

SoftSnap: Rapid Prototyping of Untethered Soft Robots Using Snap-Together Modules

Luyang Zhao^{1*}, Yitao Jiang¹, Chun-Yi She¹, Muhao Chen²,
Devin Balkcom^{1*}

^{1*}Department of Computer Science, Dartmouth College, Hanover, NH,
03755, USA.

²Department of Mechanical and Aerospace Engineering, University of
Kentucky, Lexington, KY, 40506, USA.

*Corresponding author(s). E-mail(s): luyang.zhao@dartmouth.edu;
devin.balkcom@dartmouth.edu;
Contributing authors: yitao.jiang@dartmouth.edu;
chun-yi.she@dartmouth.edu; muhao.chen@uky.edu;

Abstract

Soft robots offer adaptability and safe interaction with complex environments. Rapid prototyping kits that allow soft robots to be assembled easily will allow different geometries to be explored quickly to suit different environments or to mimic the motion of biological organisms. We introduce *SoftSnap* modules: snap-together components that enable the rapid assembly of a class of untethered soft robots. Each SoftSnap module includes embedded computation, motor-driven string actuation, and a flexible thermoplastic polyurethane (TPU) printed structure capable of deforming into various shapes based on the string configuration. These modules can be easily connected with other SoftSnap modules or customizable connectors. We demonstrate the versatility of the SoftSnap system through four configurations: a starfish-like robot, a brittle star robot, a snake robot, a 3D gripper, and a ring-shaped robot. These configurations highlight the ease of assembly, adaptability, and functional diversity of the SoftSnap modules. The SoftSnap modular system offers a scalable, snap-together approach to simplifying soft robot prototyping, making it easier for researchers to explore untethered soft robotic systems rapidly.

Introduction

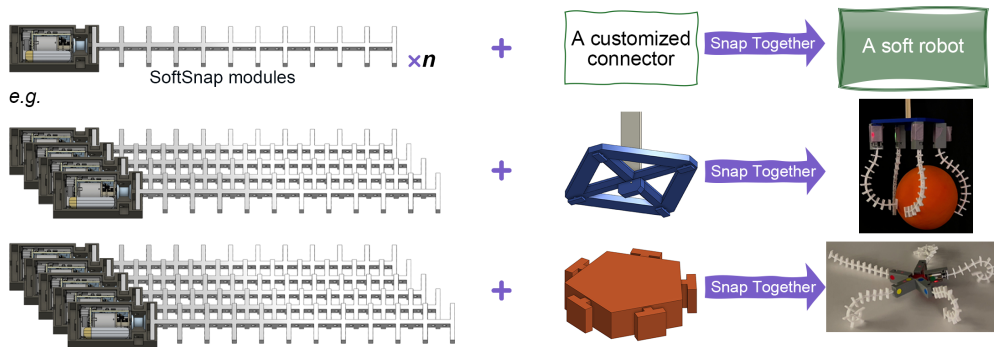


Fig. 1 Overview of the prototyping process. The first row demonstrates that combining n SoftSnap modules with a customized connector can construct a soft robot. The second row illustrates a gripper robot made from four SoftSnap modules arranged in a square and connected in the center. The third row depicts a brittle star-like robot created with five SoftSnap modules in a pentagonal configuration, also connected centrally.

Soft robots are adaptable and compliant [1], enabling the safe manipulation of delicate objects and navigation through challenging terrain, with applications in medical devices and search and rescue operations [2]. Integrating controllers, batteries, actuators, and sensors into untethered, compliant systems poses a challenge due to the complexity of the assembly process [3, 4]. Current actuation technologies for soft robots include pneumatic [5, 6], cable-driven [7, 8], electrically-responsive [9, 10], magnetically-responsive [11], and thermal-responsive systems (e.g., shape memory alloys) [12, 13]. Pneumatic actuation, for example, is widely used for its simplicity and smooth, fluid-like motion, but it often requires external pumps and tubing, limiting its practicality in untethered applications [10, 14–16]. The selection of the actuation method informs the design of a rapid-prototyping kit.

Systems like Legos and similar kits have enabled rapid prototyping of articulated rigid-body structures or robots [17]. Modular robots, leveraging similar systems, can adapt their physical structure to different tasks through reconfiguration [18–21]. Most modular robots rely on rigid components [22–24]. Recent research has begun integrating soft robotics principles into modular designs [25–28], aiming to develop reconfigurable soft robots that combine compliance and adaptability for diverse applications [29–31]. However, these approaches often involve intricate manual pre-assembly of modules, making it difficult to rapidly explore alternative geometries or to scale up to very large numbers of modules [32–34]. Simplifying the assembly process and designing soft modules that support rapid prototyping is, therefore, a promising area for research [35, 36].

This paper introduces a technique and components for quickly prototyping a certain class of soft robots with different geometries. The system is primarily intended for scientists with a range of backgrounds interested in exploring robotic analogs of

