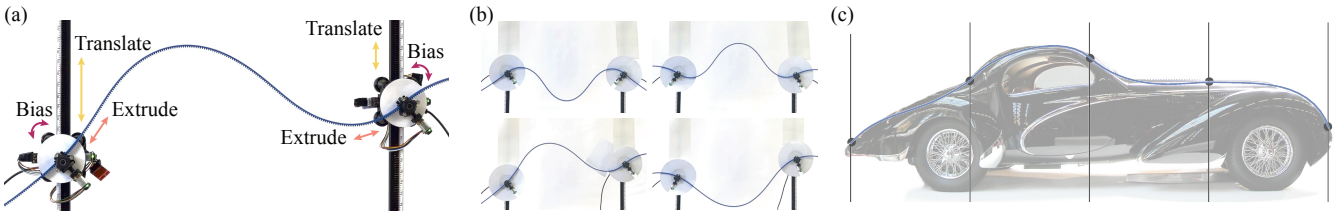
2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

A Multi-Stable Curved Line Shape Display

Wing-Sum Law[1], Sofia Di Toro Wyetzner[2], Raymond Zhen[3], and Sean Follmer[3]



(a) Our curved line display and its degrees of freedom. (b) Four examples of curves it can create. (c) A simulation of a multi-segment display creating the shape of the top of a Talbot-Lago (photo by Jack Snell, CC BY 2.0. Background removed and image overlaid).

***Abstract*— Shape-changing displays enable real-time visualization and haptic exploration of 3D surfaces. However, many shape-changing displays are composed of individually actuated rigid bodies, which makes them both mechanically complex and unable to form smooth surfaces. In this work, we build a multi-stable curved line display inspired by physical splines. By using circular splines to initialize a discrete elastic rods simulator, we can model multiple stable shapes that fit specific boundary conditions. We then generate actuation instructions based on the circular spline initialization to drive the physical display. We demonstrate our display's ability to create 16 shapes with 8 different boundary conditions. Our display is consistent in shape output, with an average standard deviation in height of 0.75 mm or 0.47% of the display's maximum vertical range. We also show that our model is consistent with our display, with a mean RMSE of 6.68 mm or 3.85% of the display's maximum vertical range for shapes we could stably simulate. We then demonstrate potential scalability by simulating a multi-segment version of the system and show the display's ability to withstand loads during contour following in haptic exploration.**

I. Introduction

While much of modern design is done with computational tools, physical models remain an important part of the prototyping process. Physical prototyping is essential, from automotive design [1] to architecture [2], allowing multiple users to simultaneously view and interact with a tangible design. Previous work has shown that physical models give a richer impression of a prototype's aesthetic and function than would be available through a digital image alone, mitigating design fixation [3]. However such models are often static and must be entirely re-fabricated to reflect iteration.

One way to empower designers and engineers with physical feedback is through the development of shape-changing displays, which are a class of robotic systems that dynamically generate multiple physical forms. Pin displays are a well-studied type of shape-changing display. They are composed of an array of linear actuators that move up and down in concert to discretely approximate 1.5D [4] and 2.5D surfaces [5]–[7]. This type of device has been used as a computational design tool [8]. Another type of shape-changing display are displays that approximate shapes using an articulated series [9] or grid of rigid bodies connected by actuated hinges [10]–[12], otherwise known as a formable crust display. Both aforementioned types of shape-changing displays have similar limitations. First, they are mechanically complex and, therefore, difficult to scale. Second, they can only discretely approximate smooth surfaces due to being composed of discrete, rigid bodies.

In this work, we propose a curved line display inspired by physical splines [13] as a step toward minimally-actuated, continuous surface displays. We designed and built the system pictured in Fig. a that manipulates a flexible rod by controlling rod length, boundary positions, and boundary angles. Flexible rods are multi-stable when deformed by their end points [14], complicating simulation and control, but also enabling a larger shape space. To determine the actuation strategy for our under-actuated, multi-stable system, we employed a discrete elastic rods model [15], [16]. To model multi-stability, we developed a novel optimization scheme over circular splines to initialize the simulator. We then used the actuation strategy determined from our model to demonstrate the physical display's ability to reliably generate a variety of shapes (Fig. b). These shapes were validated against the simulation. We then simulated a multi-segment version of the curved line display (Fig. c) as a preliminary demonstration of scalability. Finally, we tested the system loading response to determine fitness for haptic exploration.

II. Related Work

Attempts have been made at augmenting pin displays and formable crust displays to better display continuous surfaces. Some have augmented the pin display with a spatial low pass filter, in which a material is stretched over the pin display surface [5], [17]. While this approach partially smooths out the pins, the material does not perfectly interpolate a smooth surface, leaving the final shape with bumpy artifacts [18].

*This work was supported by the National Science Foundation (NSF) grant no. 1925030. This work was also supported by two NSF Graduate Research Fellowships for the first two co-authors, grants no. DGE-1656518 and no. DGE-2146755.

[1]Department of Mechanical Engineering, Stanford University wslaw@stanford.edu
[2]Department of Computer Science, Stanford University
[3]Department of Mechanical Engineering, Stanford University

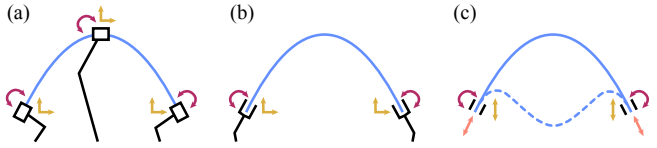


Fig. 1. (a) Direct control of points along continuous display. (b) End point control of continuous display. (c) Proposed single-segment, bi-stable curved line display.

