MuA-Ori: Multimodal Actuated Origami

Antonio Elia Forte\textsuperscript{a,b,c,*}, David Melancon\textsuperscript{a,*}, Leon M. Kamp\textsuperscript{a,d}, Benjamin Gorissen\textsuperscript{e}, and Katia Bertoldi\textsuperscript{a,1}

\textsuperscript{a}J. A. Paulson School of Engineering and Applied Sciences, Harvard University, Cambridge, MA 02138, USA; \textsuperscript{b}Department of Electronics, Information and Bioengineering, Politecnico di Milano, Milan, 20133 Italy; \textsuperscript{c}Department of Engineering, King’s College London, London, WC2R 2LS, UK; \textsuperscript{d}Department of Applied Physics, Eindhoven University of Technology, Postbus 513, 5600 MB, Eindhoven, The Netherlands

Recently, inflatable elements integrated in robotics systems have enabled complex motions as a result of simple inputs. However, these fluidic actuators typically exhibit unimodal deformation upon inflation. Here, we present a new design concept for modular, fluidic actuators that can switch between deformation modes as a response to an input threshold. Our system comprises bistable origami modules in which snapping breaks rotational symmetry, giving access to a bending deformation. By tuning geometry, the modules can be designed to snap at different pressure thresholds, rotate clockwise or counterclockwise when actuated, and bend in different planes. Due to their ability to assume multiple deformation modes as a response to a single pressure input, we call our system MuA-Ori, or Multimodal Actuated Origami. MuA-Ori provides an ideal platform to design actuators that can switch between different configurations, reach multiple predefined targets in space, and move along complex trajectories.

Origami | Bistability | Inflatable | Actuator

Introduction

Deformable and inflatable components have increasingly been integrated into robotic systems, as they provide complex deformations (1, 2), the ability to morph into target shapes (3–7), and inherent compliance, which in turn enables safe interactions (8–10). These, however, suffer from a common drawback: an intrinsic one-to-one relationship between input pressure and output deformation. Upon inflation, fluidic actuators exhibit unimodal deformation, which accentuates as pressure increases (11). In order to overcome this and create complex functionality, multiple actuators can be assembled and carefully sequenced (12–16), or multiple chambers of a single actuator can be pressurized independently (17, 18). As an alternative route, bidirectional bending has been achieved by harnessing material inextensibility (19) and non-linearity (20, 21). However, a design strategy that enables arbitrary deformation modes with a single input is still absent.

In an effort to design novel robotic systems, engineers and scientists have recently explored origami principles to create machines that are self-foldable (22–24) and realizable in a variety of materials using both planar and 3D fabrication techniques (25–28). The functionality of such origami robots can be further expanded if the crease pattern supports a non-convex energy landscape, which enables multiple stable states (29–37). For example, self-locking grippers (38) and energy-absorbing components for drones (39) were designed out of multistable origami sheets based on the tiling of the degree-four vertex; bistable mechanical bits and logic elements were created by introducing multistability in the classic waterbomb origami pattern (30, 40, 41) and, finally, bistable configurations of the Kresling pattern (31, 42) were exploited to generate locomotion via peristaltic motion (43) or differential friction (44), create flexible joints for robotic manipulation (45), and store mechanical memory (36, 46). All together, these examples show that multistable origami are a promising platform to realize fluidic actuators capable of supporting arbitrary deformation modes when actuated with a single input.

Inspired by the potential of the Kresling pattern in the design of robotic systems, here we employ this classic origami fold as a building block to realize multi-output, but single-input actuators. As part of our strategy, we start with a monostable Kresling pattern and make it bistable by introducing two additional valley creases in one of its panels (see Figs. 1a-b). During inflation, this panel unfolds and snaps outward, breaking the rotational symmetry of the module. Importantly, upon vacuum such asymmetry gives rise to bending in the opposite direction to the bistable panel up until a critical negative pressure is reached: at this point the panel snaps inward, resetting the module to its initial state. We use a combinatorial approach to couple multiple modules and create actuators with prescribed and complex deformation states that can be reached by controlling a single pressure input. Given that they output multiple deformation modes with a single input, we call our devices Multi-modal Actuated Origami, or MuA-Ori. MuA-Ori offers new opportunities for the design of robotic systems capable of performing complex tasks despite the simple actuation scheme, as demonstrated by the design of a land rowing robot.

The MuA-Ori building blocks based on Kresling pattern

To realize our nonlinear and reconfigurable MuA-Ori, we use origami building blocks comprising one layer of the classic Kresling pattern (also known as nejiri ori) (42). More specifically, in its initial, undeformed state, the single module consists of two hexagonal caps with edges of length $l = 30\text{ mm}$, separated by a distance $h = 24\text{ mm}$, and rotated by an angle $\alpha = 30^\circ$ with respect to each other (see Fig. 1a). The hexagons are connected at each side by a panel comprising a pair of triangular facets coupled by alternating mountain (i.e. edges $A'B$ and $AB'$) and valley (i.e. edge $BB'$) creases. Since the Kresling pattern is not rigid foldable (31), any change in its internal volume will lead to an incompatible configuration. To accommodate the resulting geometrical frustration, we 3D-print 1-mm thick triangular facets out of a compliant material (TPU95A from Ultimaker with Young’s modulus $E = 26\text{ MPa}$) and reduce the thickness locally to 0.4 mm to create the hinges (see prototype in Fig. 1a). Further, to facilitate coupling between different modules, we 3D-print the hexagonal caps out of a stiffer material (PLA from Ultimaker with Young’s modulus $E = 2.3\text{ GPa}$) and, to form an inflatable cavity, we coat the origami unit with a thin layer of polydimethylsiloxane (PDMS) (see Supplementary Materials, Section S1 for fabrication details). Finally, it is worth noticing that the chosen values of
Fig. 1. Bistable origami modules as building blocks for multi-output, single input inflatable actuators. (a) Schematics of a monostable module based on the hexagonal-base Kresling origami pattern, along with a 3D-printed prototype. The panels of the monostable modules remain always folded inward. We refer to this state of deformation as state \( s^0 \). (b) Bistable module with a modified panel (highlighted in orange) made of four triangular facets \( A'B'O', AO'B', AO'B', \) and \( A'BO' \) and characterized by a depth \( \Delta \) from vertex \( O \) to \( O' \), along with a 3D-printed prototype displayed in its two stable states: state \( s^0 \) for which all panels (including the modified panel) are folded inward; and state \( s^1 \) for which the modified panel is popped outward (while all other panels are still folded inward). (c) State diagram of the pressurized origami modules. (d-e) Pressure vs. bending angle curves for the monostable and bistable origami modules during inflation and deflation. At the onset of deflation for the bistable unit, \( \theta \) is slightly larger than in the monostable module due to the geometrical incompatibility that arises in the structure when the modified panel is in the outward configuration. (f) Experimental positive and negative pressure thresholds, \( p^+_{\Delta} \) and \( p^-_{\Delta} \), as a function of the modified panel’s depth, \( \Delta \).