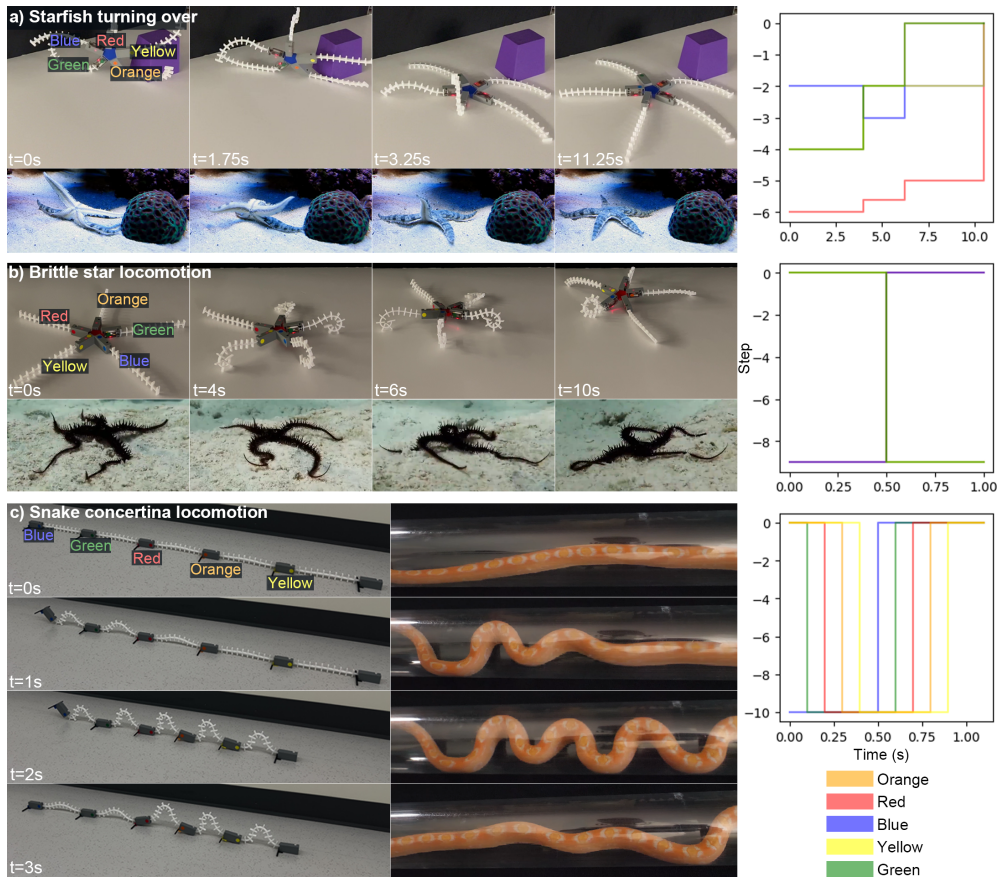


Fig. 2 Examples of the cable-driven SoftSnap modular system applied to bio-inspired robotic designs. The image illustrates three robotic motions inspired by nature. The right images show control signals for five modules, highlighted in blue, red, yellow, orange, and green. **(a)** The starfish-like robot replicates the turning-over behavior observed in real starfish during 10.5 s. At $t = 0$ s, the robot leans on a purple block shaped to resemble a starfish. By $t = 1.75$ s, the robot achieves a similar posture to a real starfish, with its top leg raised. It continues to receive control signals until it falls due to gravity at $t = 3.25$ s. The robot vividly emulates starfish motion until $t = 11.25$ s, after which the control is switched off, and the robot relaxes, resembling a starfish resting on the seafloor. **(b)** The brittle star-inspired robot demonstrates locomotion through periodic movements with a cycle of 1 s. Initially, the robot lies flat with its legs extended. A large step control signal is applied for 0.5 s, causing the legs to contract rapidly, followed by a second control signal for the remaining 0.5 s to release the actuation and contract the other legs. This stretch and shrink motion generates momentum, showing clear displacement from its starting position. **(c)** The snake-like robot performs concertina locomotion by sequentially deforming its modules. In this demonstration, the robot exhibits a periodic motion with a cycle of 1.1 s. It starts in a straight position, contracts its body, achieves full contraction, begins to unwind, and progressively returns to its initial straight state from head to tail. This is achieved by periodically alternating the step signals in different modules. The picture on the left shows that at $t=0$ s, the robot is in its initial state; at $t=1$ second, it begins to contract; at $t=2$ s, it is fully contracted; and by $t=3$ s, it is midway through returning to its original state. Credits for the three real animal images: *SeaAnimal4k* (Star fish 4k Amazing Starfish in Undersea Ultra Hd, <https://www.youtube.com/watch?v=MWIKfaxROd0>), *keepondiving3140* (Black brittle star Small Giftun Red Sea Egypt, <https://www.youtube.com/watch?v=cX9c43sF3vo>), and *SnakeDiscovery* (How Snakes Move! (They don't just slither!), <https://www.youtube.com/watch?v=7-AKPFiIEEw&t=30s>).

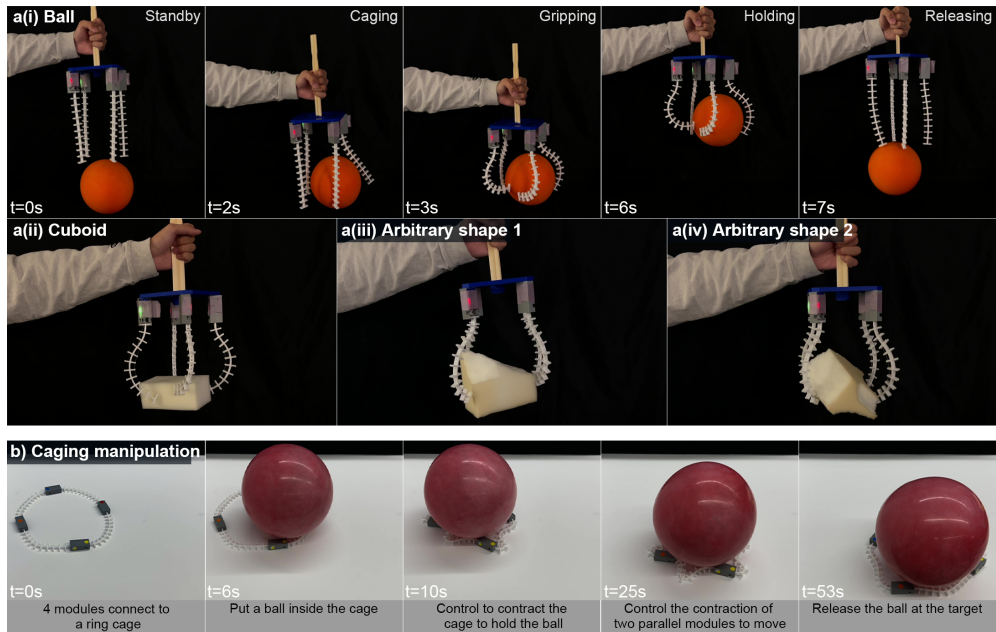


Fig. 3 Demonstrations of the cable-driven modular system applied to manipulation tasks. (a) The gripper robot effectively grasps objects of different shapes, such as a ball and a cuboid. In **a(i)**, we illustrate the five stages of ball grasping. In **a(ii)**, **a(iii)**, and **a(iv)**, we demonstrate grasping with three body shapes: one cuboid and two irregular forms, all controlled by the same signal for the four modules. (b) A four-module ring configuration is used for caging-type object transport, showcasing the ability to transport a large ball. At $t = 0$ s, the ring cage is in its initial state.

biological organisms, or soft robot geometries suitable for manipulation or locomotion tasks. The system enables easy component reuse and rapid testing of different ideas about mechanical design, control, and applications. We detail the design and initial implementation of cable-driven, modular soft actuator units that can be snapped together to quickly prototype untethered soft robotic systems. There are two main components: (1) actuable SoftSnap modules that each integrate a battery, motor, Wi-Fi communication, and control system, and (2) connectors that snap the modules together. Connectors can be easily 3D-printed to meet specific application requirements and can be re-used once built. An overview of the system is given in Fig. 1, and Fig. 2 shows some biologically-inspired example robots we constructed with the system. The SoftSnap modules, in some ways, serve a similar purpose to direct-driven joints in a traditional robot arm: the modules have one degree of freedom actuated by changing the cable length. Higher-degree-of-freedom serial or parallel structures for manipulation or locomotion can be formed by connecting these modules in various topologies using the connectors.

Each SoftSnap module incorporates a passive skeleton with holes through which the actuating cable is threaded. Using different holes leads to different *threading configurations*, each leading to a different deformation pattern for the module. Snapping modules together is trivially easy; the slowest steps of building a new design include

threading modules and possibly printing custom connectors. Once modules and connectors of sufficient variety have been constructed, they form a kit that allows very fast exploration of new soft robot structures.

Our current platform is meant to enable human designers to imagine, duplicate, or explore soft robot geometries. One difficulty in the process is that human designers may not have good intuition about what shapes a module with a particular threading pattern might achieve during actuation. We, therefore, present a quasi-static simulator that minimizes potential energy to predict the shapes achieved for different string lengths for a particular threading of the module. We also modified the simulator for use as a design tool. This tool allows users to input a desired curve, and then attempts to find the threading design that most closely matches the curve.

We demonstrate the versatility of the snap-together soft modular system through four distinct example assemblies: a gripper, a starfish robot, a snake robot, and a ring-shaped robot. These configurations showcase how the snap-together modules can be easily reassembled to create a range of functional soft robots, enabling untethered operation and supporting rapid prototyping, as shown in Figs. 2 and 3. In this paper, we detail the design, modeling, and experimental validation of these modular soft robots, highlighting their adaptability for diverse applications.