Layer jamming is a way to augment the formable crust display to display smoother surfaces, replacing previously rigid grids with flexible materials [19], [20]. This approach is also smoother than the pin display, but requires selectively locking segments along the surface itself, constraining display resolution to the grid size. Some works combine aspects of these approaches by using rigid actuators to bend, stretch, and twist flexible material by directly actuating points (Fig. 1a) [21]–[23] on a flexible material, or actuating a shape memory alloy embedded in a flexible material [24].

Compliant rods present a promising alternative to purely rigid or soft materials by allowing two fixed actuators at either end to control a large state space (Fig. 1b). Prior works in computer graphics have used this paradigm as an input method [25], a fabrication tool [26], and a haptic display [27]. These inherently smooth (continuous) devices come with their own challenges. While they do produce smooth lines or surfaces, all of these devices use a fixed length or area of deformable material for their displays (Fig. 1a & b). This means that while these devices can produce continuous surfaces, they are limited in curvature by the area or elasticity of the flexible material employed. Our insight is that it may be possible to overcome this limitation by injecting or removing material from a display (Fig. 1c).

In addition to the shape space limitations, the accuracy of shape generation in smooth, shape-changing displays is more challenging than in their discretely actuated counterparts. In the cases where devices directly actuate points on a flexible material [21]–[23], the devices have positional control over the end effectors of their actuators, but do not measure or control the shapes of the intervening flexible material. A validated computational model for the bending of material between actuators would not provide real-time positional information, but it would provide additional geometric information without requiring additional sensors or cameras. To this end, we propose a forward solution to model the rod's nonlinear deformation in between adjacent actuators and show agreement with our constructed device.

One of the challenges in modeling flexible rods is their multi-stability under certain boundary conditions [14], [28]. Our system utilizes sets of boundary conditions that produce up to two stable states. To find these, we optimize over spline control points, then discretize and relax these splines into physically stable curves. The system multi-stability means that to control the output shape, we must understand how to reach a specific stable state. This both motivates our investigation of different deployment strategies and expands the accessible shape space of our device.

III. Design of a Multi-Stable Curved Line Display

A. Mechanical Design

A single segment of this display is comprised of a flexible rod actuated between two individually operated nodes (Fig. 2a). As shown in Fig. 2a, the controllable dimensions of the system are the length of the flexible rod between the nodes (L), the height between the nodes (y), and the angle of the tangent vectors ($\mathbf{t_1}$ and $\mathbf{t_2}$) at each end of the curved line (θ_1 and θ_2, respectively). The discrete elastic rods model uses a perfectly elastic rod with a constant cross-section. However, for ease of actuation, we chose a flexible rack (KHK DR1-2000, Duracon) as the flexible rod for this curved line display, which is not perfectly elastic.

Each node has three actuated mechanisms. The extrude mechanism is a locally prismatic joint through which the flexible rod moves. It controls how much of the rod is on either side of the node (Fig. 2b, orange). The value of L between the two nodes is controlled by the extrude mechanisms on either side of the segment working in concert. The biasing mechanism is a revolute joint that controls the angle (θ_i) at which the rod passes through the node (Fig. 2b, magenta), corresponding to the node's tangent vector, $\mathbf{t_i}$. Rigid bearing surfaces above and below the rod constrain a 20 mm section of the rod to the desired angle at the center of

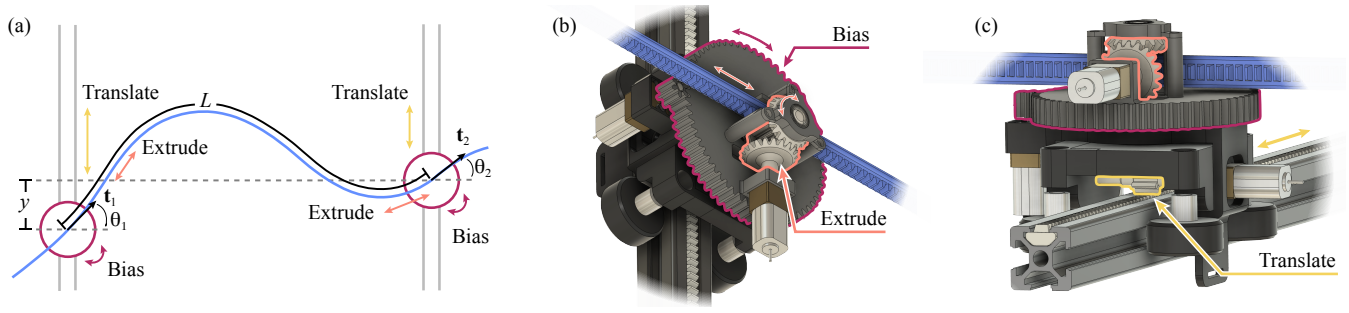


Fig. 2. (a) Curved line display between two nodes showing the controllable dimensions of the system: the length of flexible rod between the nodes (L), the height between the nodes (y), and the angle of the tangent vectors ($\mathbf{t_1}$ and $\mathbf{t_2}$) at each end of the curved line (θ_1 and θ_2, respectively). The two nodes are a fixed 500 mm apart horizontally. CAD model of node module of the curved line display showing (b) extrude mechanism labeled with orange, bias mechanism labeled with magenta, and (c) translate mechanism labeled with yellow.

