Nonlinear inflatable actuators for distributed control in soft robots

Bert Van Raemdonck[a], Edoardo Milana[a,c], Michael De Volder[a,b], Dominiek Reynaerts[a], and Benjamin Gorissen[a]

[a]Department of Mechanical Engineering, KULeuven and Flanders Make, Leuven, Belgium; [b]Institute for Manufacturing, University of Cambridge, Cambridge, UK; [c]Department of Microsystems Engineering (IMTEK), University of Freiburg, Freiburg, Germany

As soft robotic systems grow in complexity and functionality, the size and stiffness of the needed control hardware severely limits their application potential. Alternatively, functionality can be embodied within actuator characteristics, drastically reducing the amount of peripherals (1, 2). Functions such as memory (3), computation (4) and energy storage (5) then result from the intrinsic mechanical behavior of precisely designed structures. Here, we introduce actuators with tuneable characteristics to generate complex actuation sequences from a single input. Intricate sequences are made possible by harnessing hysteron characteristics (6) encoded in the buckling of a cone-shaped shell incorporated in the actuator design. A large variety of such characteristics are generated by varying the actuator geometry. We map this dependency and introduce a tool to determine the actuator geometry that yields a desired characteristic. Using this tool, we create a system with six actuators that plays the final movement of Beethoven's Ninth Symphony with a single pressure supply.

Soft robots | Inflatable actuators | Embodied control

In traditional robot architectures, a control signal is generated separately for every actuator (Fig. 1A(i)). The actuators then perform a one-to-one mechanical mapping of their input signal to an output motion. This architecture allows for individual programming of all outputs, but also requires a large number of peripheral components. In inflatable soft robots these components (pumps and valves) are typically stiff and heavy in comparison to the actuators that they control, severely limiting autonomous operation. Therefore, soft and light-weight pumps and valves (7–9) have been developed and incorporated into fluidic circuits that generate complex pressure signals (10–13) However, a large amount of these components is required in a robot with many outputs. The functional complexity of a soft robot with a traditional control architecture is thus limited.

The number of components in a robot architecture can be reduced by encoding part of the functionality into the mechanics of the actuators rather than exclusively in the input signals. This framework of embodied intelligence (14, 15) allows to design robots with an underactuated control scheme where the number of input signals is lower than the number of degrees of freedom required to describe the state of the system. All the actuators share a single control signal but have distinct input-output characteristics designed to produce the desired individual deformation profiles (Fig. 1A(ii)). If all these input-output characteristics perform a one-to-one mapping, however, the sequence in which the actuators reach certain output levels when increasing the control signal is always symmetric to the sequence when the control signal decreases. To introduce complex functionalities, this symmetry needs to be broken. In previous research on inflatable soft robotics, this is done by harnessing viscous damping in the input-output characteristics (16, 17). However, a large amount of damping is needed to create significant asymmetry even when it is amplified by

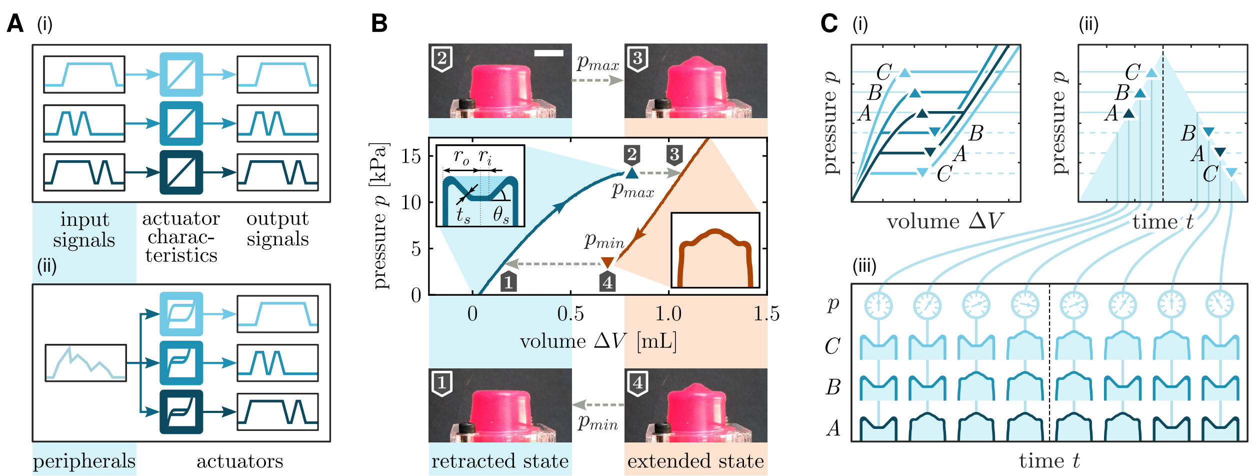

Fig. 1. Underactuating nonlinear actuators. A, Diagram of two robot control architectures. In traditional architectures (i), the control logic generates a unique signal for every actuator. In underactuated architectures (ii), a single control signal driving multiple actuators with distinct input-output characteristics yields the same output signals. **B**, pressure-volume (PV) curve and snapshots of a conical shell actuator. When the pressure is ramped up, the volume changes continuously following the left branch of the PV curve (solid line). For all points on this branch, the shell is retracted (diagram of the actuator cross-section with geometric parameters on the left inset). When the threshold p_{max} (▲) is reached, an unstable snapping transition (dashed arrow) is triggered to the right branch. On that branch, the shell is always extended (right inset) until the pressure is ramped down to p_{min} (▼) and the shell snaps back to the retracted state. The scale bar is 10 mm. **C**, (i) Underactuation of a system of three conical shell actuators labeled A, B and C with different PV curves. (ii) When the common pressure signal is increased and decreased (light blue), the isobaric snapping transitions to the extended and retracted state (▲ and ▼, respectively) are triggered in sequence. (iii) This leads to different combinations of actuator states over time.

dynamic snap-through transitions (18–20), resulting in systems that are impractically slow and energy-inefficient.

Here, we present an underactuated system that can generate asymmetric sequences of discrete actions with high actuation speeds and minimal loss of energy. The sequence asymmetry is generated directly by the actuators that behave like hysterons. This means that each actuator has two stable states and that discrete transitions between both states are triggered when the control signal passes a critical threshold (6, 21). In inflatable soft robotics, several actuators have been proposed with hysteron characterstics that stem from the used hyperelastic materials (20, 22, 23) or from nonlinear geometric effects in the buckling of beams (24), flat sheets (25), singly-curved shells (26–28) or doubly curved shells (5, 29). However the critical thresholds of these solutions are not separately tuneable. To this end, we introduce a novel soft actuator in the shape of a truncated conical shell (cone angle θ_s, shell thickness t_s, outer radius r_o, truncation radius r_i as indicated on the inset image of Fig. 1B) that buckles on inflation to produce a hysteron characteristic. On Fig. 1B, the pressure and volume (PV) inside the actuator are measured during inflation and deflation by an external pressure source. As the actuator evolves from point 1 to point 2, the conical shell deforms continuously but remains retracted in the actuator. At the threshold pressure p_{max} the shell suddenly buckles to point 3 where it extends out of the actuator. During this snapping transition, elastic energy stored in the actuator is released and converted into kinetic energy or useful work. The extended state persists until the actuator is deflated to pressure p_{min} and the shell snaps back from point 4 to the retracted state at point 1.