The parameters \( (h, l, \alpha) \) yield a monostable origami module (i.e. the Kresling pattern is only stable in its initial, undeformed state).

To investigate the response of a single module, we inflate it with water while submerged in water (to eliminate the effects of gravity - see Supplementary Materials, Section S2 for details). As expected (31), the Kresling unit extends and contracts (while twisting at the same time) upon inflation and deflation (Fig. 1c) and returns to its undeformed state as soon as the pressure is removed. This state of deformation, in which all panels are folded inward, is named \( s^0 \). When in this state the bending angle, \( \theta \), remains close to zero (Figs. 1d-e).

Aiming at unlocking different deformation modes with one single pressure input, we then take inspiration from bistability in degree-four vertices (33, 41, 47) and modify one of the original Kresling panels by introducing two additional valley creases (i.e. \( AO \) and \( A'O \) with \( O \) being the midpoint of \( BB' \), see Fig. 1b). While this effectively creates a degree-four vertex, it results in a monostable origami unit, as no snap-through instability is recorded upon inflation (see Supplementary Materials, Section S2 for details). To increase the geometric incompatibility during deployment and achieve bistability in the unit, we then move the degree-four vertex inward by \( \Delta \) (see Fig. 1b where \( \Delta \) is the norm of vector \( \overrightarrow{OO'} \) perpendicular to vectors \( \overrightarrow{AA'} \) and \( \overrightarrow{BB'} \)). Choosing \( \Delta = 3 \) mm, for example, we can fabricate an origami unit that can easily transition between two stable states: state \( s^0 \) for which all panels are folded inward, and state \( s^1 \) for which the modified panel is popped outward (while all other panels are still folded inward). Similar to the unit based on the classic Kresling pattern, upon inflation this modified module simply extends for which all panels bent inward if \( p < 26.1 \pm 0.9 \) kPa. However, at \( p^+ = 26.1 \pm 0.9 \) kPa (where the subscript refers to \( \Delta = 3 \) mm and the superscript refers to positive pressure), the unit snaps from state \( s^0 \) with the modified panel folded inward to state \( s^1 \) where it is popped outward (Fig. 1c)—a transition which is accompanied by a discontinuity of the bending angle \( \theta \) (see zoom-in in Fig. 1d). Finally, a further increase in pressure...
causes the unit to elongate until the maximum structural limit is reached. Afterward, when the input pressure is removed, the modified panel remains popped outward (see Fig. 1c) because of bistability. As such, when we apply negative pressure the unit not only contracts, but also bends (see Figs. 1c and e), exhibiting a behaviour that radically differs from that of the Kresling module. Specifically, we find that at first \( \theta \) monotonically increases until the two hexagonal caps come into physical contact effectively clipping the available range of bending deformation to \( \theta_{\text{max}} = 21.7 \pm 0.3^\circ \). As previously mentioned, this bending deformation is caused by the modified panel, which remains in the popped outward configuration (while the other panels fold under increasing negative pressure) and breaks the radial symmetry. Finally, when the negative pressure passes the threshold \( p^*_n = -21.2 \pm 0.7 \text{kPa} \) (where the superscript refers to negative pressure), the modified panel snaps back to the inward position (see snapping transition from state \( s^i \) to state \( s^0 \) in Fig. 1c) and \( \theta \) suddenly decreases (dashed orange arrow in Fig. 1e). If one continues to apply negative pressure to the module, the unit folds (almost) flat, as clearly shown by the decreasing trend of \( \theta \).

Next, we investigate the effect of the depth \( \Delta \) of our degree-four vertex panel on the positive and negative pressure thresholds, \( p^*_{\Delta^+} \) and \( p^*_{\Delta^-} \), as well as the deformed configurations reached upon snapping. The experimental results reported in Fig. 1f for \( \Delta = 2, 3, \) and \( 4 \text{ mm} \) indicate that the absolute value of the pressure thresholds increases with \( \Delta \) within the considered range. Differently, we find that the bending angle, the axial displacement of the end caps and their relative twist remain almost constant with \( \Delta \). In fact, the first reaches an upper limit when the caps get in contact with each other and the other two are dominated by the geometric characteristics of the Kresling pattern (see Fig. S7). Note that for \( \Delta < 2 \text{ mm} \), the modules are found to be monostable. This means that negligible bending is recorded upon application of negative pressure, since the degree-four vertex panel snaps back immediately. Differently, for \( \Delta \geq 4 \text{ mm} \), the positive pressure required to snap the modified panel outward is so high that the module fails (see Fig. S6).

### Extending the design space for complex outputs

After demonstrating that our bistable module can transition between different stable states with distinct deformation modes (i.e. extension, twisting, and bending), we next combine these units to form actuators able to achieve complex outputs: the MuA-Ori. By combining \( n \) modules, we can construct \((3 \times 2 \times 6 + 1 \times 2)^n = 38^n \) different actuators since for each module \( k \) we can select (i) either a regular Kresling module or a unit comprising a modified, degree-four vertex panel with depth \( \Delta_k \in \{2, 3, 4\} \text{ mm} \); (ii) the upper cap to be rotated clockwise or anticlockwise with respect to the bottom one, \( \theta_k \in \{+1, -1\} \), and (iii) the side on which the modified panel is located, \( f_k \in \{1, \ldots, 6\} \) (see Fig. 2a). Note that the domain of all constraints is discrete since we choose a limited set of values for \( \Delta \) and all end plates to have their sides aligned.

To navigate the vast design space systematically and efficiently, we develop a simple algorithm that predicts all stable states and snapping transitions for actuators comprising \( n \) modules upon inflation and deflation. First, we extract key geometric features from the experiments conducted on single units, i.e. height, twisting, and bending angle of the top cap associated to all stable states and snapping transitions (see Figs. S7-S8). When assuming pressure continuity, these data allow the prediction of all configurations for the stable states and snapping transitions of any \( n \text{-units actuator} \) (see Supplementary Materials, Section S3 for details on the algorithm).