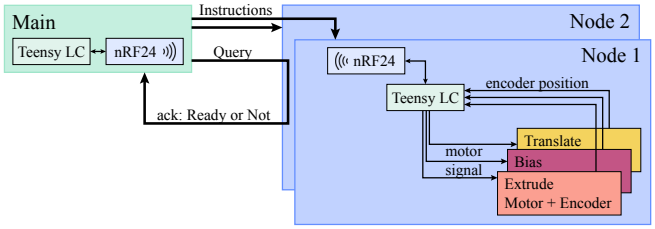


Fig. 3. Diagram of electronic components in main and nodes and the communication between main and nodes.

rotation for the bias mechanism. The translation mechanism is a prismatic joint that controls where the node is along a 20 mm x 20 mm aluminum extrusion (Fig. 2c, yellow), which corresponds to y. There is an aluminum extrusion on either side of the curved line segment for each node to travel on, as pictured in Fig. 2c, and these aluminum extrusions are fixed at 500 mm apart horizontally.

B. Electronics and Communication

Each node is an individually operated robot, with multiple nodes working simultaneously to actuate the curved line display. We used an N20 geared DC motor with an attached magnetic encoder (Adafruit 4641) driven via a motor driver chip (Texas Instruments DRV8833RTYT) to actuate each degree of freedom (extrude, bias, translate). A Teensy LC microcontroller manages the logic computation and communication (Fig. 3). A main computer was used to wirelessly communicate pre-determined shape generation instructions with each node using a 2.4 GHz radio chip (nRF24L01+ chip on SparkFun WRL-00691 breakout board). Motors, encoders, and the wireless communication module are attached to the microcontroller using embedded electronics on a custom PCB (Bay Area Circuits). Each node is continuously polled for state information. When all nodes are finished with their current instruction, the main computer sends out the next step from the generated instructions.

IV. Forward Solution Using Discrete Elastic Rods

While some analytical solutions exist for elastic rods [30], [31], they exist only under specific boundary conditions, requiring a numerical model to describe the bending of flexible rods in most practical applications. One popular approach is to approximate a flexible rod or strip as being a series of discrete pieces. This approach has been used to describe the positioning of continuous robotic parts [32], [33] and estimate the shape of a deformed rod manipulated by its ends [26], [34]–[36]. Such models have not yet been adopted in the design of shape-changing displays.

We rely on the discrete elastic rods method [15], [16], a model that discretizes Kirchhoff rods and computes bending, twisting, and stretching energy on the discrete segments. This model and has been physically validated [37] and can accommodate different rod material properties and cross-sections. We use the manufacturer-given young's modulus and density along with a rectangular cross-section to approximate the notched rod in our device. Because the energy landscape of elastic rods is non-convex, we witness multi-stability, where different stable states lie at local minima. We first discuss the mathematical boundary conditions of our system, then generate different initializations to try to push the solver to converge to different stable states.

A. Mathematical Boundary Conditions

One module of our system consists of an elastic strip coupled to a vertical frame at two points. The boundary conditions for one module of our system consist of the left and right nodes' relative vertical position (y) and tangent vectors ($\mathbf{t_1}$ and $\mathbf{t}_2$), and the flexible strip's arc length between nodes (L). Given these conditions alone, we cannot uniquely determine the output shape of the strip, due to system multi-stability. We can however choose the output shape by using different initializations to our solver that satisfy the boundary conditions but converge to different final shapes (Sec. IV-B).

Our strip can be approximated by a 1D curve that is an embedding in the frame plane of a real-valued function $c(t) : [0, 1] \rightarrow \mathbb{R}^2$. The endpoints of this curve are given as two planar points, $\mathbf{p}_1$ and $\mathbf{p}_2$, and we set our frame origin to $\mathbf{p}_1$. The vertical component of $\mathbf{p}_2$ is a degree of freedom, y:

$$c(0) = \mathbf{p_1} = \{0, 0\}, \quad c(1) = \mathbf{p_2} = \{x_2, y\} \in \mathbb{R}^2 \qquad (1)$$

There is a variable tangent vector at each of these points,

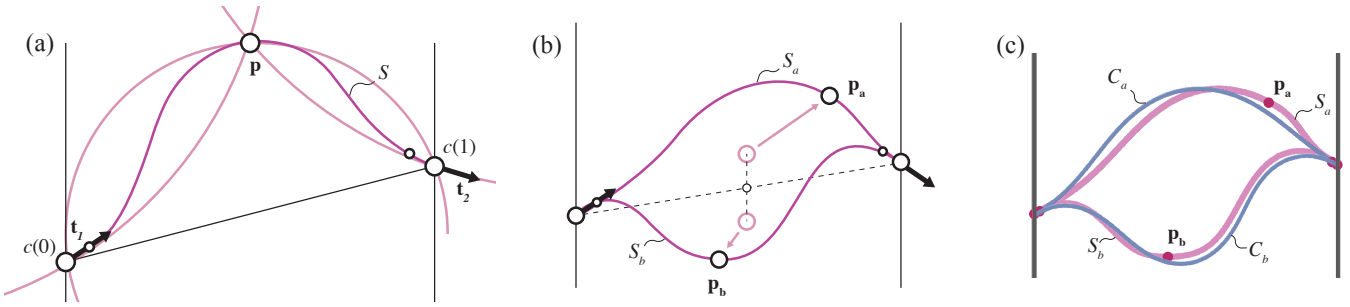


Fig. 4. Diagram of the curve initialization process. (a) A curve used to initialize the solver with the boundary conditions $c(0)$, $c(1)$, $\mathbf{t_1}$ and $\mathbf{t}_2$. This curve, S, is a spline creating by blending interpolating circles (see [29]). (b) We optimize over control points $\mathbf{p}_a$ and $\mathbf{p}_b$ to match a given arc length and minimize the splines' curvature. (c) These two splines, S_a and S_b, which have equal boundary conditions, are used to initialize the solver and relax into physically-stable curves C_a and C_b.