If multiple conical shell actuators with different snapping thresholds p_{max} and p_{min} are inflated by the same pressure signal, any shells in the retracted state snap to the extended state in order of ascending p_{max} (▲ in order ABC on Fig. 1C). Similarly, on deflation all shells in the extended state snap back to the retracted state in order of descending p_{min} (▼ in order BAC) regardless of whether p_{min} is positive or negative. Therefore, the range of possible sequences is fundamentally determined by the ordering of the snapping thresholds. Solving the inverse problem of determining the snapping thresholds to achieve a certain sequence is akin to optimizing the actuator geometry to obtain a desired deformation (30–32), but with optimization occurring in the PV domain. Here, we will show that conical shell actuators are a predictable platform for designing sequences that are encoded in the snapping thresholds. In the following sections we show how to predict these thresholds for conical shell actuators, how to design a conical shell actuator for a particular combination of snapping thresholds and how to determine the thresholds for multiple actuators that result in a desired sequence. Finally, we introduce an algorithm to determine the thresholds for multiple actuators that result in a desired sequence and we use the developed methods to design a system of six actuators that is able to play Beethoven's "Ode to Joy" on a keyboard with a single controlled pressure supply.

Conical shell actuator modelling

To design underactuated systems for practical applications, it must be possible to design nonlinear actuators for a desired set of p_{max} and p_{min} where these snapping thresholds (i) show a large variety over the design space, (ii) are robust

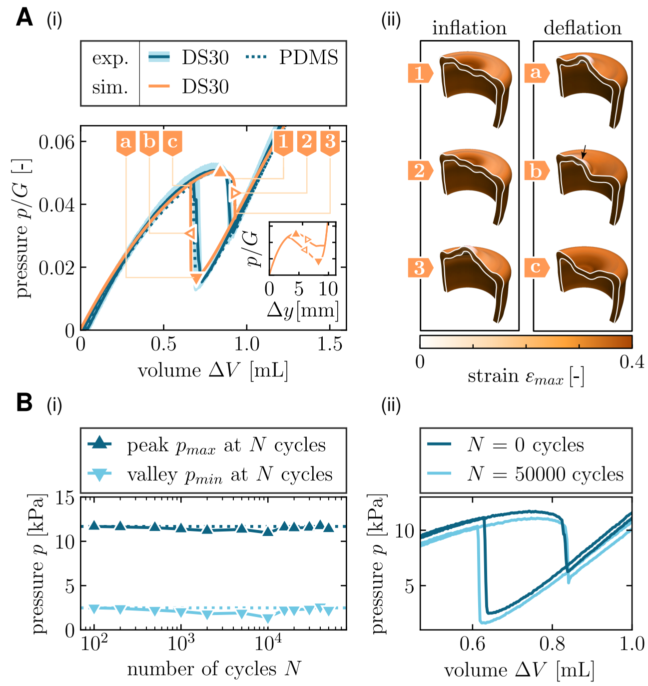

Fig. 2. Conical shell actuator model accuracy. A, (i) Internal pressure when the internal volume of a conical shell actuator (θ_s = 46°, t_s = 1.44 mm) is ramped up and down. Measured (blue) and simulated (orange) pressure-volume (PV) curves with pressure normalized by material shear modulus G. Solid blue curve and shaded blue area are the average and range of the PV curves of five actuators with the same geometry made from Dragonskin 30 (DS30, G=262 kPa). For the dashed curve the material is PDMS (G=701 kPa). Snapping occurs isobarically at thresholds ▲ and ▼ under pressure control and isochorically at ▷ and ◁ under volume control. The inset shows the pressure-displacement curve for the vertical displacement of the shell apex Δy matching the simulated PV curve. (ii) Simulated deformation at the start (1), middle (2) and end (3) of the isochoric snapping transition on inflation and deflation ((a), (b) and (c), respectively). ϵ_{max} is the maximal principal logarithmic strain and reaches a maximum value of 0.37 at the black arrow in (b). **B**, (i) Measured isobaric snapping thresholds p_{max} and p_{min} after N repeated loading cycles for an actuator with the same geometry as in subfigure A. The dashed lines are the values of the snapping thresholds for $N = 0$. (ii) Section of the PV curve in the first and the 50 000-th loading cycle.

to imperfections in the manufacturing process and (iii) remain constant over a large numbers of actuation cycles. In the case of inflatable actuators with hysteron characteristics, literature (22, 23, 26, 33) offers no designs that meet these requirements. For our conical shell actuator however, it is possible to accurately predict its p_{max} and p_{min} in all these cases by means of a Finite Element Model (see S4). To quantify the accuracy of this model, we perform a simulation and manufacture an actuator with the same geometry ($\theta_s = 46°$, $t_s = 1.44$ mm, $r_o = 10$ mm, $r_i = 2$ mm) and material (Dragon Skin 30, Smooth-on) as the simulation (see S2). Next, a volumetric flow source slowly increases and decreases the internal volume of the actuator V while the resulting pressure p is monitored for both the manufactured actuator (see S3 and Supplemental Video 1) and the model (see S4.5 and Supplemental Video 2). Compared to the case where pressure is controlled (Fig. 1B), the resulting pressure-volume (PV) characteristic is identical until p_{max} and p_{min} are reached (▲ and ▼). No snapping occurs at these points but instead the pressure goes through a local extremum. Snapping is delayed until the actuator reaches the isochoric snapping threshold and undergoes a rapid volume-preserving motion (▷ and ◁) typical for the buckling of deep shells (34, 35). For the considered actuator, this transition between the retracted and extended state is symmetric during inflation but asymmetric during deflation (Fig. 2A(ii)). In all cases, however, the vertical displacement of the shell's apex Δy and the internal pressure in the actuator suddenly change, where the relation between these two variables is given by the inset on Fig. 2A(i).Compared to previous studies on the inflation of spherical (35) or spherical-derived (36) shells where the isobaric and isochoric snapping points coincide, we find for our actuators an intermediate characteristic in which both snapping threshold are separated. This leads to a reduced sensitivity of p_{min} and especially p_{max} to random imperfections in the geometry (see S4.2). We verify this result by comparing an idealized simulation to measurements for five actuators manufactured from the same mold, as shown on Fig. 2A(i). The simulated values for the isobaric snapping thresholds p_{max} (13.29 kPa) and p_{min} (4.28 kPa) agree well with the values averaged over the manufactured actuators (13.30 kPa and 3.93 kPa) and for the isochoric snapping thresholds under volume control the maximum modelling error is limited as well (0.04 mL). Moreover, the standard deviation on p_{max} (0.28 kPa) and p_{min} (0.31 kPa) for the five PV curves is small so random variations in the manufacturing process have limited impact on the snapping thresholds.

The same finite element model also accurately captures the PV curve of actuators with the same geometry but a different overall size or material. Dimensional analysis shows that V/r_o^3 is a dimensionless group that captures geometric scaling and p/G is a dimensionless group that captures changes in the material. In general, the material behavior of silicone rubbers is described by more than one parameters such that p/G is not constant for different materials. In the conical shell actuator shown in Fig. 2A(ii), however, snapping occurs at small strains with a maximum strain of 0.37 throughout an actuation cycle. Consequently, the material behavior can be captured by a single material parameter G and p/G is constant for different materials (see S4.4). This is illustrated in Fig. 2A(i), where we also plot the normalized results of a PDMS actuator. For normalisation we use $G = 262$ kPa for Dragon Skin 30 (9) and $G = 701$ kPa for PDMS (fitted on the initial slope of the PV curve). Another advantage of the low material strain is that material degradation on the snapping thresholds remains limited throughout the lifetime of the actuator. This is confirmed by an experiment where the previously discussed conical shell actuator is subjected to 50 000 loading cycles and the snapping thresholds are measured periodically (see S3.4). As shown in Fig. 2B, the shift in the snapping thresholds remains relatively small and shows no consistent downward trend as the number of cycles increases. This means that a single simulation suffices to predict the behavior of a conical shell actuator across a range of materials, length scales and loading cycles.