---

**Fig. 2. Extending the design space.** (a) For each \( k \) module we define 3 geometrical parameters: \( \Delta^k, \theta^k, \) and \( f^k \). Note that, for simplicity, we also assign \( f^k \), but only the location of the modified panel relative to its neighbor is important. (b) State diagram for a 2-units actuator characterized by \( \{\Delta^1, \theta^1, f^1; \Delta^2, \theta^2, f^2\} = \{2/3, 4/6\} \); geometrical model prediction on the left, experimental snapshots on the right. (c) Normalized deployment \( ||d||/h \) and angle \( \theta_{\text{act}} \) for every stable state of the 2-units actuator characterized by \( \{\Delta^1, \theta^1, f^1; \Delta^2, \theta^2, f^2\} = \{2/3, 4/6\} \). (d) Normalized deployment \( ||d||/h \) and angle \( \theta_{\text{act}} \) for every stable state of any possible 2-units (triangles) and 3-units (circles) actuator in the design space.
Note that we also assume perfect coupling between units, so that the pressure thresholds, \( p_+ \) and \( p_- \), found in the experimental characterization of Fig. 1f, remain unchanged and identical for units with the same geometrical parameters.

As an example, in Fig. 2a, we consider an actuator comprising \( n = 2 \) modules characterized by \( [\Delta c_1 f^1; \Delta c_2 f^2] = [2/3; 4/6] \), where we assume the first unit to be the one at the bottom. As expected, this MuA-Ori has four stable states at atmospheric pressure, \( s^{01} \) (where the subscripts \( i, j = 0, 1 \), refer to the state of the modified panels, with \( \Delta = 2 \) and 4 mm, respectively), and six snapping transitions that can be triggered by varying the internal pressure (Fig 2b). Whereas the stable states \( s^{10} \) and \( s^{11} \) can be readily obtained by simply increasing pressure, a more complex pressure path is required to achieve state \( s^{01} \), as one has to (i) increase pressure above \( p_+ \) and then (ii) decrease it below \( p_- \). To validate the model, we then conduct experiments on units connected via 3D-printed screws (see Fig. S2 for details) and find that they closely match the predictions (see Fig 2b and Movie S3).

Since our final goal is to build actuators with programmed deformation modes, we further quantify the deployment of our actuator by recording the vector connecting the caps’ centroids, \( \mathbf{d} \), at each stable state and snapping transition. In Fig. 2c we report the norm of \( \mathbf{d} \) normalized by the height of a single module, \(|\mathbf{d}|/h_i\), and the angle between each \( \mathbf{d} \) and the \( z \)-axis, \( \theta_{act} \), for all 12 configurations supported by the actuator characterized by \( [\Delta c_1 f^1; \Delta c_2 f^2] = [2/3; 4/6] \). We find that in the positive pressure regime the achievable deformation is limited to simple extension, resulting in configurations all clustered in a portion of the design space characterized by \( \theta_{act} \approx 5^\circ \) and \(|\mathbf{d}|/h \approx 2 \) (gray and red triangles). By contrast, behaviors characterized by large bending angles (\( \theta_{act} \approx 20^\circ \)) unfold under negative pressure (blue triangles). Importantly, such bending provides opportunities for the design of actuators that can reach a wider range of configurations. This becomes even clearer when expanding the range of possible configurations by varying the parameters \( \Delta k_i \), \( c_i \), and \( f_i \), and considering not only \( n = 2 \) but also \( n = 3 \) building blocks (triangles and circles in Fig. 2d, respectively). Once again, we notice that all configurations under positive and atmospheric pressure cluster on low values of bending angle and a normalized deployment that is roughly equal to the number of building blocks considered. Differently, in the negative pressure regime a rich set of deformations unfolds, characterized by a wide range of achievable \( \theta_{act} \) (the maximum values of \( \theta_{act} \) increases from 40° to 50° from \( n = 2 \) to \( n = 3 \)) at near-constant normalized deployment (\(|\mathbf{d}|/h\) increases slightly from about 1.5 to 2 from \( n = 2 \) to \( n = 3 \)). These results indicate that by stacking our origami modules to form arrays, we can realize actuators capable of supporting a variety of deformation modes, which can be selected by varying the level of applied pressure.

**Inverse design to reach multiple targets**

Motivated by the results in Fig. 2, we then investigate the behavior of arrays with a larger number of units. Our goal is to build actuators capable of switching between target deformation modes when inflated with a single pressure source. However, since the use of \( n \) modules leads to \( 3^n \) possible actuator designs (i.e. \( 54,872 \) for \( n = 3 \) and 2,085,196 for \( n = 4 \)), it is crucial to use a robust algorithm to efficiently scan the range of responses that can be achieved and identify configurations leading to the target deformation modes. Toward this end, given the discrete nature of our design variables, we use a greedy algorithm based on the best-first search method (48, 49) — a progressive local search algorithm that, at each iteration, minimizes the cost function by looking at a set of available solutions. Although there exists many algorithms to solve this type of discrete optimization problems (50, 51), we find that the greedy algorithm provides the best trade-off between accuracy and computational cost (see Supplementary Information, Section S4 for details and comparison of the different algorithms). Specifically, our greedy algorithm identifies MuA-Ori comprising \( n \) units built out of \( n_s \) super-cells each with \( n_u \) modules (so that \( n = n_u \cdot n_s \)) whose tip can reach a desired set of targets arbitrarily positioned in the surrounding space. At the first iteration, the algorithm starts by selecting...
the actuator super-cell design that minimizes
\[
\Psi = \frac{1}{n_{\text{targets}}} \sum_{n=1}^{n_{\text{targets}}} \min ||d - T_n||,
\]
where \(n_{\text{targets}}\) is the number of targets, and \(T_n\) is the vector connecting the \(m\)-th target with the origin. Once the first super-cell is chosen, the algorithm stores it in memory and starts a second iteration. This comes to an end when the algorithm identifies a second super-cell that connects to the first, one minimizes Eq. (3). The first two super-cells are then stored in memory and the algorithm advances to the next one. Note that in this study, to balance the number of available designs and computational cost, we set the greedy algorithm to consider super-cells made of 3 units (i.e. \(n_u = 3\)) (see Figs. S15 and S18 for a comparison across super-cells made with different \(n_u\) and, to avoid fabricating excessively long actuators whose response could be affected by gravity, terminate the algorithm after stacking 5 super-cells.

To demonstrate our approach, we select two distinct sets of targets within the reachable space (see black and red circular markers in Fig. 3a and Supplementary Materials Figs. S15-S19 for additional targets). For the first set of black targets (T1, T2) with small height and radius of action, we find that the objective function \(\Psi\) is minimized for an actuator with \(n_s = 1\) and \([\Delta c^1 f^1; \Delta c^2 f^2; \Delta c^3 f^3] = [4; 2; 3; 6; 4; 2]\) (see Fig. 3b). As this 3-unit actuator only comprises modules with \(\Delta = 3\) and 4 mm, its state diagram (shown in Fig. 3d) is qualitatively similar to the one reported in Fig. 2b. To reach the target T1, the internal pressure must first be raised above \(p_4^+\) and then lowered to \(p_5^-\) (see shaded gray line in Fig. 3c). Successively, the target T2 is reached by further lowering the pressure to \(p_4^-\). It is worth noticing that, in this example, increasing the pressure above \(p_4^+\), decreasing it below \(p_4^-\), and finally bringing it back to zero results in a triangular path for the centroid of the top plate, which is spanned in a counterclockwise direction (Fig. 3d). Importantly, when we apply this pressure history to a physical sample, the actuator’s tip follows the predicted trajectory and approaches the two targets very closely (see Fig. 3d), confirming the validity of our design strategy. As predicted by our model, the actuator starts straight, bends toward T1, bends toward T2 and finally returns to the straight configuration \(O\) after reset.

