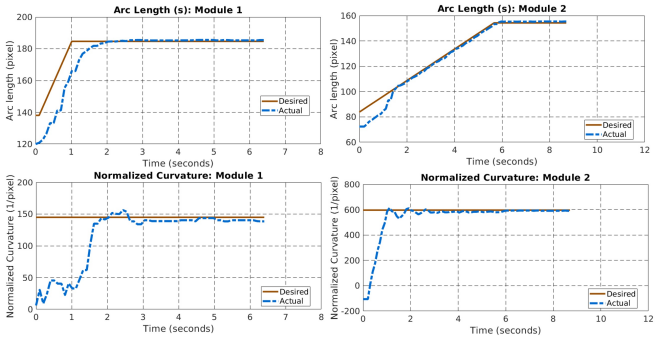
This work is inspired from reaching motions observed in nature. Such motions enable complex environmental interactions such as object manipulation and inspection in unstructured settings. The use of vision-based control to close the shape control loop is advantageous to incorporate task relevant information in the control cycle. Our work is fundamental to ensuring safe operation of variable length robots in cluttered and constrained environments. It unlocks an important shape-based control skill for reaching areas or objects of interest with a variable length continuum arm in such environments. Further investigation in robust techniques for estimating the whole body configuration of the continuum arm in the image without relying on static transforms would enhance the ability of our system to react to unexpected disturbances to the inactive module. Additionally, expanding this work to control the robot in 3-D is of interest to us.

VII. Acknowledgements

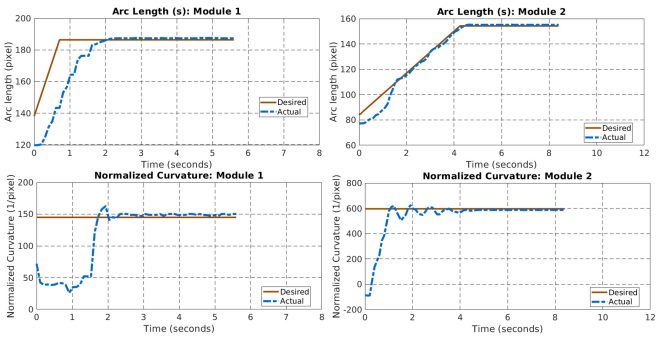
The authors would like to acknowledge Jennifer Shaughnessy, who is a student at Massachusetts Academy of Math and Science, whose contributions have been very meaningful towards the completion of this work.

References

[1] M. Cianchetti, C. Laschi, A. Menciassi, and P. Dario, "Biomedical applications of soft robotics," *Nature Reviews Materials*, vol. 3, no. 6, 2018.



(a) The growing reference speed is increased by 50% when compared to the baseline shown in Fig. 7f.



(b) The growing reference speed is increased by 100% when compared to the baseline shown in Fig. 7f.

Fig. 11. Shape feature convergence plots for trials comparing controller performance when the speed of growing references is increased.

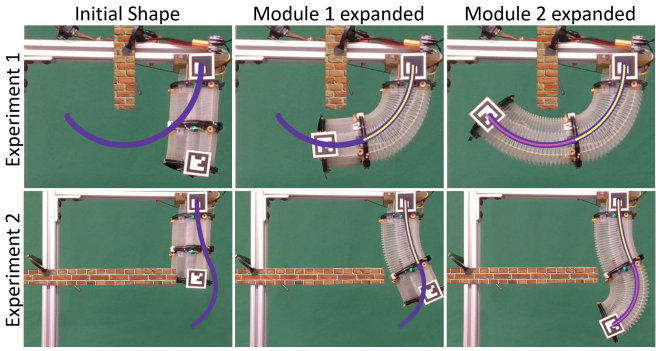


Fig. 12. The continuum manipulator grows along the desired shape reference while avoiding obstacles in the workspace to reach the final shape targets in the image.

[2] J. Fathi, T. J. Vrielink, M. S. Runciman, and G. P. Mylonas, "A deployable soft robotic arm with stiffness modulation for assistive living applications," in *IEEE International Conference on Robotics and Automation*, 2019.

[3] Y. Zhang and M. Lu, "A review of recent advancements in soft and flexible robots for medical applications," *International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 16, no. 3, jun 2020.

[4] K. Liu, W. Chen, W. Yang, Z. Jiao, and Y. Yu, "Review of the research progress in soft robots," *Applied Sciences*, vol. 13, no. 1, 2022.