Design for actuator characteristics

The accuracy of the numerical model allows us to develop a mapping that directly translates desired values for p_{max} and p_{min} into an actuator design that embodies these snapping thresholds. To this end, we repeat the simulations for actuators with r_i and r_o fixed to 2 mm and 10 mm while θ_s and t_s are sampled uniformly from the rectangular domain $[20°, 55°] \times [0.5\,\text{mm}, 2\,\text{mm}]$ with steps of 2.5° and 0.1 mm. We report all variables in dimensionless groups (θ_s, t_s/r_o, p/G, V/r_o^3) such

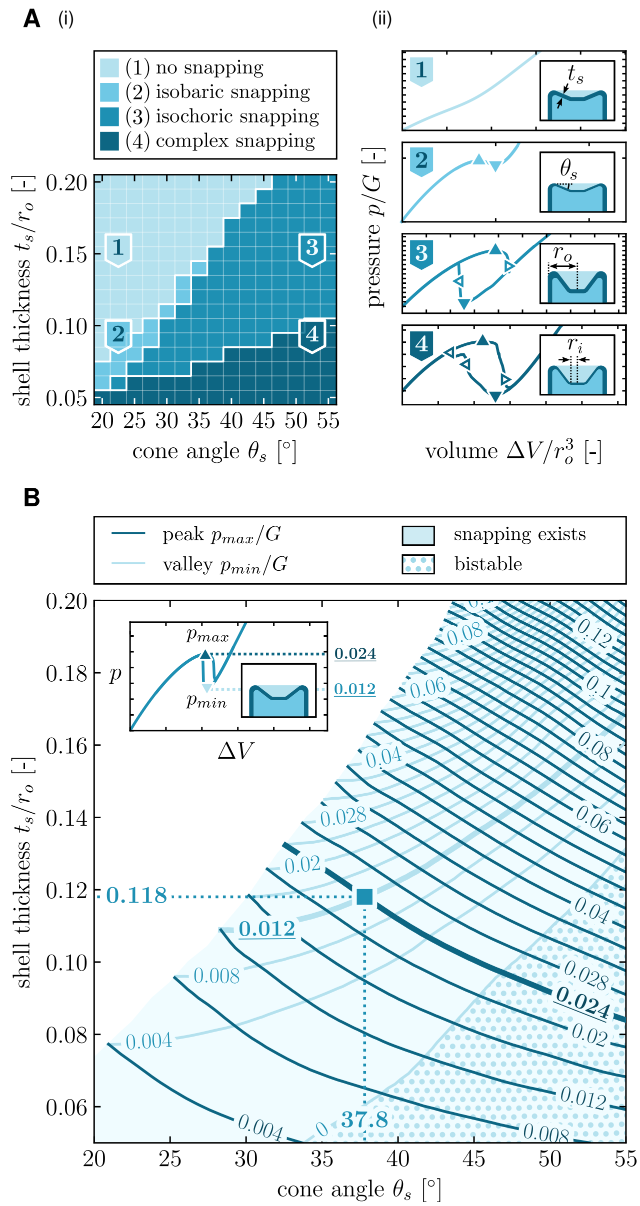

Fig. 3. Shaping pressure-volume curves. A, Qualitative dependence of simulated conical shell actuator PV curves on cone angle θ_s and normalized shell thickness t_s/r_o. (i) The curves are classified in four categories: monotonic curves without snapping, curves with an isobaric snapping pair, curves with a single additional isochoric snapping pair and curves with multiple additional isochoric snapping pairs. (ii) PV curve samples from every category. Markers 1-4 in the (θ_s,t_s/r_o)-plane indicate the corresponding geometric parameters and the insets show the resulting actuator geometry. ▲ and ▼ indicate isobaric snaps on inflation and deflation, and ▶ and ◀ indicate isochoric snapping on inflation and deflation, respectively. **B**, Selection chart consisting of contours connecting points in the (θ_s,t_s/r_o)-plane that yield the same normalized peak (p_{max}/G) or valley (p_{min}/G) pressure.For actuators in the dotted area, $p_{min} < 0$, meaning that they are bistable at atmospheric pressure ($p = 0$). The contours of $p_{max}/G = 0.024$ and $p_{min}/G = 0.012$ are highlighted. The inset shows the PV curve with the corresponding peak and valley pressure. This curve corresponds to a conical shell actuator with θ_s = 37.8° and $t_s/r_o = 0.118$, which are the coordinates of the intersection point of the highlighted contours.

that the results can be universally adopted regardless of the scale and material required for a particular application.

The shape of the PV curve changes drastically across the sampled domain (Fig. 3A(i)). As θ_s increases and t_s/r_o decreases, the number of possible snapping transitions in the PV curves increases. We track this evolution by separating the PV curves in four categories with different numbers of snapping thresholds (threshold detection algorithm in S5), each exemplified by a sample actuator 1-4 in Fig. 3A(ii). The category of sample 1 features no snapping thresholds and occurs in the part of the domain where θ_s is low and t_s/r_o is high. Near the border of this region, the PV curve develops an inflection point which eventually gives rise to a local maximum (▲) and minimum (▼) in pressure. For low angles, this leads to PV curves like sample 2 with only isobaric snapping thresholds. This category only occupies a small part of the domain, as the occurrence of isobaric snaps is quickly followed by the development of a pair of isochoric snaps like in sample 3 (inflation: ▷, deflation: ◁). While for the first two categories the deformation of the shell always remains axisymmetric, in this third category at least one of the isochoric snaps is accompanied by a transient asymmetric motion of the shell during which the cone tilts (Fig. 2B(ii)). For the deepest and thinnest shells, one or multiple of these transient asymmetric configurations can become stable. This leads to complex PV curves like sample 4, where multiple consecutive snaps occur during volume-controlled inflation and deflation.

Even though the shape of the PV curve changes drastically across the sampled design parameter space, the main features of interest follow consistent trends. On the one hand, p_{max}/G increases monotonically with both θ_s and t_s/r_o, ranging from 0.002 to 0.142 over the sampled domain. p_{min}/G, on the other hand, decreases with θ_s and increases with t_s/r_o, covering values between -0.003 and 0.101. Negative values for p_{min} correspond to actuators that are bistable at atmospheric pressure and occur when $\theta_s^2 r_o/t_s > 7.1$ (θ_s expressed in radians). In order to harness these data to design actuators with a particular set of snapping thresholds, we introduce the selection chart shown on Fig. 3B. This chart is the superposition of the contours of p_{max}/G and p_{min}/G in (θ_s, t_s/r_o)-space (see S6.1). In order to find the geometric parameters θ_s and t_s that lead to a certain set of snapping thresholds, it suffices to identify the intersection point between the contours of p_{max}/G (dark lines) and p_{min}/G (light lines) that match the desired values. The horizontal and vertical coordinate then correspond to t_s/r_o and θ_s, respectively, that generate a characteristic with the desired p_{max} and p_{min}. As an example, Fig. 3B shows that for an actuator with $p_{max}/G = 0.024$ and $p_{min}/G = 0.012$, the contours intersect at $\theta_s = 37.8^\circ$ and $t_s/r_o = 0.118$, which are the geometric parameters that yield the characteristic shown on the inset. The resulting tool is valid regardless of the envisioned material or scale of the actuator, making Fig. 3B universally valid for the design of conical shell actuators.