For the second set of red targets (T1, T2, T3), the larger height and radius of action of the targets lead to an optimal actuator comprising \(n_s = 4\) super-cells (note that the convex shape of \(\Psi\) is due to a correlation between the optimal number of units and the average distance of the targets from the origin—see Supplementary Materials Fig. S19). As shown in Fig. 3b, this 12-unit actuator comprises the classic Kresling module as well as bistable units with \(\Delta = 2, 3, 4\) mm. This results in 8 stable states, 14 snapping transitions, and a more complex state diagram in which not all targets are reached consecutively by continuously decreasing pressure (Fig. 3e).

More specifically, to move from T1 to T2, the actuator has to be reset by decreasing the pressure below \(p_5^-\) before increasing above \(p_4^+\) and then lowering it to \(p_5^-\). As such, in this case the centroid of the top plate of the actuator passes through the straight configuration \(O\) when moving from T1 to T2 and its trajectory comprises two disconnected loops, \(O - T1\) and \(O - T2 - T3\) (Fig. 3f). Note that we can add additional constraints to our greedy algorithm to make sure the targets fall within the same closed loop on the state diagram. This obviously leads to a different design and may increase the targets error (see Supplementary Materials Fig. S17 for details). However, the ability to fully control the actuator’s sequential trajectory makes the platform ideal for robotic applications.

MuA-Ori for robotic applications
To show the potential of the MuA-Ori for robotic applications we take inspiration from the closed trajectory of a rowing stroke, and use the closed triangular loop of the 3-unit actuator in Fig. 3e to propel a land rowing robot (Figs. 4a). The robot comprises two symmetric arms (corresponding to the 3-unit MuA-Ori in Fig. 3) connected through a single fluidic line and mounted on a wheeled chassis. Further, to harnesses the cyclic motion of the actuator and generate positive mechanical work with the ground, we connect two rigid rods to the outer caps that serve as stroke amplifiers and attach silicon patches at their ends to increase friction with the ground. When increasing the pressure above \(p_4^+\) and then lowering it to \(p_5^-\) the two actuators reach \(T1\). At this point the rigid rods touch the ground. Then, when we further decrease the pressure, the rigid rods move backward to \(T2\) creating a forward thrust and eventually lift off from the ground, completing the stroke. Finally, lowering the pressure below \(p_5^-\) resets the locomotion cycle (see Figs. 4c and d). As shown in the experimental snapshots in Fig. 4e we can harness this particular trajectory instructed by the model to create locomotion: the robot ad...
vances of about 16 cm in 20 cycles. Note that, differently from other robotic platforms with similar performance but requiring one or more actuators per leg with dedicated pressure sources (12, 16), our robots operate with a single pressure input, which largely simplifies its control.

Conclusion

To summarize, in this work we have presented a platform to design fluidic origami robots that can switch between a range of different configurations using only one pressurized input: the MuA-Ori. The key characteristics of our platform is an origami building block with a degree-four vertex panel, which can be geometrically programmed to snap at a certain input threshold, inducing bending. This, together with the position of the modified panel in the origami module and their direction of rotation, constitute the parameters of a rich design space that we can efficiently scan with a custom greedy algorithm. Therefore, instructed by this model, we are able to automatically design and build optimal robotic arms that can reach a set of defined targets, and actuators capable of tracing spacial trajectories in a programmed sequence, enabling mechanical tasks and functions. While in this study we have used a simple geometric model to identify optimal designs, a fully mechanical model may be needed in order to extend the proposed approach to larger designs for which the effect of gravity is not negligible. In addition, the current design space could be further expanded through investigating the effect of other geometrical parameters (e.g., f, k, and α) on the resulting deformation of the modules, as well as expanding the range of the considered values of \( \Delta \). While this could lead to more complex deformation modes and enhanced functionality, a drawback is a more complex state diagram. This means that a given actuator might have to go through a longer loading history to reach some prescribed targets, increasing the operational time-span. A potential solution to this is to measure the volume at which the module snaps inward and outward, assume constant flow rate, and derive the time associated to each snapping transition. This time span could then be included as variable in the optimization algorithm, in order to find a design that reaches the target in the shortest possible time. To conclude, we envisage that the MuA-Ori concept hereby presented will serve as future strategy for the design of fluidic origami robots able to overcome the unimodal deformation constraint, despite the simple actuation.

Materials and Methods

Details of the design, materials, and fabrication methods are summarized in Supplementary Materials, Section S1. The experimental procedure of the inflation with water to measure the pressure-volume curve is described in Supplementary Materials, Section S2, along with additional experimental data.experimental data. Details on the numerical mode are provided in Supplementary Materials, Section S3. Finally, the optimization algorithms used in this study are described in detail in Supplementary Materials, Section S4, along with additional data.

Acknowledgments

We thank Eder Medina for his assistance in the development of the fabrication methodology. Author contributions: A.E.F., D.M., B.G. and K.B. proposed and developed the research idea. A.E.F designed and fabricated the samples. A.E.F., D.M., B.G. and L.K. performed the experiments. A.E.F. and D.M. designed the optimization. D.M. conducted the numerical calculations. A.E.F., D.M., and K.B. wrote the paper. K.B. supervised the research. Funding: Research was supported by the NSF grants DMR-2011754, DMR-1922321 and EFRI-1741685. A.E.F. acknowledges that this project has received funding from the European Union’s Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No 798244. Competing interests: The authors declare no conflict of interest. Data and materials availability: All data needed to evaluate the conclusions of this study are present in the paper or the Supplementary Materials.