Results

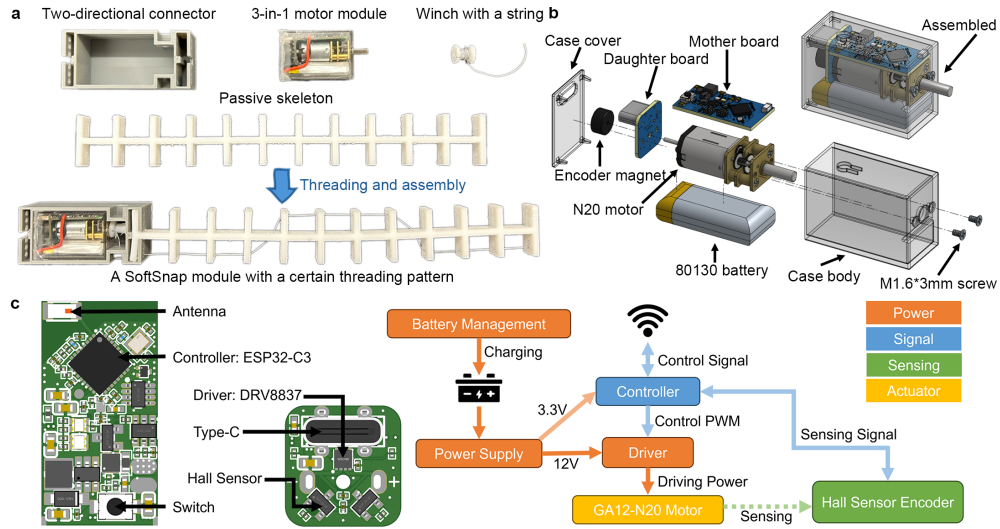


Fig. 4 SoftSnap module design and motor module details. (a) Three components of the actuable module: passive skeleton, 3-in-1 motor module, and two-directional connector. The passive skeleton is linked to the 3-in-1 motor module via a winch. A string extends from the motor, passes over the winch, threads through the holes on the passive skeleton, and continues to the far end of the passive skeleton. (b) The exploded-view drawing of the 3-in-1 motor module components which satisfied the three functionalities: Power, Control, and Actuation. (c) The PCB's mother board and daughter board are displayed on the left. The system function flow is present on the right.

Module and connector design

Each SoftSnap module houses all electronic components in a compact box connected to a passive skeleton. Each spine of the skeleton has holes that can be used to thread the single cable; different selections of holes lead to different threading patterns and different deformations when the cable is shortened. As shown in Fig. 4(a), the module includes: (i) Passive Skeleton with a uniformly designed structure featuring adjustable rib lengths, thickness, and middle rod dimensions for customized mechanical properties. (ii) 3-in-1 Motor Module (Fig. 4(b)) combines a motor, PCB (Fig. 4(c)), and battery in one compact unit for untethered operation and Wi-Fi control, including USB Type-C charging. (iii) Two-Directional Connector connects the passive skeleton to the motor module and allows for easy modular assembly via a snap-together mechanism. For example, as shown in Figs. 2 and 3, it supports the assembly of modules in configurations such as a chain (a snake-like robot) or a ring-shaped robot.

Multiple SoftSnap modules can combine with customized connectors to form various soft robots, as shown in Fig. 1. These connectors facilitate various configurations by linking different numbers of modules in diverse positions and orientations. For instance, as shown in Fig. 2(a), the starfish configuration employs a 3D pentagonal connector with small connectors to arrange five modules in different orientations. In the gripper configuration, shown in Fig. 3(a), a cuboid connector arranges four modules symmetrically.

Quasi-static modeling and simulation

To understand how different threading patterns influence the deformation and configuration space, we developed a quasi-static model that approximates the deformation behavior for a given threading pattern.

Forward quasi-static modeling: Given a threading pattern T and target string length L_{target} , we compute the resulting deformation shape of the skeleton. We can also explore the complete set of deformations for a module by iteratively reducing the string length, using the geometry from the previous iteration as the initial guess for the next step.

The skeleton is modeled as a collection of segments, with each pair of adjacent ribs storing potential energy to resist bending. At each iteration, a trust-region-constrained optimizer minimizes the total potential energy, subject to the string length constraint. The potential energy is modeled as the sum of squared bending angles α_i between adjacent ribs. Given a threading pattern T_i for the i -th pair of ribs, the string length L_i is calculated as a function of the bending angle α_i , and the target total string length L_{target} must satisfy $L_{\text{target}} < L_{\text{original}}$. The optimization problem is formulated as:

$$\min \sum_i \alpha_i^2 \quad \text{s.t.} \quad \sum_i L_i = \sum_i L(\alpha_i, T_i) = L_{\text{target}}. \quad (1)$$

Although L_i can be directly calculated from α_i , the reverse is more complex due to the non-linear relationship. To find α_i given L_{target} , numerical methods like Newton-Raphson can be employed. The relationship between L_i and α_i is detailed in the Supplementary ‘‘String Length Calculation’’ section.

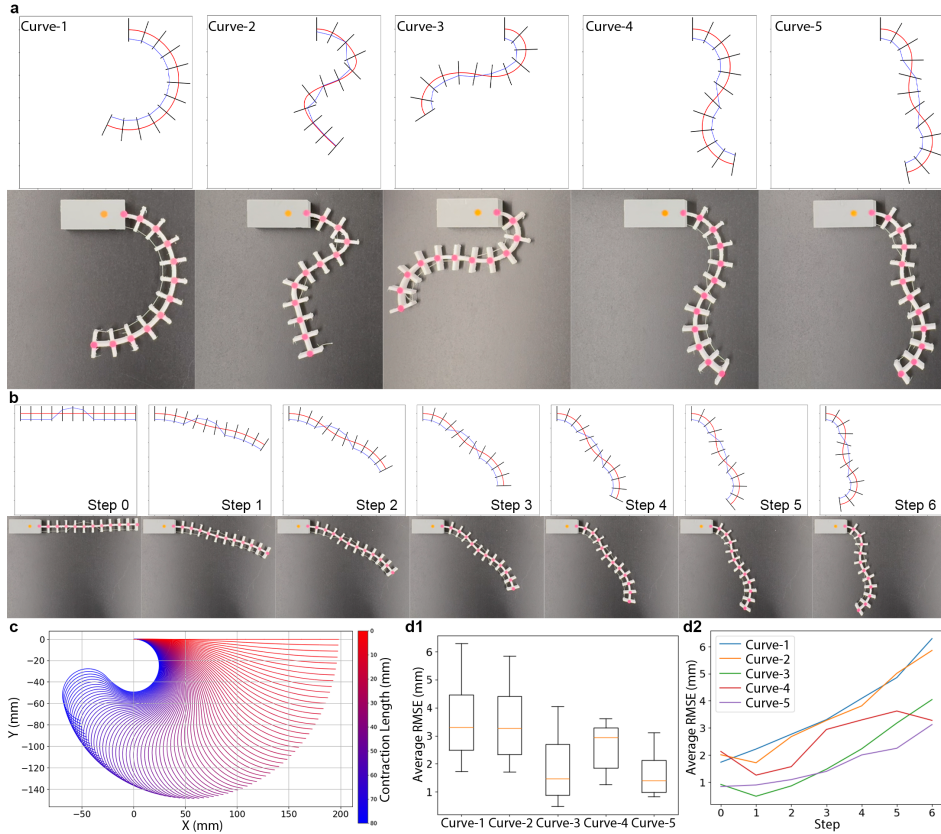


Fig. 5 Five different threading methods and their resulting deformations after shortening the strings to the same length in both simulations and real experiments. **(a)** The left end of the skeleton, attached to the body case, is fixed to the ground. The blue line illustrates the single string that runs from the motor to the skeleton and reaches its far end. The red line represents the deformed shape of the skeleton along its longitudinal axis. **(b)** Example of a “w” (curve-5) shape configuration, comparing seven steps of shortening the strings evenly in experiments and simulations. **(c)** Displays all the configurations with the ‘Curve-3’ threading method and shows the workspace of the middle of the skeleton from 0 mm to 80 mm string contract length; the contraction process is associated with the color bar. **(d1)** and **(d2)** The average RMSE is calculated with the RMSE of the 12 center points (marked in pink) of each rod on the skeleton. **(d1)** Illustrates the average RMSE across five experiment threading patterns across seven testing steps. Within the experiment range, the real robot shows a good match with the simulation. **(d2)** Shows the RMSE of five different experiment threading patterns over steps. The average RMSE, with a maximum of about 6 mm, is relatively small compared to the skeleton’s length of 201 mm. The error has an increment trend with the string length contraction.