In addition to pressures, contours of other key actuator characteristics (volume or displacement) can be plotted in (θ_s, t_s/r_o)-space as well (see S6.2). Superimposing any two sets of contours allows to find (θ_s, t_s/r_o) for any pair of desired characteristics, with G and r_o as a given. Moreover, if G and r_o can be chosen freely, up to four characteristics can be inversely designed for using adapted selection charts (see S6.3). While these design procedures generalize to other actuators, the conical shell actuator is uniquely suited for applications involving embodied intelligence since (i) the achievable range of (p_{max}, p_{min}) or any other combination of characteristics is larger (see Fig. 4D), (ii) the modelling error on the isobaric snapping thresholds is small ($< 0.0036\,G$ for the measurements in 2A), and (iii) the variation in those thresholds remains small over the lifespan of the actuator ($< 0.0042\,G$ for the measurements in 2B).

Design for embodied system characteristics

Precise control over nonlinear actuator characteristics enables the design of systems with embodied intelligence for a specific functionality. To demonstrate this, we use Fig. 3B to design an underactuated system consisting of six actuated degrees of freedom that can play the 62 note sequence that constitutes Beethoven's "Ode to Joy" (Fig. 4A) on a piano keyboard with a single pressure supply line (Fig. 4B). The supply line provides compressed air with varying pressure $p_c(t)$ and the system consists of six conical shell actuators in an underactuated architecture as in Fig. 1A(ii). Each of the six actuators is placed above a dedicated key of a piano (D_4, G_4, A_4, B_4, C_5 and D_5) such that the corresponding note is played if p_{max} of the actuator is exceeded by $p_c(t)$. Designing an underactuated system that can play "Ode to Joy" thus consist of tuning six actuator snapping thresholds (p_{max}, p_{min}) in addition to generating the continuous pressure signal $p_c(t)$ that needs to be applied to generate the melody. This design process consist of a qualitative step involving a symbolic algorithm and a quantitative step involving Fig. 3B.

In the qualitative step, we abstract both the song and the physical system playing it. First, we reduce "Ode to Joy" to a sequence of key presses represented by the combinatorial word $B_4 B_4 C_5 D_5 D_5 C_5 B_4 A_4 \ldots$. As such, we only consider the relative order of the key presses and disregard the pitch, duration and spacing of the associated notes. Second, we abstract the operation of each pair of a piano key and its matching actuator as a two-state hysteron (Fig. 4C) In the first state, the conical shell is retracted and the piano key is up and in the other state the conical shell is extended and the piano key is down. The transitions between these states correspond to the isobaric snapping transitions at pressures p_{max} and p_{min}. While in most systems of hysterons the output is determined by the static states of the hysterons, a piano only produces a note when a key transitions from the up to the down state, after which the note quickly fades out. In our case, the Finite State Machines that describe the hysterons are therefore of the Mealy type (37), where playing notes is associated with the snapping transitions at p_{max}. The sequence in which these transitions can occur and hence the range of melodies that an underactuated system of actuators can play is fully determined by the relative order of the actuators p_{max} on the one hand, and the order of their p_{min} on the other hand. In S7.2, we formalize this relation as a set of constraints on the playable sequences for a certain system. By harnessing these constraints, we develop an algorithm that efficiently prunes the search space of all possible orders of p_{max} and p_{min} to find the combination that can play a given sequence (S7.3). Beyond playing melodies, this algorithm can be used to design a system of hysterons for any task that can be described by a Finite State Machine of the Mealy type that satisfies return point memory (38). For the melody of "Ode to Joy", the algorithm only needs to check 24 of the $(6!)^2 = 518{,}400$ possible designs to find that the system with $G_4 < A_4 < D_4 < B_4 < C_5 < D_5$ for p_{max} and $D_4 < G_4 < A_4 < B_4 < C_5 < D_5$ for p_{min} is able to play the melody. While this solution is unique for "Ode