which have unit length (Fig. 4a):

$$c'(0) = \mathbf{t_1}, c'(1) = \mathbf{t_2} \in \mathbb{R}^2$$
$$|\mathbf{t_1}| = |\mathbf{t_2}| = 1 \tag{2}$$

Additionally, we have an arc length of $c(t)$, $L \in \mathbb{R}$, which corresponds to the length of injected material, which we define as:

$$L = \int_0^1 \sqrt{1 + c'(t)} dt \tag{3}$$

Based on empirical observations, we assume that there are two or fewer achievable stable states for each set of boundary conditions. These states can be loosely defined as (1) where the rod's material is mostly above the endpoints and (2) below it (Fig. 4b). We built on this observation by using spline integrals to reduce the infinite-degree problem space to a finite number of physically plausible solutions.

B. Finding Initializations to the Simulator

Our goal is to develop an initialization that can match the natural shapes of a flexible rod and provide a close point from which we can simulate the relaxed state. The Kirchhoff model for elastic rods uses a term that minimizes geometric curvature to capture rod bending energy [15]. Therefore we want an initial geometry that is smooth and twice differentiable such that curvature is a defined quantity. Rather than use a single function to describe this possibly complex initial curve, we opt to use a spline, which we can easily edit through its control points.

Our work uses splines based on a C^2 blending function, adapted from Yuksel et al [29]. The Yuksel formulation with circular interpolation offers several advantages for our problem. It is interpolating, which allows us to encode tangent constraints as two points along a vector of infinitesimal length. This circular interpolation also creates splines that are close to the final stable shapes they produce (Fig. 4c). We posit that this is because the geometric properties of Kirchhoff rods – the minimization of bending energy often corresponds with the minimization of curvature inherent in circular splines. We use five control points per spline, with two points on each end representing the endpoints and the tangent constraints respectively, and one point in the center, which we use to control the end shape (Fig. 4a). We call this central point $\mathbf{p}$. We can adjust $\mathbf{p}$ to both match our arc length constraint and find an initialization that is close to our final relaxed shape (Fig. 4b).

To find these initializing splines we set up a light-weight optimization problem over the coordinates of $\mathbf{p}$. The objective function both minimizes planar curvature and uses a least-squares approach to match the arc length of the spline to the amount of injected rod. We want to minimize planar curvature to find the spline that fits the bending energy in our discrete elastic rods model.

We estimate curvature and arc length, which are integrated quantities, with a discrete sum over n samples of the spline S, $s(t) \in \mathbb{R}^2$. This estimation becomes more accurate as n grows. We first discretize our spline into n points:

$$s_n(t) = s(t/n) \tag{4}$$

$$e_n(t) = s_n(t+1) - s_n \tag{5}$$

where $t \in [0, \ldots, n-1]$. And $e_n(t)$ refers to the edge between spline samples $s_n(t)$ and $s_n(t+1)$. We then approximate arc length with the following sum:

$$l = \sum_{t=0}^{n-1} ||s_n(t+1) - s_n(t)|| \tag{6}$$

where l is the approximate arc length of the curve. We approximate curvature, κ, by looking at the change in tangent directions with each pair of edges. This is normalized by the Voronoi length region around the point the edges share:

$$\kappa = \sum_{t=0}^{n-2} \frac{1 - \frac{e_n(t)}{||e_n(t)||} \cdot \frac{e_n(t+1)}{||e_n(t+1)||}}{\frac{1}{2}||e_n(t)|| + \frac{1}{2}||e_n(t+1)||} \tag{7}$$

Finally we can set up our optimization problem,

$$\mathbf{p}^* = \text{argmin}_{\mathbf{p}}(\kappa^2 + \alpha(l - l_0)^2) \tag{8}$$

where we find the central spline control point $\mathbf{p}^*$ that achieves the desired arc length. We use a soft constraint on arc length with constant $\alpha = 10$ while minimizing curvature squared, κ^2. This objective function is solved using automatic differentiation [38] along with Newton's method and a line search. To initialize this optimization we start with the midpoint of the two boundary points and offset it vertically by a small amount up and down (see Fig. 4b). This optimization typically converges in fewer than 10 iterations.

V. Shape Generation on a Curved Line Display

Actuation instructions to match simulated shapes were generated based on the spline initialization corresponding to that shape. Following the intuition described in Sec. IV-A, we first actuate such that the majority of the material is close to the final shape's center of mass, then configure end conditions. Each degree of freedom is controlled by a