Thus, the optimization procedure iteratively updates α_i , ensuring the total string length matches L_{target} while minimizing potential energy to find the optimal bending configuration. The complete procedure is outlined in Algorithm 1.

Inverse quasi-static modeling: Given a target deformation shape, a human designer may wish to determine the threading pattern and corresponding string length. We

simplify the input by representing the desired deformation as a sequence of target bending angles $(\alpha_0, \alpha_1, \dots, \alpha_n)$. Using these angles, the positions of all intersection points between the ribs and the middle rods are explicitly calculated based on geometric relations. Suppose each rib has m holes, labeled as t_1, t_2, \dots, t_m . For each pair of adjacent ribs (each segment), there are m^2 possible threading patterns. For a specific combination of threading, such as t_i and t_j , we calculate the corresponding string length L_{ij} based on the required bending angle. In addition to the threading pattern, we also store the string length for each segment. If L_{ij} exceeds the original string length (the length between the holes when the angle is zero), that threading pattern is discarded. This process yields a set of valid threading options for each pair of ribs.

Next, we applied a backtracking method to find all valid threading paths from the first rib to the final rib. This method ensures that each threading path is consistent with the geometrical constraints of adjacent ribs. During backtracking, we also calculate the total string length for each threading path by summing the string lengths of all segments. Once all valid paths are generated, we iterate through them, using the threading pattern and the total string length of each path as inputs. For each path, we apply the forward quasi-static model to compute the resulting bending angles. We then calculate the distance between these resulting angles (in radians) and the target angles using the L2 norm. The path that produces the minimum distance is selected as the optimal solution. The algorithm outputs the corresponding threading pattern and the total string length of the chosen path.

There are several key points to note. First, the number of holes on each rib significantly affects computational cost; for example, with 3 holes per rib, the total number of possible threading patterns for n ribs is 3^n . Although the filtering method reduces the number of valid threadings, multiple configurations may still achieve the same target angle, leaving many possible solutions. To address this in practice, parallel computing can be employed, or a maximum number of threading options can be set, with the best solution chosen from this subset. Figure S4 (a) and (b) demonstrate that different threading patterns and string lengths can produce the same desired angle sequence. Second, some angles may be unachievable due to physical constraints of the threading hole positions, and the inverse algorithm finds the closest solution given the number of holes on each rib, though it may not always perfectly match the desired angles. Figures S4 (c) and (d) illustrate examples where 8 holes per rib were used, and the best solution was found within these threading constraints.

Comparison between simulation and experiments: To validate the skeleton design and compare the robot’s physical behavior with simulations, we employed five different threading methods, as shown in Fig. 5. Color-coded markers (pink and orange) were placed at the intersections of ribs and middle rods. This tracked the skeleton’s deformation for comparison with simulated results. We captured top-down images of the skeleton using a calibrated camera to ensure accurate spatial measurements. Standard computer vision techniques, such as color thresholding to isolate the pink and orange markers, contour detection to identify marker boundaries, and image moments to calculate marker centroids, were employed. These centroids provided the x and y coordinates of the markers. The line connecting the pink and orange markers

defined the x -axis, and the image was rotated to align the skeleton with the horizontal axis for consistent analysis.

After aligning the skeleton, we measured the adjusted marker positions relative to the first pink marker, which is closest to the orange marker and is designated as the origin. These measurements allowed us to evaluate how various threading methods influenced the skeleton’s deformation. The experimental results were then compared with the simulation data to assess the model’s accuracy and examine how threading variations and string lengths affected the robot’s performance. Moreover, we also illustrate the five different threading methods and the resulting deformations when the strings are shortened to equal lengths, both in simulation and real experiments.

Case studies and experimental validation

In this section, we present case studies and experimental results.

Starfish and brittle star robots

To create a starfish-like robot, we designed a 3D pentagonal connector with small pegs for securely attaching the connectors of the actuatable modules. Two connector configurations were developed: one (red connector) positions all modules with their deformation direction perpendicular to the connector, while the other (blue connector) aligns some modules perpendicular and others parallel to the connector. Fig. 2(a) and Movie S1 illustrate how the robot mimics the righting response of a real starfish by deforming its five limbs. In contrast, Fig. 2(b) and Movie S1 demonstrate locomotion that closely resembles the movement of brittle stars, achieved by coordinating the deformation of its limbs in sequences. The control sequence for each skeleton is illustrated in Fig. 2. Both robots use the same threading patterns for the five skeletons, with details of the threading pattern provided in Figure S3.

Snake-like robot

The snake-like robot was constructed by connecting a chain of five skeleton modules, each threaded in a consistent pattern to form an “ ω ” shape (Figure S3). To enhance ground stability, six short rods were attached to the connectors. Figure 2(c) and Movie S1 showed an example. Forward movement was achieved by sequentially deforming the modules from front to back, then restoring them to their original shape, mimicking the concertina locomotion observed in real snakes, especially in confined spaces or rough terrain.

Gripper

The gripper was developed by attaching four actuatable modules to a rectangular connector arranged in an “X” configuration, as shown in Figure 3(a) and Movie S1. By adjusting the connector design and threading pattern, different types of grippers can be produced. In this example, the modules were threaded to form an “S” shape (Figure S3), with a larger lower curve and a smaller upper curve. This setup allows the gripper to grasp objects of various shapes, such as balls, cuboids, and irregular objects.

However, the current design is only suitable for handling lightweight items. Handling heavier objects would require a stiffer skeleton or perhaps a different threading configuration.

Caging-type object transport

A caging-type object transport system was developed by connecting four modules into a ring configuration, as shown in Figure 3(b) and Movie S1. The threading pattern used is the same as that for the snake-like robot. This system was used to transport a lightweight ball, using sequential deformation of the modules. Initially, two modules were actuated to deform and grip the ball, after which the remaining two modules returned to their original shape and were then deformed again to continue the transport. This alternating deformation allowed for continuous object movement. While the current system successfully demonstrated the transport of lightweight objects, such as the tested ball, transporting heavier objects would require further modification, such as increasing the stiffness of the skeleton.

Discussion

Soft robotics systems often encounter challenges in rapid prototyping and achieving untethered, actuatable modules adaptable to various applications. Our research aims to address this by introducing a cable-driven modular system that facilitates the quick and scalable assembly of soft robots using snap-together components. We have integrated key electronic components—such as communication hardware, a battery, and a motor—into a compact box that combines with a flexible skeleton to form an untethered, actuated, and bendable module. These modules can be easily snapped together to create different soft robotic configurations, such as starfish-like, snake-like, and gripper robots. This approach helps to reduce the complexity of soft robot manufacturing, allowing researchers to focus more on design exploration and application functionality.

A key insight from our research is the role of threading patterns in shaping the skeleton’s deformation. We developed a model demonstrating how different threading configurations affect the overall shape and flexibility of the skeleton. By simply adjusting the threading, users can achieve various deformations, which can then be used to prototype distinct robot forms. This capability allows researchers to focus more on control strategies and specific applications rather than the intricacies of the construction process.

Despite these advancements, several limitations remain. The current work focused primarily on the effects of threading on deformation patterns, while other critical factors—such as material properties of the skeleton, rib dimensions, and mechanical load capacities—were not fully explored. Additionally, the skeleton design operates mainly in 2D, limiting the complexity and range of motions. Expanding the system into a full 3D configuration and incorporating multiple strings per module could provide greater control and precision, broadening the scope of possible applications. Moreover, advanced control algorithms, including real-time feedback mechanisms and adaptive behaviors, should be explored to improve the system’s responsiveness to environmental changes.