Supplementary materials

Section S1. Fabrication
Section S2. Testing
Section S3. Model
Section S4. Optimization
Figure S1. 3D-printed origami modules
Figure S2. Multi-unit actuator fabrication and assembly
Figure S3. Single pressure input origami robot
Figure S4. Experimental setup of the inflation with water
Figure S5. Experimentally measured response of the \( \Delta = 3 \) mm module
Figure S6. Experimental pressure, displacement, and bending angle curves of our origami modules
Figure S7. Effects of \( \Delta \)
Figure S8. Modeling the stable states and snapping transitions of our origami modules
Figure S9. Actuator’s parameters
Figure S10. State diagrams
Figure S11. A 2-units actuator characterized by \([\Delta^4 c^1 f^1; \Delta^2 c^2 f^2] = [2/3, 4/3] \)
Figure S12. Design space for actuators with \( n \in \{1, 2, 3, 4\} \)
Figure S13. Greedy algorithm
Figure S14. Comparison between optimization algorithms with integer constraints
Figure S15. Greedy algorithm results
Figure S16. A 6-units actuator reaching three targets
Figure S17. The 12-units actuator with additional constraint
Figure S18. Random targets error
Figure S19. Optimal number of units as a function of the target radius
Movie S1. Fabrication
Movie S2. 2-units actuators
Movie S3. 12-units actuator, triple target objective
Movie S4. Robotic application

References

1. D. Rus and M. Tolley. Design, fabrication and control of soft robots. Nature, 521:467–475, 2015.
2. Elliott W. Hawkes, Laura H. Blumenschein, Joseph D. Greer, and Allison M. Okamura. A soft robot that navigates its environment through growth. Science Robotics, 2(8), 2017. . URL https://robotics.sciencemag.org/content/2/8/eaan3028.
3. Meléna Skouras, Bernhard Thomaszewski, Peter Kaufmann, Ashok Garg, Bernd Bickel, Eitan Grinspun, and Markus Gross. Designing inflatable structures. ACM Trans. Graph., 33(4), July 2014. ISSN 0730-0301. . URL https://doi.org/10.1145/2601097.2601166.
4. J. H. Pikul, S. Li, H. Bai, R. T. Hanlon, I. Cohen, and R. F. Shepherd. Stretchable surfaces with programmable 3d texture morphing for synthetic camouflaging skins. Science, 358(6360): 219–214, 2017. ISSN 0036-8075. . URL https://science.sciencemag.org/content/358/6360/210.
5. Mina Konaková-Lukovič, Julian Poonata, Keenan Crane, and Mark Pauly. Rapid deployment of curved surfaces via programmable auxetics. ACM Trans. Graph., 37(4), July 2018. ISSN 0730-0301. . URL https://doi.org/10.1145/3170517.3201373.
6. Emmanuel Sifert, Etienne Reysset, José Bico, and Benoît Roman. Bio-inspired pneumatic shape-morphing elastomers. Nature Materials, 18(1):169–169, 2019. . URL https://doi.org/10.1038/s41563-018-0219-x.
7. Lushuai Jin, Antonio Elia Forte, Boile Deng, Ahmad Rafsani, and Katia Bertoldi. Kirigami-inspired inflatables with programmable shapes. Advanced Materials, page 2001863, 2020.
8. Mark Runciman, Ara Darzi, and George P. Mylonas. Soft robotics in minimally invasive surgery. Soft robotics, 2019.
9. Panagiotis Polygerinos, Zheng Wang, Kevin C Galloway, Robert J Wood, and Conor J Walsh. Soft robotic glove for combined assistance and at-home rehabilitation. Robotics and Autonomous Systems, 73:135–143, 2015.
Supplementary Information

S1. Fabrication

The actuators tested in this study are constructed by connecting 3D-printed origami modules. This section gives details on the design and fabrication of the 3D-printed modules as well as their assembly to create multi-unit actuators and robots.

Design of the 3D-printed origami units. Each origami module is 3D-printed using a commercially available multi-material printer (Ultimaker 3). To account for geometric incompatibility during inflation, we print the 1-mm thick triangular facets out of compliant thermoplastic polyurethane (Ultimaker TPU 95A with tensile modulus $E = 26$ MPa). The thickness decreases to 0.4 mm at the junction of the triangular facets (hinges), to allow more compliance. This value is the lowest possible thickness our printer is able to print with a 0.4 mm print core.

Further, to enable rigid connection of different units and increase bistability, we print the end caps as well as the four triangular facets of the bistable cell out of stiff polyactic acid (Ultimaker PLA with tensile modulus $E = 2.3$ GPa). As shown in Fig 5, the single module consists of two hexagonal caps with edges of length $l = 30$ mm, separated by a distance $h = 24$ mm, and rotated by an angle $\alpha = 30^\circ$ with respect to each other. To enable coupling of different units, we print a screw and a threaded hole on the top and bottom surfaces with length $w = 6$ mm and thread size $d_T = 24$ mm.

Assembly of an inflatable multi-unit actuator. Below are the eight steps needed to fabricate and assemble an inflatable origami actuator sample made of multiple modules (see Fig. 6 and Movie S1):

- **Step 1**: we 3D-print (Ultimaker 3) each origami unit out of polyactic acid (Ultimaker PLA) and thermoplastic polyurethane (Ultimaker TPU 95A), using 0.4 mm print cores with the fine default setting.
- **Step 2**: we cut the 3D-printed adhesion skirt with scissors.
- **Step 3**: we remove the 3D-printed support material inside of the origami unit with pliers.
- **Step 4**: we insert a toric joint on the connection screw to make the unit airtight (see Steps 5-8).
- **Step 5**: we assemble multiple units together through the connection screws ensuring a tight assembly through the toric joints inserted in Step 4.
- **Step 6**: we coat the sample with a 0.5 mm layer of polydimethylsiloxane (PDMS) and let it cure for 24 hours.
- **Step 7**: we fit end caps, making sure to have a tight assembly through the toric joints inserted in Step 4, to create an airtight cavity. Note that one of the end caps has an inlet for fluidic actuation.
- **Step 8**: we test the origami actuator by connecting it to a fluidic supply.
Design of the single pressure input origami robot. The origami robot is constructed by mounting two multi-unit actuators (each one acting as a crawling arm) on a wheeled chassis made of LEGO bricks (see Figs. 7a and b). To create a single airtight cavity, we connect the two multi-unit actuators through a fluidic line (1/16" soft plastic tubing and luer lock connectors) and a coupler (1/16" acrylic plate fixed on the chassis). Finally, we glue 10 cm-long amplifiers (1/4" acrylic plate) on the top caps of the actuators to accentuate the stroke during inflation and deflation. Note that we add a patch of silicon rubber on the amplifiers’ tip (ecoflex 5 from Smooth-on) to increase the friction coefficient when in contact with the ground.
Fig. 7. **Single pressure input origami robot.** (a) The two identical actuators forming the arms of the robot. (b) Schematics of the origami robot. (c) Components of the origami robot. (d) Assembled origami robot.
S2. Testing

To characterize the experimental response of the fabricated origami building blocks, we inflate them with water— to eliminate the influence of air compressibility— and measure their internal pressure while tracking their height and bending angle. As shown in Fig. 8, we use a syringe pump (Pump 33DS, Harvard Apparatus) to displace water into the origami unit at 10 mL/min, measure the pressure using a pressure sensor (ASDXRRX015PDAA5 with a measurement range of ±15 psi by Honeywell), and track the height, bending angle, and twist angle of the upper cap using two digital cameras (front and top view with two SONY RX100 V). Note that we submerge the entire unit in a water tank to eliminate the influence of gravity while eliminating air from all supply tubes and calibrate the pressure sensor to atmospheric pressure before each measurement cycle.