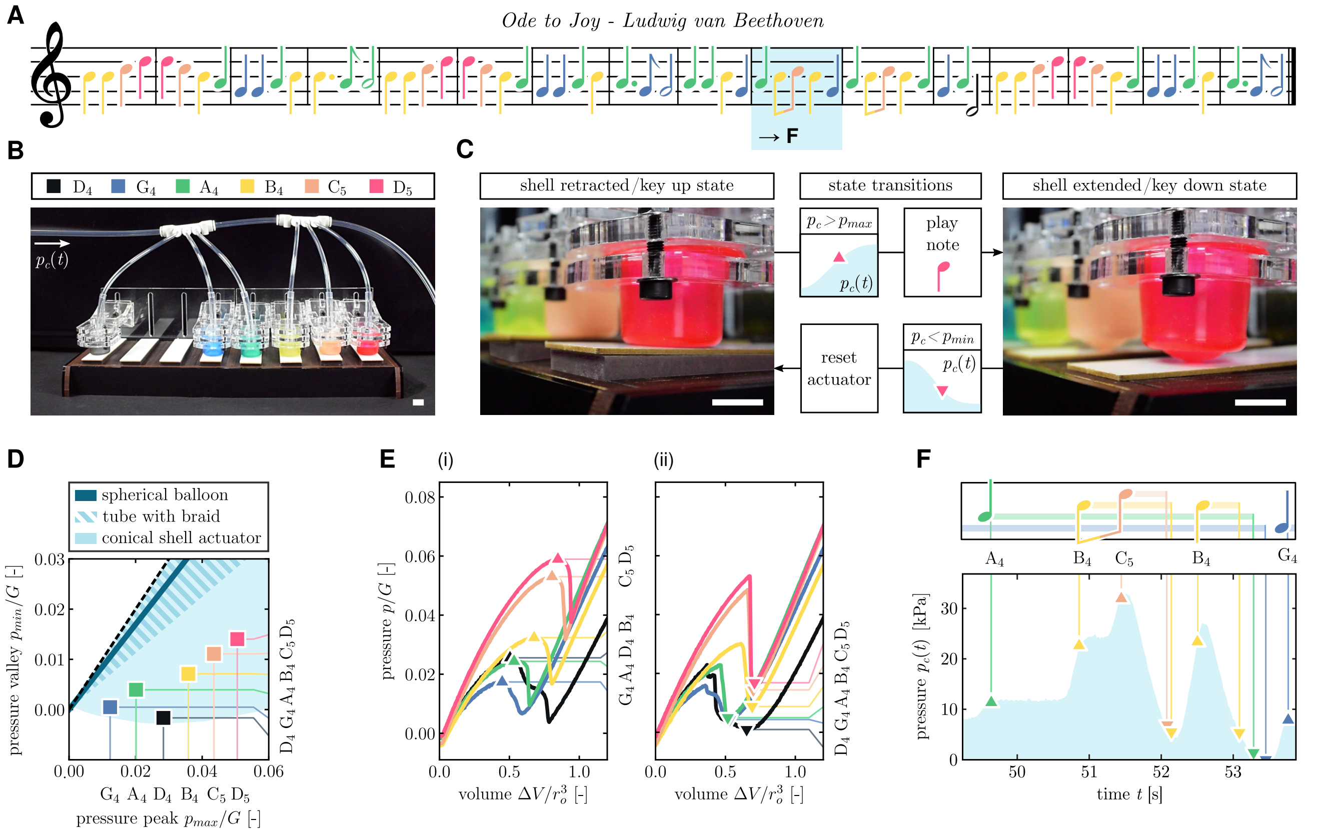

Fig. 4. Underactuated piano playing. **A**, "Ode to Joy" is a melody of 62 notes containing six different tones. **B**, Demonstrator that plays "Ode to Joy" on a keyboard with six conical shell actuators and a common pressure signal $p_c(t)$ generated by a single valve. Each actuator has a unique set of geometric parameters and a unique color. These colors carry over to the other subfigures to refer to the matching notes or PV curves. **C**, Close-up of the D_5 actuator playing a note. If the actuator is in the retracted state (left) and $p_c(t)$ exceeds p_{max} (▲) of the actuator, the actuator plays a note as it snaps to the extended state (right). The actuator stays in that state until $p_c(t)$ drops below p_{min} (▼) and it resets without playing a note. **D**, Design values for p_{max} and p_{min} of the six keyboard actuators plotted in the Preisach plane. The order of the markers on the horizontal and vertical axis correspond to the s_{max} and s_{min} from the sequence design algorithm, respectively. The markers are overlaid on the regions of the Preisach plane that can be reached by spherical balloons (33), tubes with braids (22) and conical shell actuators. **E**, Experimental PV curves of the keyboard actuators measured in inflation (i) and deflation (ii) with orderings s_{max} and s_{min} as designed for in subfigure D. **F**, Bottom: measured pressure signal $p_c(t)$ on playing the part of the sequence highlighted in subfigure A. ▲ and ▼ mark transitions from the low- to the high-volume state and back again, respectively. Top: timing of the notes played by the actuators where colored horizontal lines indicate the time intervals when they are in the high-volume state. Scale bars on photos are 10 mm

to Joy", for some melodies multiple solutions exist while for others no solution exists (see S7.5).

In the quantitative step, we match every note to a conical shell actuator design. The values for the actuators p_{max} and p_{min} are free to choose as long as their relative order is maintained and they lie within the range of achievable (p_{max}, p_{min}). For the conical shell actuator, this range is much larger than for other nonlinear actuators presented in literature. (Fig. 4D). Moreover, for the demonstrator we take into account the required stroke to actuate the piano keys as well as the robustness of the snapping threshold to imperfections in the manufactured geometry (see S8). Before using the selection chart, we select $r_o = 10\,\mathrm{mm}$ for all actuators to fit over the piano keyboard. To limit the influence of the pressing forces on the snapping thresholds of the actuator, we minimize the stiffness of the piano keys and maximize the stiffness of the actuators by selecting PDMS ($G = 701\,\mathrm{kPa}$) as their material. With the selected values for p_{max}, p_{min}, r_o and the material G, we use the selection chart (Fig. 3B) to find the geometric parameters of the actuators. We then manufacture six actuators with the identified geometry (see S2) and measure their PV characteristic experimentally (see S3). As shown on Fig. 4E, the ordering of p_{max} and p_{min} is as designed for, so the ability to play "Ode to Joy" is hard coded into the design of the actuators. The discrepancies between the measured and desired values for p_{max} and p_{min} are largely due to variations in the material characteristics of PDMS, which are known to be very sensitive to manufacturing parameters such as the mixing ratio and the curing procedure.

To play the melody, an electronic proportional valve generates $p_c(t)$ as a stepwise signal between the measured actuator snapping thresholds. This is schematically shown in Fig. 4F which zooms in on a 5-note section of the music piece $A_4B_4C_5B_4G_4$. As the values for p_{max} of A_4, B_4, C_5 are in ascending order and all actuators are in the retracted state at the start of the section, playing the first three notes can be done by consecutively exceeding their p_{max}. Playing B_4 again first requires a reset by lowering the pressure below its p_{min}. In this case the pressure must stay above p_{min} of G_4 as it has a lower p_{max} but can not be played at this point in the sequence (see S7.4). After playing B_4, however, the next note to be played is G_4 so then the pressure can drop below its p_{min}, resetting actuators B_4, A_4 and G_4 in that order. Note that here we do not reset C_5, as it was already reset together with resetting B_4 for the first time. Because of the high bandwidth of the underactuation mechanism, the minimal time interval between subsequent notes is limited by the time needed for the pressure controller to reach the required snapping thresholds, which scales with the difference in pressure between them. With the length of an eighth note set to the smallest possible time interval attainable by the pressure controller, the resulting sequence of snaps translates in a faithful rendition of “Ode to Joy” on the piano (Supplemental Video 3).

For a given set of conical shell actuators, only a limited number of sequences can be achieved, as this is mechanically encoded in the snapping pressures of the actuators. One way to expand the range of possible sequences is to harness the the viscoelasticity of the actuators or the pressure drop in the tubes that connect them to the pressure source. Both effects cause a time delay between the pressure input and the deformation of the actuator such that the sequence depends on the timescale of the pressure input (16, 17, 39). Another approach is to manufacture the actuators out of functional materials such that their thresholds can be tuned by external electric (40) or magnetic (41) fields. Finally, the snapping thresholds also depend on the external forces acting on each actuator (see S3.3, S5.3 and S6.2). In the demonstrator, the low stiffness of the piano keys limits this effect, but in other applications external forces can modify the order of the thresholds and the resulting sequence. On the one hand, the pressure controller then needs to keep track of all previous internal and external inputs to the system to accurately predict the sequence. On the other hand, this property can be harnessed to deliberately change the sequence in response to inputs from the environment without electronic control (11). Despite these limitations, the proposed design strategy can be applied for other applications regardless of the desired material, size or deformation mode of the actuator. As shown by the dimensional analysis, changing the material and the size of the actuator only scales all PV curves within the scale limits of continuum mechanics. In practice, a large range of materials can be used because the low strains during actuation put less strong constraints on the ultimate strain than most nonlinear inflatable actuators. Regarding its size, the monolithic and low-complexity geometry of the actuator is compatible with a wide range of manufacturing techniques including lithography, which allows miniaturization. In order to achieve different deformation modes, it is possible to use the conical shell actuator as a valve (9) or pump (36) that drives a secondary actuator. In addition, future research can be directed towards using multiple conical shell actuators as building blocks to shape compound actuators where the snapping thresholds and the relative orientation of the building blocks determine the global deformation sequence of the compound actuator. For instance, a serial stacking of multiple conical shells, eccentrically positioned with respect to the neutral axis of a structure, can instigate a bending deformation when snapped. In all these cases, the design tools for shaping the nonlinear characteristic remain valid and the actuation mode is only determined by the design of the compound actuator. The presented actuator and tuning methods therefore form a universal building block for inflatable systems with embodied intelligence.