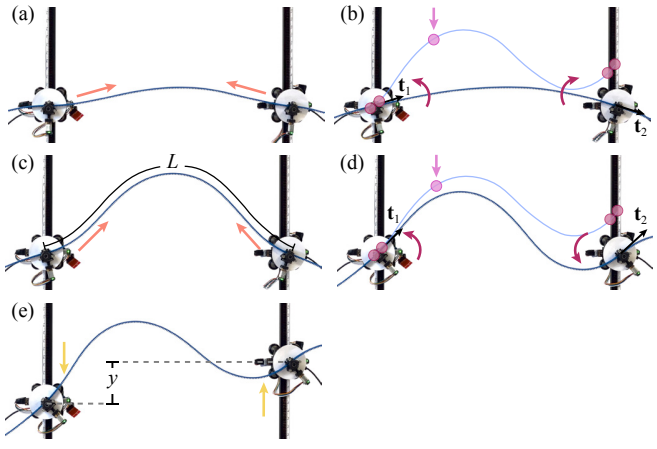


Fig. 5. Shape generation occurs by (a) extruding enough material to allow biasing, (b) biasing toward the central control point (the pink circle), (c) extruding until the interior rod is the desired arc length, (d) biasing to match desired boundary conditions, and (e) translating to desired height difference. Actuation steps pictured are annotated still frames pulled from a video of the device being operated.

motor-encoder pair, making position control of each degree of freedom closed loop. Overall shape control is open loop, as the current system lacks sensors that would detect which of the stable shapes the system is in. The following actuation order was determined empirically to reliably generate the same shape on a single-segment curved line display starting with both nodes at the same height ($y = 0$) and with the flexible rod at each node horizontal to the ground ($\theta = 0$):

1) Extrude symmetrically until the length of flexible strip between nodes has enough extra material to allow for biasing on either end (Fig. 5a).
2) Bias the nodes on either side of segment to the bias angles such that the lowest curvature arc between tangent vectors $\mathbf{t_1}$ and $\mathbf{t_2}$ bends in the same direction as the central control point $\mathbf{p}$ (Fig. 5b).
3) Extrude symmetrically until the length, L, of the flexible strip matches that of objective shape (Fig. 5c).
4) Bias the nodes on either side of the segment to bias angles θ_1 and θ_2 such that they match those of the objective shape (Fig. 5d).
5) Translate the nodes until height, y, between nodes is reached (Fig. 5e).

We used this sequence of actuation instructions to control the curved line display for every shape generated in the following section.

A. Comparison of Physical Display Segment to Model

In order to test the consistency of the display and compare the model to the physical display, we generated a series of shapes both in simulation and physically. Boundary conditions were chosen to demonstrate the variety of achievable shapes. We tested bias angles that were neutral, biased up, biased down, and asymmetrically biased with the nodes either vertically level or vertically staggered. Based on simulation initializations, different actuation instructions for each shape were then generated as described in Sec. V.

Physical shape generation was recorded using an iPhone 13 camera. Still frames were then exported and corrected for lens distortion using an AprilTag calibration [39]. Still frames were then converted to binary for analysis.

The results of these experiments are shown in Fig. 6. Over a total of 16 different shapes, the overall mean standard deviation within $N = 3$ trials over all 16 shapes of 0.75 mm or 0.47% of the display's maximum vertical range. These results show that our curved line display can consistently generate the same shape given the same actuation instructions, and can do so for a variety of different final shapes. For most of the tested shapes, there is also good agreement between the simulation and the physical display. Excluding the two shapes marked with ** in Fig. 6, the mean RMSE between the height physical display and our model was 6.68 mm or 3.85% of the display's maximum vertical range.

We were unable to match the physical display for the two shapes marked with ** in Fig. 6. Both of these shapes had asymmetric boundary conditions and a stable "peak" shape on the physical display, but when we simulated the display using those boundary conditions, the final shape collapsed into the other stable configuration. If the rightmost boundary condition is allowed to be more horizontal, we can simulate a shape with better agreement with the physical system (Fig. 6, curves marked with * in the rightmost column). We hypothesize that the discrepancy between the simulation and the physical display may be a result of the irregular flexible rack cross-section, or also the physical boundary hardware being incompletely rigid.