Our research provides a starting point for a more accessible and scalable method of soft robot development. By simplifying the construction process, researchers can now focus on exploring new forms, behaviors, and control strategies for their robots. We anticipate that SoftSnap modules will enable rapid prototyping for a range of applications, from biomimetic locomotion to manipulation tasks, while also lowering the barrier to entry for soft robotic research. Future work will focus on enhancing the design with multi-string actuation for finer control, as well as conducting field tests to assess the system’s robustness and scalability in real-world environments.

Methods

SnapSoft module and connector fabrication

A SnapSoft module is composed of three main components. 1) The passive skeleton is 3D-printed using TPU filament (1.75 mm, Amazon Basics), selected for its flexibility, durability, and ability to deform under applied loads. 2) The 3-in-1 motor module integrates a motor, a custom-designed printed circuit board (PCB), and an 80130 lithium-ion battery (3.7 V, 200 mAh, 30C). The battery’s high energy density allows for satisfactory operation while minimizing bulk. The PCB, as shown in Figure 4(c), incorporates the controller and driver and soldered two parts perpendicular to minimize bulk. The motor, type GA12-N20 (50,000 RPM) with a gear ratio of 1:100, was selected for its balance between torque and speed. All of these components are housed in a compact casing, 3D-printed using resin material, ensuring efficient integration with the skeleton. 3) The two-directional connectors, which are 3D-printed using PLA filament (1.75 mm, Overture), enable modular reconfiguration of the skeleton. These connectors allow for the formation of different structures, depending on the application. The snap-together mechanism has been tested for durability, ensuring reliable connections between modules and the connector structure.

The customized connectors are fabricated from polylactic acid (PLA) filament (1.75 mm, Overture), chosen for its rigidity and compatibility with other components.

Data availability

All data needed to evaluate the conclusions in the paper are present in the paper or the Supplementary Materials. Computer code is available from the first author and corresponding authors upon request.

References

- [1] Rus, D., Tolley, M.: Design, fabrication and control of soft robots. *Nature* **521**, 467–75 (2015) <https://doi.org/10.1038/nature14543>
- [2] Dong, X., Luo, X., Zhao, H., Qiao, C., Li, J., Yi, J., Yang, L., Oropeza, F.J., Hu, T.S., Xu, Q., Zeng, H.: Recent advances in biomimetic soft robotics: fabrication approaches, driven strategies and applications. *Soft Matter* **18**, 7699–7734 (2022) <https://doi.org/10.1039/D2SM01067D>

- [3] Lee, C., Kim, M., Kim, Y.J., Hong, N., Ryu, S., Kim, H.J., Kim, S.: Soft robot review. *International Journal of Control, Automation and Systems* **15**(1), 3–15 (2017) <https://doi.org/10.1007/s12555-016-0462-3>
- [4] Lipson, H.: Challenges and opportunities for design, simulation, and fabrication of soft robots. *Soft Robotics* **1**, 21–27 (2014) <https://doi.org/10.1089/soro.2013.0007>
- [5] Kobayashi, R., Nabae, H., Endo, G., Suzumori, K.: Soft tensegrity robot driven by thin artificial muscles for the exploration of unknown spatial configurations. *IEEE Robotics and Automation Letters* **7**(2), 5349–5356 (2022) <https://doi.org/10.1109/LRA.2022.3153700>
- [6] Baines, R., Patiballa, S., Booth, J., Ramirez, L., Sipple, T., Garcia, A., Fish, F., Kramer-Bottiglio, R.: Multi-environment robotic transitions through adaptive morphogenesis. *Nature* **610**, 283–289 (2022) <https://doi.org/10.1038/s41586-022-05188-w>
- [7] Chen, M., Fraddosio, A., Micheletti, A., Pavone, G., Piccioni, M.D., Skelton, R.E.: Analysis of clustered cable-actuation strategies of v-expander tensegrity structures. *Engineering Structures* **296**, 116868 (2023) <https://doi.org/10.1016/j.engstruct.2023.116868>
- [8] Lai, J., Lu, B., Zhao, Q., Chu, H.K.: Constrained motion planning of a cable-driven soft robot with compressible curvature modeling. *IEEE Robotics and Automation Letters* **7**(2), 4813–4820 (2022) <https://doi.org/10.1109/LRA.2022.3152318>
- [9] Mao, Z., Peng, Y., Hu, C., Ding, R., Yamada, Y., Maeda, S.: Soft computing-based predictive modeling of flexible electrohydrodynamic pumps. *Biomimetic Intelligence and Robotics* **3**(3), 100114 (2023) <https://doi.org/10.1016/j.birob.2023.100114>
- [10] Miriyev, A., Stack, K., Lipson, H.: Soft material for soft actuators. *Nature Communications* **8** (2017) <https://doi.org/10.1038/s41467-017-00685-3>
- [11] Chung, H.-J., Parsons, A.M., Zheng, L.: Magnetically controlled soft robotics utilizing elastomers and gels in actuation: A review. *Advanced Intelligent Systems* **3**(3), 2000186 (2021) <https://doi.org/10.1002/aisy.202000186> <https://onlinelibrary.wiley.com/doi/pdf/10.1002/aisy.202000186>
- [12] Zhao, L., Wu, Y., Yan, W., Zhan, W., Huang, X., Booth, J., Mehta, A., Bekris, K., Kramer-Bottiglio, R., Balkcom, D.: Starblocks: Soft actuated self-connecting blocks for building deformable lattice structures. *IEEE Robotics and Automation Letters* **8**(8), 4521–4528 (2023) <https://doi.org/10.1109/LRA.2023.3284361>

- [13] Zhao, L., Wu, Y., Blanchet, J., Perroni-Scharf, M., Huang, X., Booth, J., Kramer-Bottiglio, R., Balkcom, D.: Soft lattice modules that behave independently and collectively. *IEEE Robotics and Automation Letters* **7**(3), 5942–5949 (2022) <https://doi.org/10.1109/LRA.2022.3160611>
- [14] Xavier, M.S., Tawk, C.D., Zolfagharian, A., Pinskiar, J., Howard, D., Young, T., Lai, J., Harrison, S.M., Yong, Y.K., Bodaghi, M., Fleming, A.J.: Soft pneumatic actuators: A review of design, fabrication, modeling, sensing, control and applications. *IEEE Access* **10**, 59442–59485 (2022) <https://doi.org/10.1109/ACCESS.2022.3179589>
- [15] Zhai, Y., Boer, A.D., Yan, J., Shih, B., Faber, M., Speros, J., Gupta, R., Tolley, M.T.: Desktop fabrication of monolithic soft robotic devices with embedded fluidic control circuits. *Science Robotics* **8**(79), 3792 (2023) <https://doi.org/10.1126/scirobotics.adg3792>
- [16] Tawk, C., Alici, G.: A review of 3d-printable soft pneumatic actuators and sensors: Research challenges and opportunities. *Advanced Intelligent Systems* **3** (2021) <https://doi.org/10.1002/aisy.202000223>
- [17] Souza, I.M.L., Andrade, W.L., Sampaio, L.M.R., Araujo, A.L.S.O.: A systematic review on the use of lego robotics in education. In: 2018 IEEE Frontiers in Education Conference (FIE), pp. 1–9 (2018). <https://doi.org/10.1109/FIE.2018.8658751>
- [18] Liang, G., Wu, D., Tu, Y., Lam, T.L.: Decoding modular reconfigurable robots: A survey on mechanisms and design. *The International Journal of Robotics Research* (2024) <https://doi.org/10.1177/02783649241283847>
- [19] Yim, M., Zhang, Y., Duff, D.: Modular robots. *IEEE Spectrum* **39**(2), 30–34 (2002) <https://doi.org/10.1109/6.981854>
- [20] Yim, M., White, P., Park, M., Sastra, J.: *Modular Self-Reconfigurable Robots*, pp. 5618–5631. Springer, New York, NY (2009). https://doi.org/10.1007/978-0-387-30440-3_334
- [21] Yim, M., Shen, W.-m., Salemi, B., Rus, D., Moll, M., Lipson, H., Klavins, E., Chirikjian, G.S.: Modular self-reconfigurable robot systems [grand challenges of robotics]. *IEEE Robotics Automation Magazine* **14**(1), 43–52 (2007) <https://doi.org/10.1109/MRA.2007.339623>
- [22] Liu, C., Whitzer, M., Yim, M.: A distributed reconfiguration planning algorithm for modular robots. *IEEE Robotics and Automation Letters* **4**(4), 4231–4238 (2019) <https://doi.org/10.1109/LRA.2019.2930432>
- [23] Wei, H., Cai, Y., Li, H., Li, D., Wang, T.: Sambot: A self-assembly modular robot for swarm robot. In: 2010 IEEE International Conference on Robotics and