References

1. Aniket Pal, Vanessa Restrepo, Debkalpa Goswami, and Ramses V Martinez. Exploiting mechanical instabilities in soft robotics: control, sensing, and actuation. *Advanced Materials*, 33(19):2006939, 2021.
2. Cameron A Aubin, Benjamin Gorissen, Edoardo Milana, Philip R Buskohl, Nathan Lazarus, Geoffrey A Slipher, Christoph Keplinger, Josh Bongard, Fumiya Iida, Jennifer A Lewis, et al. Towards enduring autonomous robots via embodied energy. *Nature*, 602(7897):393–402, 2022.
3. Tian Chen, Mark Pauly, and Pedro M Reis. A reprogrammable mechanical metamaterial with stable memory. *Nature*, 589(7842):386–390, 2021.
4. Daniel J Preston, Philipp Rothemund, Haihui Joy Jiang, Markus P Nemitz, Jeff Rawson, Zhigang Suo, and George M Whitesides. Digital logic for soft devices. *Proceedings of the National Academy of Sciences*, 116(16):7750–7759, 2019.
5. Benjamin Gorissen, David Melancon, Nikolaos Vasios, Mehdi Torbati, and Katia Bertoldi. Inflatable soft jumper inspired by shell snapping. *Science Robotics*, 5(42):eabb1967, 2020. . URL https://www.science.org/doi/abs/10.1126/scirobotics.abb1967.
6. Ferenc Preisach. Über die magnetische nachwirkung. *Zeitschrift für physik*, 94(5):277–302, 1935.
7. V. Cacucciolo, J. Shintake, Y. Kuwajima, S. Maeda, D. Floreano, and H. Shea. Stretchable pumps for soft machines. *Nature*, 572(7770):516–+, 2019. ISSN 0028-0836. . URL <GotoISI>://WOS:000482219600043.
8. R. S. Diteesawat, T. Helps, M. Taghavi, and J. Rossiter. Electro-pneumatic pumps for soft robotics. *Science Robotics*, 6(51), 2021. ISSN 2470-9476. . URL <GotoISI>://WOS:000624493500003.
9. Philipp Rothemund, Alar Ainla, Lee Belding, Daniel J Preston, Sarah Kurihara, Zhigang Suo, and George M Whitesides. A soft, bistable valve for autonomous control of soft actuators. *Science Robotics*, 3(16):eaar7986, 2018.
10. Michael Wehner, Ryan L. Truby, Daniel J. Fitzgerald, Bobak Mosadegh, George M. Whitesides, Jennifer A. Lewis, and Robert J. Wood. An integrated design and fabrication strategy for entirely soft, autonomous robots. *NATURE*, 536(7617):451+, AUG 25 2016. ISSN 0028-0836. .
11. Dylan Drotman, Saurabh Jadhav, David Sharp, Christian Chan, and Michael T. Tolley. Electronics-free pneumatic circuits for controlling soft-legged robots. *Science Robotics*, 6(51): eaay2627, 2021. . URL https://www.science.org/doi/abs/10.1126/scirobotics.aay2627.
12. Joshua D. Hubbard, Ruben Acevedo, Kristen M. Edwards, Abdullah T. Alsharhan, Ziteng Wen, Jennifer Landry, Kejin Wang, Saul Schaffer, and Ryan D. Sochol. Fully 3d-printed soft robots with integrated fluidic circuitry. *Science Advances*, 7(29):eabe5257, 2021. . URL https://www.science.org/doi/abs/10.1126/sciadv.abe5257.
13. Shane Hoang, Konstantinos Karydis, Philip Brisk, and William H. Grover. A pneumatic random-access memory for controlling soft robots. *PLOS ONE*, 16(7):1–25, 07 2021. . URL https://doi.org/10.1371/journal.pone.0254524.
14. Rolf Pfeifer and Josh Bongard. How the Body Shapes the Way We Think. In *How Body Shapes W. We Think*. The MIT Press, 2006. . URL https://direct.mit.edu/books/book/2035/chapter/55234/how-the-body-shapes-the-way-we-think.
15. V. C. Muller and M. Hoffmann. What is morphological computation? on how the body contributes to cognition and control. *Artificial Life*, 23(1):1–24, 2017. ISSN 1064-5462. . URL <GotoISI>://WOS:000395587600001.
16. N. Vasios, A. J. Gross, S. Soifer, J. T. B. Overvelde, and K. Bertoldi. Harnessing viscous flow to simplify the actuation of fluidic soft robots. *Soft Robotics*, pages 1–9, 2019. ISSN 2169-5172. .
17. A. Di Lallo, M. G. Catalano, M. Garabini, G. Grioli, M. Gabiccini, and A. Bicchi. Dynamic morphological computation through damping design of soft continuum robots. *Frontiers in Robotics and Ai*, 6, 2019. ISSN 2296-9144. . URL <GotoISI>://WOS:000464539500001.
18. D. Glozman, N. Hassidov, M. Senesh, and M. Shoham. A self-propelled inflatable earthworm-like endoscope actuated by single supply line. *IEEE TBME*, 57(6):1264–1272, June 2010. .
19. Benjamin Gorissen, Edoardo Milana, Arne Baeyens, Eva Broeders, Jeroen Christiaens, Klaas Collin, Dominiek Reynaerts, and Michael De Volder. Hardware sequencing of inflatable nonlinear actuators for autonomous soft robots. *Advanced Materials*, 31(3):1804598, 2019.
20. Eran Ben-Haim, Lior Salem, Yizhar Or, and Amir D. Gat. Single-input control of multiple fluid-driven elastic actuators via interaction between bistability and viscosity. *Soft Robotics*, 7 (2):259–265, 2020. .
21. Hadrien Bense and Martin van Hecke. Complex pathways and memory in compressed corrugated sheets. *Proceedings of the National Academy of Sciences*, 118(50):e2111436118, 2021.
22. Johannes T. B. Overvelde, Tamara Kloek, Jonas J. A. D'haen, and Katia Bertoldi. Amplifying the response of soft actuators by harnessing snap-through instabilities. *PNAS*, 2015. .
23. Edoardo Milana, Bert Van Raemdonck, Andrea Serrano Casla, Michael De Volder, Dominiek Reynaerts, and Benjamin Gorissen. Morphological control of cilia-inspired asymmetric movements using nonlinear soft inflatable actuators. *Frontiers in Robotics and AI*, 8:413, 2022. ISSN 2296-9144. . URL https://www.frontiersin.org/article/10.3389/frobt.2021.788067.
24. Thomas George Thuruthel, Syed Haider Abidi, Matteo Cianchetti, Cecilia Laschi, and Egidio Falotico. A bistable soft gripper with mechanically embedded sensing and actuation for fast grasping. In *2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pages 1049–1054. IEEE, 2020.
25. Won-Kyu Lee, Daniel J Preston, Markus P Nemitz, Amit Nagarkar, Arthur K MacKeith, Benjamin Gorissen, Nikolaos Vasios, Vanessa Sanchez, Katia Bertoldi, L Mahadevan, et al. A buckling-sheet ring oscillator for electronics-free, multimodal locomotion. *Science Robotics*, 7 (63):eabg5812, 2022.
26. Aniket Pal, Debkalpa Goswami, and Ramses V. Martinez. Elastic energy storage enables rapid and programmable actuation in soft machines. *Advanced Functional Materials*, 30(1): 1906603, 2020. . URL https://onlinelibrary.wiley.com/doi/abs/10.1002/adfm.201906603.
27. Ofek Peretz, Anand K Mishra, Robert F Shepherd, and Amir D Gat. Underactuated fluidic