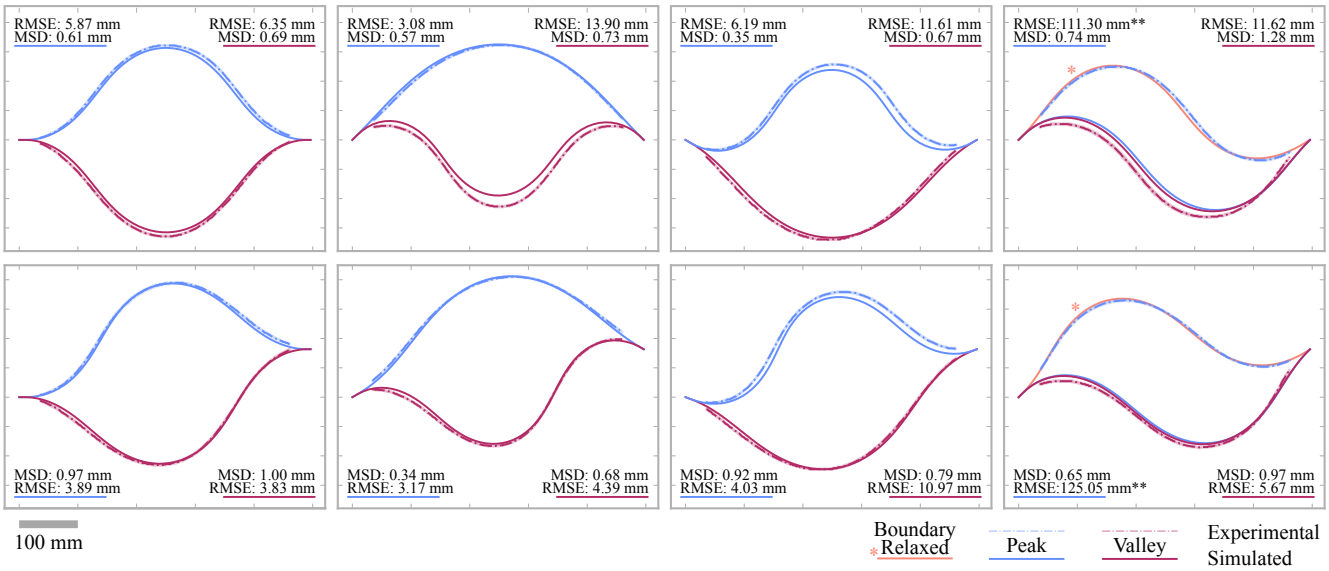


Fig. 6. Results comparing physical display to model. Each pane represents a set of boundary conditions, and each boundary condition has two associated shapes (a "peak" and "valley"). Experimental curves represent the mean of $N = 3$ trials per shape on the physical display. Mean $\pm$ standard deviation is shaded in, but due to relatively small standard deviation between trials is difficult to see. Mean standard deviation (MSD) between trials and root mean squared error (RMSE) between trials and simulation are reported on each pane. Curves marked with * represent simulations done with a more relaxed boundary condition.

B. Modeling Multiple-Segment Curved Line Display

Our simulator can be used to physically simulate a more complex display with multiple segments that would expand the shape space of our device. A multi-segment display offers a range of possibilities for physically creating controllable curves. We demonstrate a four-segment display in Fig. c, but there is no mathematical limit to how many segments may be added to the system by scaling the number of actuators. Note that while we can achieve smooth curves with high curvature, even a multiple-segment display cannot achieve discontinuities like corners or gaps in the output shape.

VI. Contact Forces on Curved Line Display

To evaluate the viability of our display for haptic surface exploration, we loaded two shapes at two different points along our curved line display and determined the deflection of that point in response to the applied force (Fig. 7). We added additional load to the force in 0.5 N increments until the curve snapped through to a different curve. Previous work in haptic texture exploration observed that most users employed less than 1 N of force when searching for a tactile target [40]. Another study in exploratory procedures found that users employ 2-5 N of force in contour following [41]. Our display can support these previously observed forces, but not the larger forces employed in other exploratory procedures such as pressure or enclosure.

VII. Conclusions and Future Work

In this work, we motivate the use of a curved line display in conjunction with a discrete elastic rods model as a method for creating a smooth line shape-changing display. We built a physical curved line display and then developed a model using the discrete elastic rods formulation. By employing a circular spline initialization, we can model the inherent multi-stability of a flexible rod under a single set of boundary conditions. This then allows us to generate actuation instructions for our curved line display that utilize multi-stability as a way of expanding the number of achievable shapes. Our display is consistent in output shape and matches well with our model. We demonstrate scalability by simulating a multi-segment version. Finally, we characterized the behavior of our display under loading, and we demonstrated that it can support contour following during haptic exploration.

There are a few notable drawbacks to our current system. While it has a large variety of shape outputs, it is lacking in tactile stiffness. Many shape-changing displays are made with haptic interaction in mind, but certain shapes on this display only support a few Newtons of load before buckling. Future work should consider improving mechanical robustness of the system. Furthermore, the current system controls final shape in an open-loop manner, which means that the system has no sensing for shape snap-through and recovery. Future iterations should consider instrumentation for closed-loop control. Another limitation of our system is that it can solve the forward problem of determining shape output from boundary conditions, but it does not solve the inverse problem of determining appropriate boundary conditions from a desired shape. Future work should implement the inverse solution to enable a curved line display to approximate any arbitrary curved shape. Finally, the current system only displays a single, flexible rod. We propose future work that configures multiple curved line displays in a grid, making it possible to display 2.5D surfaces.

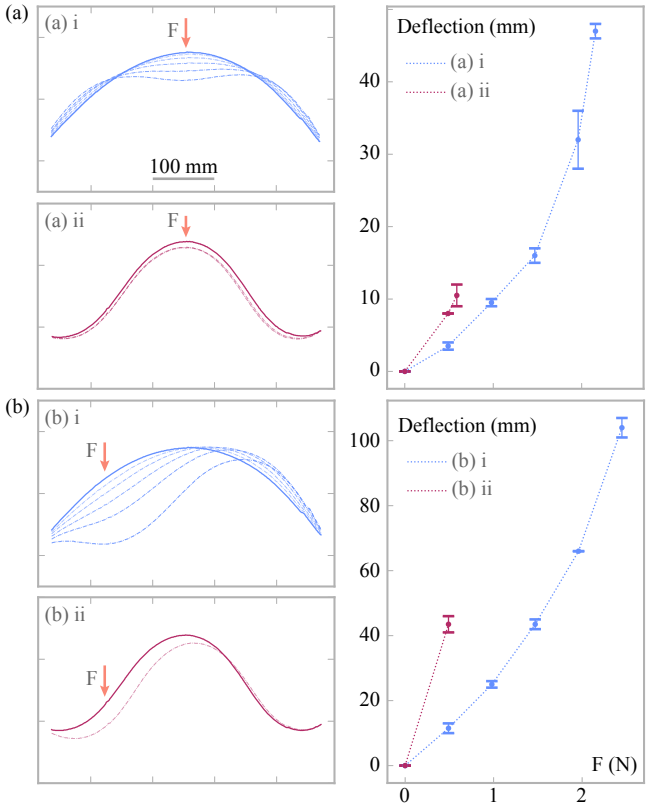


Fig. 7. A point load is applied at the spot indicated by the orange arrow labeled F at (a) the center of the shape and (b) off of the center of the shape. The left column of each sub-figure shows one example trial for each of the two shapes, with the unloaded rod shown as a solid line and the loaded rod as dashed. The right column shows the relationship between the force applied and the deflection of the point at which the force is applied. Each loading experiment was repeated $N = 3$ times.