Fig. 8. Experimental setup of the inflation with water. Schematic of the test setup used to characterize the bending angle vs./ pressure and height vs./ pressure curves of the origami units with (1) syringe pump, (2) pressure sensor, (3) water tank, (4) origami unit, and (5) digital cameras.

We report in Fig. 9 the experimentally measured pressure, \( p \), as a function of the end caps' displacement, \( u_z \), and the bending angle, \( \theta \), for the module with \( \Delta = 3 \) mm. Further, we highlight on the curves the pressure thresholds, \( p^+_3 \) and \( p^-_3 \), as well as the maximum displacement, \( u_{z \text{max}, \pm} \), and bending angle, \( \theta_{\text{max}, \pm} \), at snap-through for both the inflation and deflation regime.

Fig. 9. Experimentally measured response of the \( \Delta = 3 \) mm module. (a) State diagram of the pressurized origami module. (b) Experimentally measured pressure, \( p \), as a function both end caps' displacement, \( u_z \), and bending angle, \( \theta \), for the origami module with \( \Delta = 3 \) mm. In the plot, we highlight the pressure thresholds, \( p^+_3 \) and \( p^-_3 \), as well as the maximum displacement, \( u_{z \text{max}, \pm} \), and bending angle, \( \theta_{\text{max}, \pm} \), at snap-through for both inflation and deflation.

We summarize all experimental results in Fig. 10 and plot the pressure, displacement, and bending angle of each design tested in this study as a function of normalized time, \( T \), for both the inflation and deflation regime. To validate repeatability, we test for each design three specimens and report the mean (solid lines) and standard deviation (shaded region). We find that the classic Kresling module as well as the modified-panel module with \( \Delta = 0 \) mm do not show any snap-through instability. Modules with \( \Delta = 1 \) mm show a snapping transition during inflation, but bend marginally when later deflated. Modules with \( \Delta = 2, 3, 4 \) mm are bistable as they exhibit discontinuity in their \( p - T \), \( u_z - T \), and \( \theta - T \) curves, and show substantial bending when deflated from the snapped configuration. Modules with \( \Delta = 5 \) mm break during the inflation before the panel snaps outward.
Fig. 10. Experimental pressure, displacement, and bending angle curves of our origami modules. Experimentally measured pressure (left vertical axis), end caps displacement (right vertical axis), and bending angle (right vertical axis) of each design tested in this study as a function of normalized time, $T = t/t_{end}$ (where $t$ is real time and $t_{end}$ the duration of the test), for both (a) inflation and (b) deflation. To validate repeatability, we test for each design three specimens and report the mean (solid lines) and standard deviation (shaded region). We find that the classic Kresling module as well as the modified-panel module with $\Delta = 0$ mm do not show any snap-through instability. Modules with $\Delta = 1$ mm show a snapping transition during inflation but bend marginally when later deflated. Modules with $\Delta = 2, 3, 4$ mm are bistable as they exhibit discontinuity in their $p - T$, $u_z - T$, and $\theta - T$ curves, and show substantial bending when deflated from the snapped configuration. Modules with $\Delta = 5$ mm break during the inflation before the panel snaps outward.
From the experimental curves, we can extract the pressure thresholds at which the modified panel snaps outward during inflation, \( p^+ \), and inward during deflation, \( p^- \). We summarize the mean and standard deviation of these pressure thresholds in Fig. 11 for the bistable modules with \( \Delta = 2, 3, \) and \( 4 \) mm. We also report in Fig. 11, the displacement and the bending angle at snap-through, \( u^{\text{max,} \pm} \) and \( \theta^{\text{max,} \pm} \) for both the inflation and deflation regime.

\[
\begin{align*}
\Delta = 2 \text{ mm} \\
\Delta = 3 \text{ mm} \\
\Delta = 4 \text{ mm}
\end{align*}
\]

Inflation

\[
\begin{align*}
\text{Experimental results}
\end{align*}
\]

Deflation

\[
\begin{align*}
\text{Experimental results}
\end{align*}
\]

\[
\begin{align*}
p^{\pm}, & \quad u^{\text{max,} \pm}, \\
\theta^{\text{max,} \pm}, & \quad \phi^{\text{max,} \pm}
\end{align*}
\]

Fig. 11. Effects of \( \Delta \). Bar chart of the experimentally measured pressure thresholds, \( p^{\pm} \), end caps displacement, \( u^{\text{max,} \pm} \), bending angle, \( \theta^{\text{max,} \pm} \), and twist angle, \( \phi^{\text{max,} \pm} \), at snap-through during both inflation (a) and deflation (b). The values for \( p^{\pm} \), \( u^{\text{max,} \pm} \), and \( \theta^{\text{max,} \pm} \) are extracted from the curves shown in Fig. 10. The values for \( \phi^{\text{max,} \pm} \) were obtained measuring the twist angle at each stable configuration and snapping transition during the experiments.

Forte et al. 13
S3. Model

**Origami module.** We start by using the geometric quantities that we track in our experiments (shown in Fig. 11) to reconstruct the configuration of the origami module just before and just after each snapping transition (see Fig. 12).

![Diagram](https://example.com/origami-module-diagram.png)

**Fig. 12.** Modeling the stable states and snapping transitions of our origami modules. (a) Typical state diagram for an origami module with a modified panel of depth \( \Delta \). The unit can transition between two stable states: state \( s^0 \) when the modified panel is folded inward, and state \( s^1 \) when the panel is popped outward and stays in that position even when the input pressure is removed (b-g) Reconstructed geometry of the origami module at each stable state and before and after each snapping transition.

**Actuator comprising \( n \) modules.** We can create an actuator made of \( n \) units by simply combining the different stable states and snapping transitions found for the single unit model described in Fig. 12. Note that we impose that any \( n \)-units actuator forms a closed, inflatable cavity (i.e. they are all subjected to the same internal pressure). By combining \( n \) modules, we can construct \( (3 \times 2 \times 6 + 1) \times 2 \) different actuators since for each module \( k \) we can select (i) either a regular Kresling module or a unit comprising a modified panel with depth \( \Delta_k \in \{2, 3, 4\} \) mm; (ii) the upper cap to be rotated clockwise or anticlockwise with respect to the bottom one, \( c_k \in \{/\, \backslash\} \), and (iii) the side on which the modified panel is located, \( f_k \in \{1, \ldots, 6\} \) (see Fig. 13).