control of a continuous multistable membrane. *Proceedings of the National Academy of Sciences*, 117(10):5217–5221, 2020.
28. Haneol Lee and Hugo Rodrigue. Harnessing the nonlinear properties of buckling inflatable tubes for complex robotic behaviors. *Materials Today*, 2023.
29. David Melancon, Antonio Elia Forte, Leon M Kamp, Benjamin Gorissen, and Katia Bertoldi. Inflatable origami: Multimodal deformation via multistability. *Advanced Functional Materials*, 32(35):2201891, 2022.
30. Trevor J. Jones, Etienne Jambon-Puillet, Joel Marthelot, and P. T. Brun. Bubble casting soft robotics. *Nature*, 599(7884):229–233.
31. Fionnuala Connolly, Conor J. Walsh, and Katia Bertoldi. Automatic design of fiber-reinforced soft actuators for trajectory matching. *Proceedings of the National Academy of Sciences*, 114 (1):51–56, 2017. . URL https://www.pnas.org/doi/abs/10.1073/pnas.1615140114.
32. Robert Baines, Sree Kalyan Patiballa, Benjamin Gorissen, Katia Bertoldi, and Rebecca Kramer-Bottiglio. Programming 3d curves with discretely constrained cylindrical inflatables. *Advanced Materials*, n/a(n/a):2300535. . URL https://onlinelibrary.wiley.com/doi/abs/10.1002/adma.202300535.
33. Alan Needleman. Inflation of spherical rubber balloons. *International Journal of Solids and Structures*, 13(5):409–421, 1977.
34. WT Koiter. Nonlinear buckling problem of a complete spherical shell under uniform external pressure. i. *Proceedings of the Koninklijke Akademie van Wetenschappen Series B - Physical sciences*, 72(1):40, 1969.
35. John W Hutchinson. Buckling of spherical shells revisited. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 472(2195):20160577, 2016.
36. Chuan Qiao, Lu Liu, and Damiano Pasini. Bi-shell valve for fast actuation of soft pneumatic actuators via shell snapping interaction. *Advanced Science*, 8(15):2100445, 2021. .
37. George H Mealy. A method for synthesizing sequential circuits. *The Bell System Technical Journal*, 34(5):1045–1079, 1955.
38. Muhittin Mungan and M. Mert Terzi. The structure of state transition graphs in systems with return point memory: I. general theory. *ANNALES HENRI POINCARE*, 20(8):2819–2872, AUG 2019. ISSN 1424-0637. .
39. Yoav Matia, Gregory H. Kaiser, Robert F. Shepherd, Amir D. Gat, Nathan Lazarus, and Kirstin H. Petersen. Harnessing nonuniform pressure distributions in soft robotic actuators. *Advanced Intelligent Systems*, 5(2):2200330, 2023. . URL https://onlinelibrary.wiley.com/doi/abs/10.1002/aisy.202200330.
40. Tiefeng Li, Christoph Keplinger, Richard Baumgartner, Siegfried Bauer, Wei Yang, and Zhigang Suo. Giant voltage-induced deformation in dielectric elastomers near the verge of snap-through instability. *Journal of the Mechanics and Physics of Solids*, 61(2):611–628, 2013.
41. Arefeh Abbasi, Tomohiko G Sano, Dong Yan, and Pedro M Reis. Snap buckling of bistable beams under combined mechanical and magnetic loading. *Philosophical Transactions of the Royal Society A*, 381(2244):20220029, 2023.
42. Luc Marechal, Pascale Balland, Lukas Lindenroth, Fotis Petrou, Christos Kontovounisios, and Fernando Bello. Toward a common framework and database of materials for soft robotics. *Soft Robotics*, 8(3):284–297, 2021. . URL https://doi.org/10.1089/soro.2019.0115.
43. Frank L. Roth, Raymond L. Driscoll, and William L. Holt. Frictional properties of rubber. *Rubber Chemistry and Technology*, 16:155–177, 1942.
44. *Combinatorics on Words.* Cambridge Mathematical Library. Cambridge University Press, 2 edition, 1997. .
45. Peter van Beek. Chapter 4 - backtracking search algorithms. In Francesca Rossi, Peter van Beek, and Toby Walsh, editors, *Handbook of Constraint Programming*, volume 2 of *Foundations of Artificial Intelligence*, pages 85–134. Elsevier, 2006. . URL https://www.sciencedirect.com/science/article/pii/S1574652606800088.
46. Sally C. Brailsford, Chris N. Potts, and Barbara M. Smith. Constraint satisfaction problems: Algorithms and applications. *European Journal of Operational Research*, 119(3):557–581, 1999. ISSN 0377-2217. . URL https://www.sciencedirect.com/science/article/pii/S0377221798003646.

Acknowledgements Research was supported by the Fund for Scientific Research-Flanders (FWO) (11A4821N).

Author contributions E.M., B.G., M.D.V. and D.R. conceived the project. B.V.R. conceived the actuator concept, conducted the simulations and experiments and developed the sequence design algorithm. B.V.R., B.G. and E.M. analyzed the data. All authors contributed to the manuscript.

Competing interests The authors declare no conflict of interest

Supplementary data

Supplementary information document

Supplementary Video 1 Experimental actuator characterization. Quasi-static volume controlled inflation and deflation of the D_5 conical shell actuator at a flow rate of $50\,\mu\text{L/s}$. The pressure-volume curve on the right features the data measured during the experiment shown on the video. The video is sped up by a factor three. During the quasi-static experiment, rapid isochoric snapping transitions occur. These are shown in slow motion where the video is slowed down by a factor 0.075. In all frames, the scale bars are 10 mm.

Supplementary Video 2 Numerical actuator characterization. Finite-element model of the volume controlled inflation and deflation of the D_5 actuator side-by-side with the simulated pressure-volume characteristic. Because of the adaptive time stepping in the solver, the simulation time between any two consecutive frames is not constant. This leads to a slower apparent motion of the conical shell in the neighborhood of the isochoric snapping transitions that highlights the sequence of symmetry-breaking deformation modes.

Supplementary Video 3 Underactuated piano demonstrator. Inflation of six tuned conical shell actuators with a predefined pressure signal to play Beethoven's "Ode to Joy" on a piano keyboard. The pressure is controlled by an off-screen proportional valve, distributed through the transparent tubing and measured with a manometer and an off-screen pressure transducer. The monitor behind the setup displays the measured pressure signal in function of time and shows a colored rectangle whenever a piano key registers contact. The video is sped up by a factor 2.5.

Supplemental information to

Nonlinear inflatable actuators for distributed control in soft robots

Bert Van Raemdonck[a], **Edoardo Milana**[a,c], **Michael De Volder**[a,b], **Dominiek Reynaerts**[a], **and Benjamin Gorissen**[a]

[a]Department of Mechanical Engineering, KULeuven and Flanders Make, Leuven, Belgium; [b]Institute for Manufacturing, University of Cambridge, Cambridge, UK; [c]Department of Microsystems Engineering (IMTEK), University of Freiburg, Freiburg, Germany

S1. Actuator geometry

The conical shell actuator consists of two main functional segments joined with a compliant hinge: a conical shell and a cylindrical wall. The function of the conical shell is to convert an increase in volume into a useful deformation and to generate the peaks and valleys in the PV curve depending on the angle θ_s and thickness t_s of the cone (Fig. S1). In comparison to a spherical shell, the conical shape has two main advantages. The first advantage is that for a given outer radius r_o of the actuator, the maximum depth of a spherical shell is limited to r_o, limiting the stroke. For a conical shell, there is no geometric constraint on the depth or stroke which can take arbitrary values as θ_s asymptotically approaches $\pi/2$. The second advantage is that for a conical shell the isobaric snapping threshold on inflation occurs at a smooth maximum in the PV curve while for a spherical shell it occurs at a cusp. PV curves with such a cusp are extremely sensitive to imperfections in the shell shape such that the snapping threshold for manufactured shells can be much lower than the theoretical value (35). As a consequence, the isobaric snapping threshold on inflation of a conical shell is more predictable and reproducible than for a spherical shell (see also S4.2).