- Automation, pp. 66–71 (2010). <https://doi.org/10.1109/ROBOT.2010.5509214>
- [24] Ozkan-Aydin, Y., Goldman, D.I.: Self-reconfigurable multilegged robot swarms collectively accomplish challenging terradynamic tasks. *Science Robotics* **6**(56), 1628 (2021) <https://doi.org/10.1126/scirobotics.abf1628>
- [25] Freeman, C., Maynard, M., Vikas, V.: Topology and morphology design of spherically reconfigurable homogeneous modular soft robots. *Soft Robotics* **10**(1), 52–65 (2023) <https://doi.org/10.1089/soro.2021.0125>
- [26] Li, S., Rus, D.: Jellocube: A continuously jumping robot with soft body. *IEEE/ASME Transactions on Mechatronics* **24**(2), 447–458 (2019) <https://doi.org/10.1109/TMECH.2019.2899606>
- [27] Zou, J., Lin, Y., Ji, C., Yang, H.: A reconfigurable omnidirectional soft robot based on caterpillar locomotion. *Soft Robotics* **5**(2), 164–174 (2018) <https://doi.org/10.1089/soro.2017.0008>
- [28] Suzuki, R., Zheng, C., Kakehi, Y., Yeh, T., Do, E.Y.-L., Gross, M.D., Leithinger, D.: Shapebots: Shape-changing swarm robots. In: *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. UIST '19, pp. 493–505. Association for Computing Machinery, New York, NY, USA (2019). <https://doi.org/10.1145/3332165.3347911>
- [29] Ceron, S., Kimmel, M.A., Nilles, A., Petersen, K.: Soft robotic oscillators with strain-based coordination. *IEEE Robotics and Automation Letters* **6**(4), 7557–7563 (2021) <https://doi.org/10.1109/LRA.2021.3100599>
- [30] Malley, M., Haghghat, B., Houel, L., Nagpal, R.: Eciton robotica: Design and algorithms for an adaptive self-assembling soft robot collective. In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4565–4571 (2020). <https://doi.org/10.1109/ICRA40945.2020.9196565>
- [31] Li, S., Awale, S.A., Bacher, K.E., Buchner, T.J., Della Santina, C., Wood, R.J., Rus, D.: Scaling up soft robotics: A meter-scale, modular, and reconfigurable soft robotic system. *Soft Robotics* **9**(2), 324–336 (2022) <https://doi.org/10.1089/soro.2020.0123>
- [32] Zhang, C., Zhu, P., Lin, Y., Jiao, Z., Zou, J.: Modular soft robotics: Modular units, connection mechanisms, and applications. *Advanced Intelligent Systems* **2** (2020) <https://doi.org/10.1002/aisy.201900166>
- [33] Lee, H.C., Elder, N., Leal, M., Stantial, S., Martinez, E.V., Jos, S., Cho, H., Russo, S.: A fabrication strategy for millimeter-scale, self-sensing soft-rigid hybrid robots. *Nature Communications* **15**(1) (2024) <https://doi.org/10.1038/s41467-024-51137-8>

- [34] Usevitch, N.S., Hammond, Z.M., Schwager, M., Okamura, A.M., Hawkes, E.W., Follmer, S.: An untethered isoperimetric soft robot. *Science Robotics* **5**(40), 0492 (2020) <https://doi.org/10.1126/scirobotics.aaz0492>
- [35] Schmitt, F., Piccin, O., Barbé, L., Bayle, B.: Soft robots manufacturing: A review. *Frontiers in Robotics and AI* **5** (2018) <https://doi.org/10.3389/frobt.2018.00084>
- [36] Wehner, M., Truby, R., Fitzgerald, D., Mosadegh, B., Whitesides, G., Lewis, J., Wood, R.: An integrated design and fabrication strategy for entirely soft, autonomous robots. *Nature* **536**, 451–455 (2016) <https://doi.org/10.1038/nature19100>

Acknowledgments: We thank the creators of *SeaAnimal4k* (Star fish 4k Amazing Starfish in Undersea Ultra Hd, <https://www.youtube.com/watch?v=MWIKfaxROd0>), *keepondiving3140* (Black brittle star Small Giftun Red Sea Egypt, <https://www.youtube.com/watch?v=cX9c43sF3vo>), and *SnakeDiscovery* (How Snakes Move! (They don't just slither!), <https://www.youtube.com/watch?v=7-AKPFiEEw&t=30s>) for their contribution to providing high-quality references for the locomotion of different creatures: provided videos of starfish, brittle star, and snake locomotion.

Author contributions: Conceptualization: LZ, DB, MC, CS, YJ; Methodology: LZ, DB, MC, CS, YJ; Investigation: LZ, DB, MC, CS, YJ; Visualization: LZ, YJ; Experiments Design and Implementation: LZ, DB, MC, CS, YJ; System Design and Implementation: LZ, DB, MC, CS, YJ; Algorithm Design and Implementation: LZ, DB, MC, YJ, CS; Project administration: DB, LZ; Supervision: DB, MC; Writing: original draft: LZ; Writing: review & editing: LZ, DB, MC, CS, YJ.

Competing interests: The authors declare they have no competing interests.

Supplementary materials

This PDF file includes:

Supplementary methods

Figures S1 to S3

Algorithm 1

Other supplementary material

Movie S1

Supplementary methods

String length calculation

The string length L between two ribs is parameterized by the bending angle α and the threading locations on both ribs. The distances from the threading points on ribs 1 and 2 to the rib's endpoint (which is closest to the center (x_0, y_0)) are defined as l_1 and l_2 , respectively. The corresponding threading positions are represented by the one-dimensional scalars λ_1 and λ_2 , which are calculated by subtracting half the rib's length from l_1 and l_2 . Specifically, $\lambda_1 = l_1 - L_{\text{rib}}/2$ and $\lambda_2 = l_2 - L_{\text{rib}}/2$, an example is shown in Figure S1.