![Diagram](https://example.com/actuator-diagram.png)

**Fig. 13.** Actuator's parameters. Our reconfigurable actuators are defined by the modified panel's depth, \( \Delta^k \), the initial rotation of the upper cap, \( c^k \), and the position of the modified panel, \( f^k \), of each their \( k \) unit.

For an actuator made of \( n \) units, the number of stable states is equal to \( 2^n \Delta \), where \( n_\Delta \) is the number of unique modified panel depths \( \Delta \). Note that we assume all units with the same \( \Delta \) snap synchronously at the pressure thresholds, \( p^+_\Delta \) and \( p^-_\Delta \). Since in this study, we consider only the discrete set \( \Delta \in \{2, 3, 4\} \) mm, all our actuators have either \( n_\Delta = 0, 1, 2, \) or \( 3 \). For each different \( n_\Delta \), we report the corresponding state diagram in Fig. 14.
In Fig. 15a we show the state diagram of an actuators comprising two modules characterized by $[\Delta^1c^1f^1; \Delta^2c^2f^2] = [2//3; 4\backslash3]$ and compare the outputs from the geometric model to the experiments done on a physical prototype. Further, for each of the stable states and snapping transitions in the state diagram, we record the vector connecting the caps' centroids, $d$. In Fig. 15b, we report the normalized deployment, $||d||/h$, and angle, $\theta_{act}$, for each stable state and snapping transition of the actuator characterized by $[\Delta^1c^1f^1; \Delta^2c^2f^2] = [2//3; 4\backslash3]$. Further, in Fig. 16 we report $||d||/h$ and $\theta_{act}$ for each stable state and snapping transition of every actuator design with $n \in 1, 2, 3, 4$. 

**Fig. 14. State diagrams.** (a) State diagram for any $n$-units actuator made only of Kresling modules. (b-d) State diagram for any $n$-units actuator made with 1, 2, and 3 unique modified panel's depth, i.e. $n_A = 1, 2, 3$, respectively.

**Fig. 15.** A 2-units actuator characterized by $[\Delta^1c^1f^1; \Delta^2c^2f^2] = [2//3; 4\backslash3]$. (a) State diagram for a 2-units actuator characterized by $[\Delta^1c^1f^1; \Delta^2c^2f^2] = [2//3; 4\backslash3]$; geometrical model prediction on the left, experimental snapshots on the right. (b) Normalized deployment, $||d||/h$, as a function of the bending angle $\theta_{act}$, for every stable state and snapping transition.
Fig. 16. Design space for actuators with $n \in \{1, 2, 3, 4\}$. Normalized deployment, $|d|/h$, and bending angle, $\theta_{\text{act}}$, for all possible 1-unit (a), 2-unit (b), 3-unit (c), and 4-unit (d) constructed by combining our library of modules.
S4. Optimization

To identify actuators capable of achieving target deformation modes, we solve the following discrete optimization problem

\[
\min_{\Delta^k, c^k, f^k} \Psi \left( \Delta^k, c^k, f^k; n \right)
\]

s.t. \( \Delta^k \in \{2, 3, 4\} \text{mm} \)
\( c^k \in \{/\, \backslash\} \)
\( f^k \in \{1, 2, 3, 4, 5, 6\} \)
\( k \in \{1, 2, \ldots, n\} \)
\( n \in \mathbb{Z}^+, \)

where \( \Psi \) is the cost function, \( n \) is the number of units making the actuator and \( \Delta^k, c^k, \) and \( f^k \) are are the modified panel depth, the orientation of the upper cap with respect to the bottom one, and the side on which the modified panel is located for the \( k \)-unit in the array. Note that all design variables (i.e. \( \Delta^k, c^k, \) and \( f^k \)) are constrained to be integer value and, for the sake of simplicity, we solve the optimization problem multiple times for fixed number of units \( n \in \{1, 15\} \).

In the main text we use the optimization algorithm to identify actuators whose tip can approach a desired set of target points and therefore define the cost function as

\[
\Psi = \frac{1}{n_{\text{targets}} - h} \sum_{m=1}^{n_{\text{targets}}} \min \|d - T_m\|, \tag{3}
\]

where \( n_{\text{targets}} \) is the number of targets, and \( T_m \) is the vector connecting the \( m \)-th target with the origin. Here, we also consider two additional optimization problems and use the algorithm to identify actuators that

- maximize the angle \( \theta_{\text{act}} \) between the vector connecting the cap's centroids, \( d \), and the \( z \)-axis. For this case we define the cost function as
  \[
  \Psi \left( \Delta^k, c^k, f^k; n \right) = -\theta_{\text{act}}. \tag{4}
  \]
- maximize the deployment height \( ||d||/h \). For this case we define the cost function as
  \[
  \Psi \left( \Delta^k, c^k, f^k; n \right) = -||d||/h, \tag{5}
  \]

Optimization algorithms. There are many algorithms able to solve an optimization problem with integer constraints such as the one presented in Eq. [2]. In this study, we used three classic algorithms: (i) the genetic algorithm with integer constraints; (ii) the integer optimization via a surrogate model; and (iii) the greedy algorithm based on best-first search. Note that given the high-dimensionality and complexity of this optimization problem, there is no guarantee that these algorithms will lead to a unique global minimum.

Genetic algorithm with integer constraints. We started by using the genetic algorithm with integer constraints, which attempts to minimize a penalty function that depends on the fitness (value of the cost function \( \Psi \)) and feasibility (design variables are integer) of an individual. For this study we used the Matlab implementation of the algorithm (Matlab function \texttt{ga}) and imposed the constraint that all design variables, i.e. \( \Delta^k, c^k, \) and \( f^k \) must have integer values with upper and lower bounds reported in Eq. [2]. We ran the function \texttt{ga} multiple times, each time considering a fixed value of \( n \in \{1, 15\} \), using a population size of 200, a max stall generations (i.e. the consecutive number of generations with no change to the cost function value) of 500 and a maximum number of generations of 1000.

Integer optimization via a surrogate model. Next, we used the surrogate model optimization, which is a derivative-free method that replaces the complex and non-smooth objective function by a surrogate (i.e. an approximation of that function), which is created by sampling the objective function. For this study we used the Matlab implementation of the algorithm (Matlab function \texttt{surrogateopt}) and imposed the constraint that all design variables, i.e. \( \Delta^k, c^k, \) and \( f^k \) must have integer values with upper and lower bounds reported in Eq. [2]. We ran the algorithm for fixed values of \( n \) with a maximum number of function evaluations of 20,000.