Acknowledgment

Wing-Sum Law thanks Ahad Rauf for reviewing the PCB design and assisting with camera calibration, and Brandon Briones for testing motors. The authors acknowledge Danyang Fan for many conversations about rod materials and actuation strategies. The authors thank Savannah Cofer, Elizabeth Vasquez, Ahad Rauf, Danyang Fan, Yujie Tao, and Olivia Tomassetti for providing feedback on this paper.

References

[1] D. Gessner, "Why car companies still spend thousands on clay models," en, *Business Insider*, May 2022.

[2] I. Gibson, T. Kvan, and W. M. Ling, "Rapid prototyping for architectural models," *Rapid Prototyping Journal*, vol. 8, no. 2, pp. 91–95, Jan. 2002.

[3] R. J. Youmans, "The effects of physical prototyping and group work on the reduction of design fixation," *Design Studies*, vol. 32, no. 2, pp. 115–138, Mar. 2011.

[4] S. Jang, L. H. Kim, K. Tanner, H. Ishii, and S. Follmer, "Haptic edge display for mobile tactile interaction," in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, ser. CHI '16, San Jose, California, USA: Association for Computing Machinery, May 2016, pp. 3706–3716.

[5] H. Iwata, H. Yano, F. Nakaizumi, and R. Kawamura, "Project FEELEX: Adding haptic surface to graphics," in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, ser. SIGGRAPH '01, New York, NY, USA: Association for Computing Machinery, Aug. 2001, pp. 469–476.

[6] S. Follmer, D. Leithinger, A. Olwal, A. Hogge, and H. Ishii, "inFORM: Dynamic physical affordances and constraints through shape and object actuation," in *Proceedings of the 26th annual ACM symposium on User interface software and technology*, ser. UIST '13, St. Andrews, Scotland, United Kingdom: Association for Computing Machinery, Oct. 2013, pp. 417–426.

[7] A. F. Siu, E. J. Gonzalez, S. Yuan, J. B. Ginsberg, and S. Follmer, "Shapeshift: 2D spatial manipulation and Self-Actuation of tabletop shape displays for tangible and haptic interaction," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, ser. CHI '18, Montreal QC, Canada: Association for Computing Machinery, Apr. 2018, pp. 1–13.

[8] A. F. Siu, S. Kim, J. A. Miele, and S. Follmer, "shapeCAD: An accessible 3D modelling workflow for the blind and Visually-Impaired via 2.5d shape displays," in *Proceedings of the 21st International ACM SIGACCESS Conference on Computers and Accessibility*, ser. ASSETS '19, Pittsburgh, PA, USA: Association for Computing Machinery, Oct. 2019, pp. 342–354.

[9] K. Nakagaki, A. Dementyev, S. Follmer, J. A. Paradiso, and H. Ishii, "ChainFORM: A linear integrated modular hardware system for shape changing interfaces," in *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, ser. UIST '16, Tokyo, Japan: Association for Computing Machinery, Oct. 2016, pp. 87–96.

[10] J.-L. Huang, Z. Zhakypov, H. Sonar, and J. Paik, "A reconfigurable interactive interface for controlling robotic origami in virtual environments," *Int. J. Rob. Res.*, vol. 37, no. 6, pp. 629–647, May 2018.

[11] S. Klare, D. Forssilow, and A. Peer, "Formable object — a new haptic interface for shape rendering," in *2013 World Haptics Conference (WHC)*, Apr. 2013, pp. 61–66.

[12] A. Mazzone and A. Kunz, "Sketching the future of the SmartMesh wide area haptic feedback device by introducing the controlling concept for such a deformable multi-loop mechanism," in *First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. World Haptics Conference*, Mar. 2005, pp. 308–315.

[13] *When splines were physical objects*, en, `https://www.core77.com/posts/55368/When-Splines-Were-Physical-Objects`, Accessed: 2023-9-15.

[14] M. Vidulis, Y. Ren, J. Panetta, E. Grinspun, and M. Pauly, "Computational exploration of multistable elastic knots," *ACM Trans. Graph.*, vol. 42, no. 4, pp. 1–16, Jul. 2023.

[15] M. Bergou, M. Wardetzky, S. Robinson, B. Audoly, and E. Grinspun, "Discrete elastic rods," en, *ACM Trans. Graph.*, vol. 27, no. 3, pp. 1–12, Aug. 2008.

[16] M. Bergou, B. Audoly, E. Vouga, M. Wardetzky, and E. Grinspun, "Discrete viscous threads," en, *ACM Trans. Graph.*, vol. 29, no. 4, pp. 1–10, Jul. 2010.

[17] J. M. Lee, C. R. Wagner, S. J. Lederman, and R. D. Howe, "Spatial low pass filters for pin actuated tactile displays," in *11th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2003. HAPTICS 2003. Proceedings.*, Mar. 2003, pp. 57–62.

[18] D. Leithinger, D. Lakatos, A. DeVincenzi, M. Blackshaw, and H. Ishii, "Direct and gestural interaction with relief: A 2.5d shape display," in *Proceedings of the 24th annual ACM symposium on User interface software and technology*, ser. UIST '11, Santa Barbara, California, USA: Association for Computing Machinery, Oct. 2011, pp. 541–548.