Case 1: Straight line ($\alpha \approx 0$). When α is small, the beam connecting the ribs remains nearly straight. In this case, the string length L is the Euclidean distance between the threading points S_1 and S_2 , calculated as:

$$S_1 = (x_1 + \lambda_1 \cos \theta_1, y_1 + \lambda_1 \sin \theta_1), \quad (2)$$

$$S_2 = (x_1 + \lambda_2 \cos(\theta_1) - s \sin(\theta_1), y_1 + \lambda_2 \sin(\theta_1) + s \cos(\theta_1)). \quad (3)$$

Case 2: Curved arc ($\alpha \neq 0$). For non-zero α , the ribs form a circular arc. The radius R of the arc is determined by the relation: $R = s/|\alpha|$. The center of the arc is located at: $x_0 = x_1 - R \cos \theta_1$ and $y_0 = y_1 - R \sin \theta_1$. The endpoint of the arc (x_2, y_2) is positioned at an angle $\theta_1 + \alpha$, and is calculated as: $x_2 = x_0 + R \cos(\theta_1 + \alpha)$ and $y_2 = y_0 + R \sin(\theta_1 + \alpha)$. The threading points S_1 and S_2 on the arc are determined by:

$$S_1 = (x_1 + \lambda_1 \frac{x_1 - x_0}{R}, y_1 + \lambda_1 \frac{y_1 - y_0}{R}), \quad (4)$$

$$S_2 = (x_2 + \lambda_2 \frac{x_2 - x_0}{R}, y_2 + \lambda_2 \frac{y_2 - y_0}{R}). \quad (5)$$

Please note that there are two cases because when $\alpha \approx 0$, the values $x_1 - x_0$, $x_2 - x_0$, and R will all diverge to infinity, causing numerical instability due to floating-point precision limits. To address this, a blending method is used to achieve greater numerical stability. When the radius of the arc becomes very large, the arc behaves more like a straight line, and the trigonometric calculations based on the arc center lose accuracy. Thus, for those cases, we switched to a linear approximation based on the arc's starting point and length, where the coordinates of x_2 and y_2 are given by $x_2 = x_1 + s \cos(\theta_1 + \alpha + \frac{\pi}{2})$ and $y_2 = y_1 + s \sin(\theta_1 + \alpha + \frac{\pi}{2})$. This method ensures that even for large radii, the endpoints of the arc are computed efficiently and without the

risk of crossing or overlap. For both cases, the string length L is given by:

$$L = \|S_1 - S_2\|. \quad (6)$$

Algorithm 1 Quasi-static model.

Initialization: $\epsilon = 1 \times 10^{-6}$
while $|L_{i,k} - L_{i,k-1}| \leq \epsilon$ **do**
 Optimized L_i^* from:

$$\begin{cases} \min \sum \alpha_i^2 = \text{Length2Angle}(L_i, T_i) \\ \text{s.t. } \sum L_i = L_{total} \end{cases}$$

 $k \leftarrow k + 1$
end while
function $\alpha = \text{Length2Angle}(L_i, T_i)$
 $\alpha \leftarrow \emptyset$
 for $i = 0 : n$ **do**
 Optimized α_i from:

$$\begin{cases} \min_{\alpha_i} \|L(\alpha_i) - L_i\| \\ \text{s.t. } -\pi < \alpha_i \leq \pi \end{cases}$$

 $\alpha \leftarrow \alpha \cup \{\alpha_i\}$
 end for
end function

Supplementary Figures

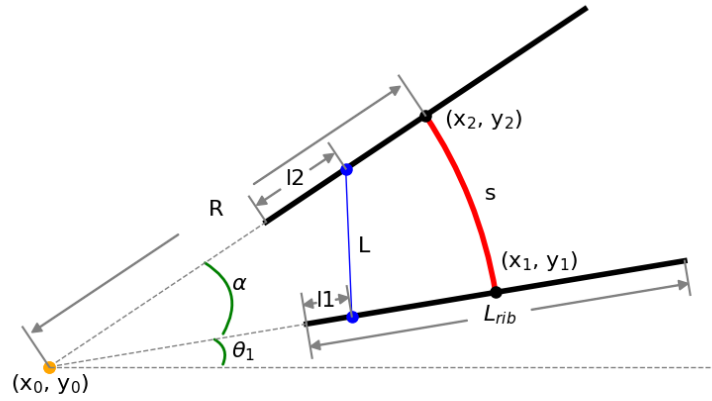


Figure S1: Example showing the deformation behavior of a segment of the skeleton. The thick black lines represent the ribs of the passive skeleton, with points (x_1, y_1) and (x_2, y_2) located at the center of the bottom and top ribs, respectively. The length of the rib is L_{rib} . The thick red line signifies the curved segments between the ribs along the skeleton's longitudinal direction, having an arc length s , which represents the spine of the skeleton. The radius of the spine-bending curve is denoted by R . The thin blue line indicates the string connecting the threading points, with length L . The angle θ_1 describes the orientation of the bottom rib relative to the horizontal, and α is the angle between the two ribs. The distances from the threading points on the bottom rib and top rib to each rib's endpoint, which is closest to the center (x_0, y_0) , are defined as l_1 and l_2 , respectively.

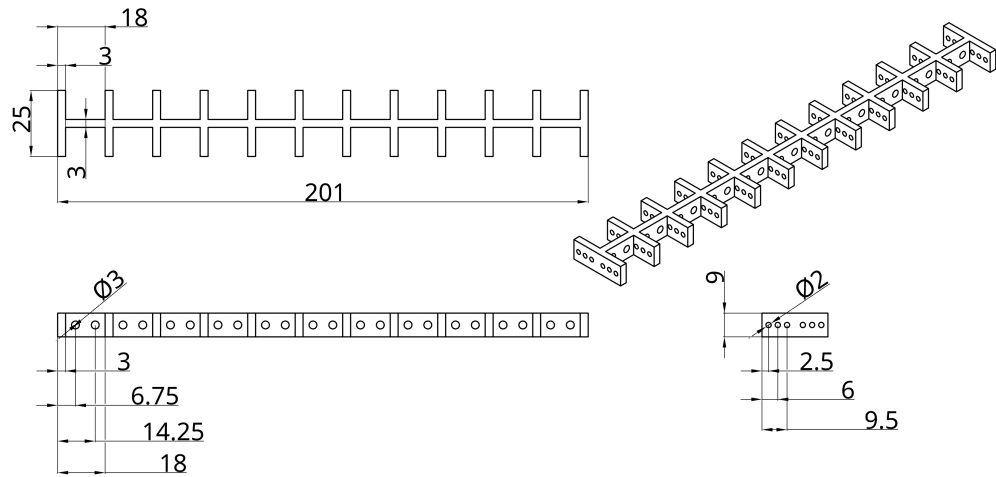


Figure S2: Specifications of the 3D-printed skeleton. The front, side, top, and axonometric drawings with dimensions are displayed, showing a total length of 201 mm, width of 25 mm, and thickness of 9 mm. Each vertical rib features six threading holes, with an additional two holes located between the ribs. This arrangement of holes provides design flexibility, allowing for the use of a single string to thread through the holes in various configurations. This enables the creation of diverse shapes with Soft-Snap modules. By designing customized connectors, these modules can be assembled into innovative, mass-produced soft robots.