Greedy algorithm based on best-first search. For the greedy algorithm, we developed and in-house Matlab code based on the best-first search method that creates an actuator with \( n \) units out of \( n_u \) super-cells, each super-cell made of \( n_s \) modules (so that \( n = n_u \cdot n_s \)). At the first iteration, the algorithm selects the super-cell design that minimizes \( \Psi \) and stores it in memory. Then, in the second iteration, we identify a second super-cell that, when connected to the first one, minimizes \( \Psi \). The first two super-cells are then stored in memory and the algorithm advances to the next iteration (see Fig. 17 and Algorithm 1 below).
Algorithm 1 Greedy algorithm based on best-first search

Set $n_{\text{max}}$;
Set $n_u$;
Set $n_s = 0$;

While $n_s \cdot n_u \leq n_{\text{max}}$

\[ n_s = n_s + 1 \]

if $n_s = 1$ then

- Calculate $\Psi$ for each actuator design with $(\Delta^k, c^k, f^k)$, $k = 1 : n_s \cdot n_u$;
- Find the actuator that minimizes $\Psi$ and set its design variables to $(\Delta^*, c^*, f^*)$;

else

- Calculate $\Psi$ for each actuator design with $(\Delta^k, c^k, f^k)$, where the set of variables from $k = 1 : (n_s - 1) \cdot n_u$ are coming from the previous iteration of $(\Delta^{k-1}, c^{k-1}, f^{k-1})$ and $k = (n_s - 1) \cdot n_u + 1 : n_s \cdot n_u$ are free.
- Find the actuator that minimizes $\Psi$ and set its design variables to $(\Delta^*, c^*, f^*)$;

Results. In the following we first compare the performance of the three algorithms and then present additional results obtained using the greedy algorithm.

Comparison between the three algorithms. To test and compare the three algorithms, we considered the set of three targets ($T_1$, $T_2$, $T_3$) shown in Fig. 3 of the main article and minimized the cost function given in Eq. [3]. In Fig. 18, we report the cost function value with respect to the number of generations/function evaluations as obtained using the three algorithms. We find that for all considered values of $n$ both the genetic algorithm with integer constraints and the surrogate model stall quickly, with a minimum value of the cost function of 1.04 and 1.12 reached for $n = 15$, respectively. Further, we find that the greedy algorithm with $n_u = 3$ outperforms the genetic algorithm and the surrogate model optimization as it identifies an actuator design that leads to $\Psi = 0.729$ for $n = 12$. Note that, for $n_u = 3$, the greedy algorithm requires about $2.75 \times 10^5$ evaluations of $\Psi$ to identify the optimal design, whereas the surrogate model takes about $1 \times 10^9$ evaluations of $\Psi$ (the genetic algorithms requires about $8 \times 10^5$ evaluations of $\Psi$ with a population size of 200). However, the greedy algorithm does not require any other operation apart from a simple computation of $\Psi$ during each iteration. Differently, the surrogate algorithm has to update the underlying model. The simplicity of the greedy algorithm leads to a CPU time of 850 s (parallelized on 24 cores) to solve the algorithm compared to 2,500 s and 4,000 s for the genetic algorithm and the surrogate model, respectively. We therefore use the greedy algorithm to identify optimal configurations for our actuators.
Additional results generated by the greedy algorithm. In Fig. 19 we present results obtained when the greedy algorithm is programmed to maximize $\theta_{\text{act}}$ (Fig. 19a-c) and $||d||/h$ (Fig. 19d-f). Note that we present present results for three different values of $n_u$ (i.e. $n_u = 1, 2$ and 3).

Then, in Fig. 20 we consider a set of three targets $(T1,T2,T3)$ different from that included in the main text and present the results for the optimal design identified by the greedy algorithm. Next, in Fig. 21 we show the inverse design of an actuator reaching the same set of three targets considered in Fig. 3 of the manuscript, but with the additional constraint that the targets much be reached successively by decreasing pressure.

Finally, in Fig. 22 we show how the minimum value of $\Psi$ found by the greedy algorithm varies with the number of targets, $n_{\text{target}}$, and the units forming a super-cell, $n_u$, and in Fig. 23 how the target radius (i.e. the radius of the sphere fitted with the targets) influences the optimal number of units of the actuator.
Fig. 20. A 6-units actuator reaching three targets. For a specific set of three targets, we use the greedy algorithm to find the actuator design that minimizes $\Psi$, i.e., the error between the targets and the top cap's centroid. Note that we fix $n_u = 3$ and consider $n_s \in \{1, 2, 3, 4, 5\}$. (a) Targets error $\Psi$ as a function of total number of units: the optimal actuator produced by the greedy algorithm for the three targets is reported as (*), along with the respective parameters for each module. The considered set of targets is shown in the inset. (b) State diagram for the 6-units actuator (*) with targets $T_1$, $T_2$, and $T_3$ highlighted. (c) Top and 3D view of the model and the experimental prototype for the 6-units actuator reaching the targets.

Fig. 21. The 12-units actuator with additional constraint. We focus on the same set of three targets considered in Fig. 3 of the manuscript and further impose that each target must be reached successively by decreasing pressure. (a) Targets error $\Psi$ as a function of total number of units: the optimal actuator produced by the greedy algorithm for the three targets is reported as (*), along with the respective parameters for each module. (b) State diagram for the optimal 12-units actuator (*) with targets $T_1$, $T_2$, and $T_3$ highlighted. (c) Top and 3D view of the model for the 12-units actuator reaching the targets.
Fig. 22. Random targets error. For a random set of \( n_{\text{targets}} \), we use the greedy algorithm to find the actuator design that minimizes the target error \( \Psi \). Note that each target is bounded by a cubic domain centered with the actuator and norm equal to \( 1/2(n_u \cdot n_s) \) to ensure that it is within reach. We report the average error based 1000 simulations with \( n_{\text{targets}} \in \{1, 2, 3, 4\} \) and \( n_u \in \{1, 2, 3\} \). Note that for each set of targets, we fix \( n_u \) and choose the \( n_s \) (with the constraint that \( n = n_u \cdot n_s \leq 15 \)) that minimizes the error.

Fig. 23. Optimal number of units as a function of the target radius. For 100 random sets of \( n_{\text{targets}} = 3 \), we measure the target radius, \( R_{\text{targets}} \), i.e. the radius of the sphere fitted with the targets, and we use the greedy algorithm to find the number of units that minimizes the error between the target and the top cap’s centroid. We report here the target radius normalized by the module height, \( R_{\text{targets}}/h \), as a function of the optimal number of units \( n \) found by the greedy algorithm. We find that the average target radius increases with the number of units to minimize the error \( \Psi \).