[19] A. M. Rauf, J. S. Bernardo, and S. Follmer, "Electroadhesive auxetics as programmable layer jamming skins for formable crust shape displays," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, May 2023, pp. 2591–2597.

[20] A. A. Stanley, K. Hata, and A. M. Okamura, "Closed-loop shape control of a haptic jamming deformable surface," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 2718–2724.

[21] A. Steed, E. Ofek, M. Sinclair, and M. Gonzalez-Franco, "A mechatronic shape display based on auxetic materials," en, *Nat. Commun.*, vol. 12, no. 1, p. 4758, Aug. 2021.

[22] A. Mansutti, M. Covarrubias, M. Bordegoni, and U. Cugini, "Haptic strip based on modular independent actuators for virtual shapes rendering," in *2014 IEEE Haptics Symposium (HAPTICS)*, Feb. 2014, pp. 7–12.

[23] M. Bordegoni, F. Ferrise, M. Covarrubias, and M. Antolini, "A linear haptic interface for the evaluation of shapes," en, *ASME 2009 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pp. 1571–1579, Jul. 2010.

[24] M. Coelho, H. Ishii, and P. Maes, "Surflex: A programmable surface for the design of tangible interfaces," in *CHI '08 Extended Abstracts on Human Factors in Computing Systems*, ser. CHI EA '08, Florence, Italy: Association for Computing Machinery, Apr. 2008, pp. 3429–3434.

[25] T. Grossman, R. Balakrishnan, and K. Singh, "An interface for creating and manipulating curves using a high degree-of-freedom curve input device," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '03, Ft. Lauderdale, Florida, USA: Association for Computing Machinery, Apr. 2003, pp. 185–192.

[26] S. Duenser, R. Poranne, B. Thomaszewski, and S. Coros, "RoboCut: Hot-wire cutting with robot-controlled flexible rods," *ACM Trans. Graph.*, vol. 39, no. 4, 98:1–98:15, Aug. 2020.

[27] L. Zhu, X. Jiang, J. Shen, H. Zhang, Y. Mo, and A. Song, "TapeTouch: A handheld shape-changing device for haptic display of soft objects," en, *IEEE Trans. Vis. Comput. Graph.*, vol. 28, no. 11, pp. 3928–3938, Nov. 2022.

[28] C. Hafner and B. Bickel, "The design space of plane elastic curves," *ACM Trans. Graph.*, vol. 40, no. 4, pp. 1–20, Jul. 2021.

[29] C. Yuksel, "A class of C2 interpolating splines," *ACM Trans. Graph.*, vol. 39, no. 5, pp. 1–14, Aug. 2020.

[30] Y. V. Zakharov and K. G. Okhotkin, "Nonlinear bending of thin elastic rods," *J. Appl. Mech. Tech. Phys.*, vol. 43, no. 5, pp. 739–744, Sep. 2002.

[31] V. V. Kuznetsov and S. V. Levyakov, "Elastica of an euler rod with clamped ends," *J. Appl. Mech. Tech. Phys.*, vol. 41, no. 3, pp. 544–546, May 2000.

[32] C. B. Black, J. Till, and D. C. Rucker, "Parallel continuum robots: Modeling, analysis, and Actuation-Based force sensing," *IEEE Trans. Rob.*, vol. 34, no. 1, pp. 29–47, Feb. 2018.

[33] W. Huang, Z. Patterson, C. Majidi, and M. K. Jawed, "Modeling soft swimming robots using discrete elastic rod method," in *Bioinspired Sensing, Actuation, and Control in Underwater Soft Robotic Systems*, D. A. Paley and N. M. Wereley, Eds., Cham: Springer International Publishing, 2021, pp. 247–259.

[34] A. Sintov, S. Macenski, A. Borum, and T. Bretl, "Motion planning for Dual-Arm manipulation of elastic rods," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6065–6072, Oct. 2020.

[35] R. Takano, H. Mochiyama, and N. Takesue, "Real-time shape estimation of kirchhoff elastic rod based on force/torque sensor," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 2508–2515.

[36] C. Baek, A. O. Sageman-Furnas, M. K. Jawed, and P. M. Reis, "Form finding in elastic gridshells," en, *Proc. Natl. Acad. Sci. U. S. A.*, vol. 115, no. 1, pp. 75–80, Jan. 2018.

[37] V. Romero, M. Ly, A. H. Rasheed, *et al.*, "Physical validation of simulators in computer graphics: A new framework dedicated to slender elastic structures and frictional contact," *ACM Trans. Graph.*, vol. 40, no. 4, pp. 1–19, Jul. 2021.

[38] P. Schmidt, J. Born, D. Bommes, M. Campen, and L. Kobbelt, "TinyAD: Automatic differentiation in geometry processing made simple," en, *Comput. Graph. Forum*, vol. 41, no. 5, pp. 113–124, Aug. 2022.

[39] E. Olson, "AprilTag: A robust and flexible visual fiducial system," in *2011 IEEE International Conference on Robotics and Automation*, May 2011, pp. 3400–3407.

[40] A. M. Smith, G. Gosselin, and B. Houde, "Deployment of fingertip forces in tactile exploration," en, *Exp. Brain Res.*, vol. 147, no. 2, pp. 209–218, Nov. 2002.

[41] S. E. M. Jansen, W. M. Bergmann Tiest, and A. M. L. Kappers, "Identifying haptic exploratory procedures by analyzing hand dynamics and contact force," en, *IEEE Trans. Haptics*, vol. 6, no. 4, pp. 464–472, 2013.