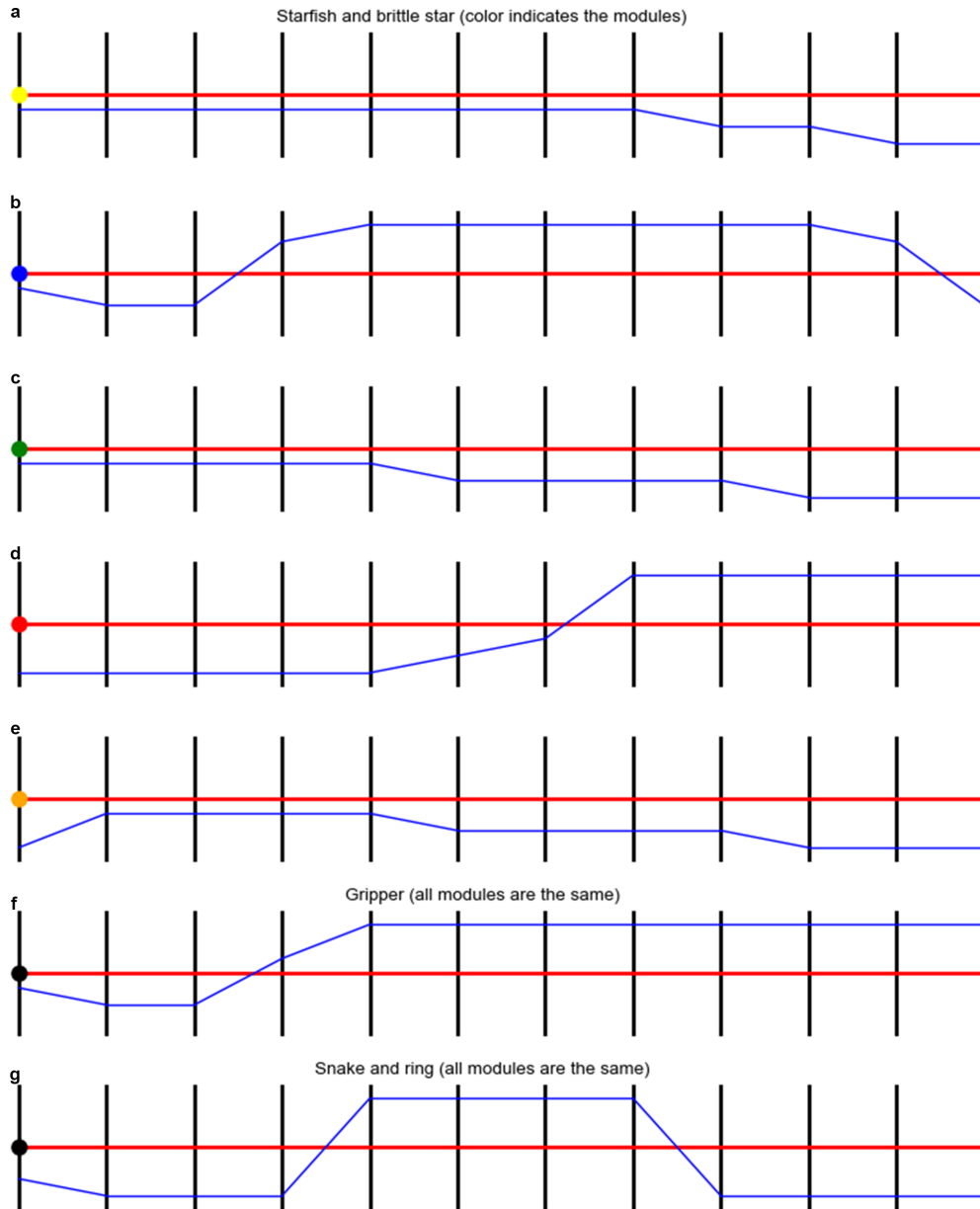


Figure S3: Threading details for all the robots in this paper. (a)-(e) the five SoftSnap modules in yellow, blue, green, red, and orange used in the Starfish and Brittle Star robots, (f) the four SoftSnap modules in the Gripper robot, and (g) the four SoftSnap modules in the Snake and Ring robots. The black and red lines depict the passive skeleton, and the blue lines are strings running through holes from the left to the right end of the skeleton.

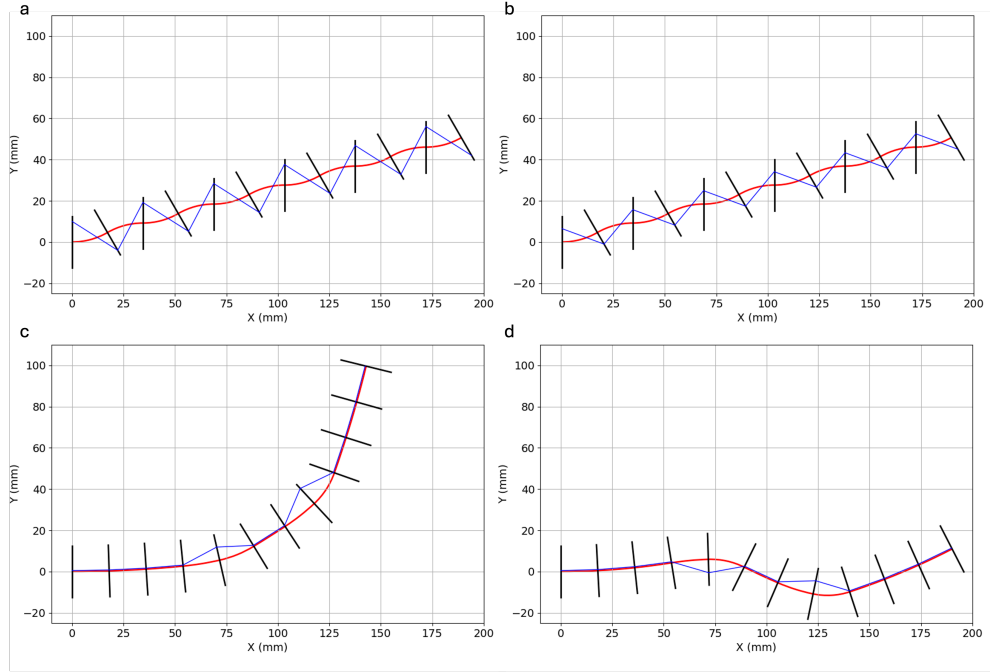


Figure S4: Calculated threading pattern and string lengths given the desired angles. (a) Desired angles: $[30^\circ, -30^\circ, 30^\circ, -30^\circ, 30^\circ, -30^\circ, 30^\circ, -30^\circ, 30^\circ, -30^\circ, 30^\circ]$, obtained angles: $[30^\circ, -30^\circ, 30^\circ, -30^\circ, 30^\circ, -30^\circ, 30^\circ, -30^\circ, 30^\circ, -30^\circ, 30^\circ]$, threading patterns: $[-10, 10, -10, 10, -10, 10, -10, 10, -10, 10, -10, 10]$ mm, string lengths: 270 mm. (b) Desired angles: $[30^\circ, -30^\circ, 30^\circ, -30^\circ, 30^\circ, -30^\circ, 30^\circ, -30^\circ, 30^\circ, -30^\circ, 30^\circ]$, obtained angles: $[30^\circ, -30^\circ, 30^\circ, -30^\circ, 30^\circ, -30^\circ, 30^\circ, -30^\circ, 30^\circ, -30^\circ, 30^\circ]$, threading patterns: $[-6.5, 6.5, -6.5, 6.5, -6.5, 6.5, -6.5, 6.5, -6.5, 6.5, -6.5, 6.5]$ mm, string lengths: 224 mm. (c) Desired angles: $[0^\circ, 0^\circ, 0^\circ, 10^\circ, 15^\circ, 0^\circ, 10^\circ, 30^\circ, 0^\circ, 0^\circ, 0^\circ]$, obtained angles: $[1.9^\circ, 1.9^\circ, 1.9^\circ, 7.2^\circ, 18.6^\circ, 1.9^\circ, 9.7^\circ, 27.7^\circ, 1.9^\circ, 1.9^\circ, 1.9^\circ]$, threading patterns: $[-0.5, -0.5, -0.5, -0.5, -6.5, -0.5, -0.5, -10, -0.5, -0.5, -0.5, -0.5]$ mm, string lengths: 199 mm. (d) Desired angles: $[0^\circ, 0^\circ, 0^\circ, -10^\circ, -30^\circ, 0^\circ, 10^\circ, 30^\circ, 0^\circ, 0^\circ, 0^\circ]$, obtained angles: $[3^\circ, 3^\circ, 3^\circ, -7.2^\circ, -28.1^\circ, 3^\circ, 11.6^\circ, 29.9^\circ, 3^\circ, 3^\circ, 3^\circ]$, threading patterns: $[-0.5, -0.5, -0.5, -0.5, 6.5, -0.5, -0.5, -6.5, -0.5, -0.5, -0.5, -0.5]$ mm, string lengths: 196 mm.