[5] S. Chen, Y. Cao, M. Sarparast, H. Yuan, L. Dong, X. Tan, and C. Cao, "Soft crawling robots: design, actuation, and locomotion," *Advanced Materials Technologies*, vol. 5, no. 2, 2020.

[6] X. Wang, Y. Li, and K.-W. Kwok, "A survey for machine learning-based control of continuum robots," *Frontiers in Robotics and AI*, p. 280, 2021.

[7] 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, Jun. 2023.

[8] A. Gandhi, S.-S. Chiang, C. D. Onal, and B. Calli, "Shape Control of Variable Length Continuum Robots using Clothoid-based Visual Servoing," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2023.

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

[10] M. Hussein, "A review on vision-based control of flexible manipulators," *Advanced Robotics*, vol. 29, no. 24, 2015.

[11] J. D. Greer, T. K. Morimoto, A. M. Okamura, and E. W. Hawkes, "A Soft, Steerable Continuum Robot That Grows via Tip Extension," *Soft Robotics*, vol. 6, no. 1, Feb. 2019.

[12] J. Santoso and C. D. Onal, "An Origami Continuum Robot Capable of Precise Motion Through Torsionally Stiff Body and Smooth Inverse Kinematics," *Soft Robotics*, Jul. 2020.

[13] J. Santoso, E. H. Skorina, M. Luo, R. Yan, and C. D. Onal, "Design and analysis of an origami continuum manipulation module with torsional strength," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sep. 2017.

[14] F. Chaumette, S. Hutchinson, and P. Corke, "Visual servoing," in *Springer Handbook of Robotics*. Springer, 2016.

[15] Y. Zhao, L. Gong, Y. Huang, and C. Liu, "A review of key techniques of vision-based control for harvesting robot," *Computers and Electronics in Agriculture*, vol. 127, 2016.

[16] M. Jagersand, O. Fuentes, and R. Nelson, "Experimental evaluation of uncalibrated visual servoing for precision manipulation," in *IEEE International Conference on Robotics and Automation*, 1997.

[17] A. Shademan, A.-M. Farahmand, and M. Jägersand, "Robust jacobian estimation for uncalibrated visual servoing," in *IEEE International Conference on Robotics and Automation*, 2010.

[18] A. Gandhi, S. Chatterjee, and B. Calli, "Skeleton-based Adaptive Visual Servoing for Control of Robotic Manipulators in Configuration Space," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2022.

[19] H. Wang, B. Yang, Y. Liu, W. Chen, X. Liang, and R. Pfeifer, "Visual Servoing of Soft Robot Manipulator in Constrained Environments With an Adaptive Controller," *IEEE/ASME Transactions on Mechatronics*, vol. 22, feb 2017.

[20] J. Hao, K. Zhang, Z. Zhang, S. Wang, and C. Shi, "An Online Model-Free Adaptive Tracking Controller for Cable-Driven Medical Continuum Manipulators," *IEEE Transactions on Medical Robotics and Bionics*, vol. 5, no. 3, pp. 623–635, Aug. 2023.

[21] 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, Aug. 2014.

[22] F. Xu, Y. Zhang, J. Sun, and H. Wang, "Adaptive Visual Servoing Shape Control of a Soft Robot Manipulator Using Bèzier Curve Features," *IEEE/ASME Transactions on Mechatronics*, pp. 1–11, 2022.

[23] R. 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, Nov. 2010.

[24] F. Chaumette and S. Hutchinson, "Visual servo control. II. Advanced approaches [Tutorial]," *IEEE Robotics & Automation Magazine*, vol. 14, no. 1, pp. 109–118, Mar. 2007.

[25] S.-S. Chiang, M. da Silva, D. Moyer, and C. D. Onal, "Amorph: Analytical model-based real-time posture heuristic inverse kinematics solver for continuum robots," in *Experimental Robotics: The 18th International Symposium*. Springer, 2023, (in-press).

[26] S.-S. Chiang, "Real-time and customizable inverse kinematics for extensible multisection continuum robots: Slinki and amorph algorithms," Ph.D. dissertation, Worcester Polytechnic Institute, 100 Institute Road, Worcester MA 01609-2280 USA, December 2023.

[27] B. A. Jones and I. D. Walker, "Kinematics for multisection continuum robots," *IEEE Transactions on Robotics*, vol. 22, no. 1, 2006.

[28] A. Gandhi, S.-S. Chiang, C. D. Onal, and B. Calli, "Simulation Package for Origami Arm," Mar. 2024. [Online]. Available: https://github.com/Abhinavgandhi09/origami-arm-sim