The function of the cylindrical wall with height h_w and thickness t_w is to provide a site for clamping at the bottom of the actuator. Because of the height of the wall, variations in the clamping boundary conditions have a negligible effect on the peak and valley of the PV curve. Instead, the influence of the cylindrical wall on the PV curve is that of a nearly linear spring in series with the conical shell. For high wall aspect ratios h_w/t_w, the wall has a high compliance which causes each point of the actuator PV curve to increase in volume with an amount proportional to the pressure at that point. Therefore, tuning the aspect ratio of the wall allows to affect the volume coordinates at every point of the PV curve without changing the peak and valley pressure. However, the precise influence of the cylindrical wall on the actuator PV curve is out of the scope of this research.

The conical shell and the cylindrical wall have the same outer radius r_o and are joined by a compliant hinge. At the inner side of the conical shell, another compliant hinge connects the shell to a central cap with with radius r_i and thickness t_s. The inner and outer corners of both hinges are rounded with a radius of curvature of ρ_i and ρ_o, respectively, in order to limit stress concentrations. Apart from θ_s and t_s, the geometric parameters have limited effect on isobaric snapping thresholds which determine the underactuation behavior. Therefore, all these geometric parameters are kept constant throughout all reported simulations and experiments with values as reported in Table S1. Where possible, the dimensions are expressed as numbers that require a small amount of significant digits to

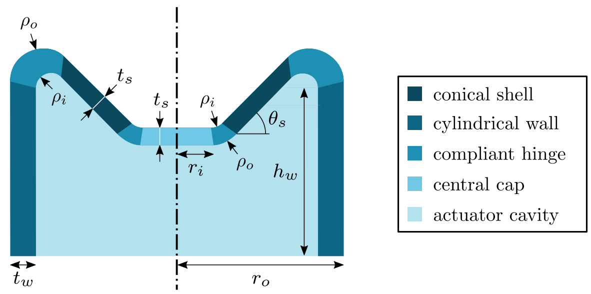

Fig. S1. Parametrized conical shell actuator geometry. The conical shell actuator consists of different functional segments as indicated by the legend. All drawn line segments are either straight lines or circular arcs. Apart from θ_s and t_s, all geometric parameters have constant values throughout all simulations and experiments. Their values are reported in Table S1.

r_o	r_i	h_w	t_w	ρ_o	ρ_i
10 mm	2 mm	10 mm	1.5 mm	2 mm	1 mm

Table S1. Conical shell actuator dimensions.

facilitate machining of the moulds with common shape tools.

With the definition of r_i and h_w at reference points along the inside of the actuator, the thickness t_s has no influence on the initial cavity volume of the actuator. However, this definition enforces a geometric constraint on the maximum allowed thickness of the conical shell:

$$t_s \leqslant \rho_o - \rho_i + \frac{r_i \sin\theta_s}{1-\cos\theta_s}. \qquad [1]$$

In the parameter study, the highest value for θ_s is 60° which with the other parameters from Table S1 results in a maximum possible t_s/r_o of 0.45. This value is well above the value for which the PV curves cease to be nonmonotonic. The limit on t_s imposed by our parametrization therefore does not constrain the range of achievable isobaric snapping thresholds.

S2. Actuator manufacturing

S2.1. Mold design. Each of the actuators characterized for the piano demonstrator is manufactured as a monolithic structure by curing PDMS (Sylgard 184, Dowsil) in a dedicated mold. There are two differences between the geometry of the mold cavity shown in Fig. S2 and the geometry described in S1, but they have a negligible effect on the actuator PV-characteristic. The first difference is that the cylindrical wall has a draft angle θ_w to limit the force required to demold the actuator. The second difference is that the cylindrical wall extends downward and widens into a flange with a toroidal ring along its outer perimeter. This feature allows airtight clamping of the

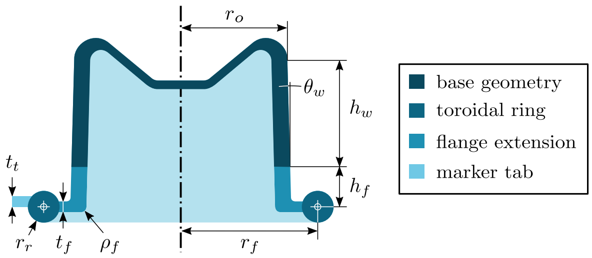

Fig. S2. Manufactured geometry. The geometry of the manufactured conical shell actuators contains additional features to facilitate manufacturing and characterization. The dimensions of these features are reported in Table S2.

θ_w	r_f	h_f	t_f	t_t	r_r	ρ_f
1.5°	12.85 mm	3.75 mm	1 mm	1 mm	1.5 mm	0.5 mm

Table S2. Dimensions of the manufactured actuator geometry.

actuator (see S2.3) while respecting the boundary conditions used in the simulation. Additionally, a small marker tab protrudes from the toroidal ring. It breaks the axisymmetry of the actuator and therefore provides a reference to measure angles in the radial plane. This allows to determine whether any asymmetric deformation in a set of actuators with the same geometry originates from a bias the mold geometry or from factors unrelated to the orientation within the mold (e.g. the clamping boundary conditions or random material heterogeneities). The dimensions of these features are the same for all manufactured actuators and are given in Table S2.

As shown on Fig. S3A, the mold consists of a male (1) and a female (2) half with a parting surface (3) coinciding with the equatorial plane of the toroidal ring. The small area of the parting surface limits the possibility of particles being trapped between the mold halves and affecting their spacing. In the male half, three overflow channels (4) allow excess rubber to spill from the mold as it is closed. The placement of both the parting line and the overflow channels concentrates any excess material left after demolding in those parts of the structure that do not deform under inflation. Therefore, these uncontrollable additions of extra material do not alter the PV characteristic. Additional features of the mold include three holes for 2 mm diameter alignment pins (5) and three holes for M3 bolts to clamp the mold halves together (6). Finally, both mold halves feature a flattened side (7), a chamfered widening (8) of the parting surface and threaded M3 holes located on the external faces at the axis of revolution (9). These three features facilitate separating the mold halves with a variety of tools.

The male and female parts of the molds are machined from a brass round bar (CuZn39Pb3, diameter 65 mm). In a first machining step, all axisymmetric features of the molds are turned on a precision lathe (Spinner MB). Then, all non-axisymmetric features (the tab on the toroidal ring, all holes and the flattened side) are milled on a 5-axis micro milling machine (Kern MMP). Finally, the holes for the alignment pins are reamed on the latter machine to achieve the required tolerance. Using these precision machines minimizes the geometric imperfections present in the mold to minimize uncontrollable deviations from the simulations. Since the produced surfaces are smooth, the moulding process requires no demolding agent

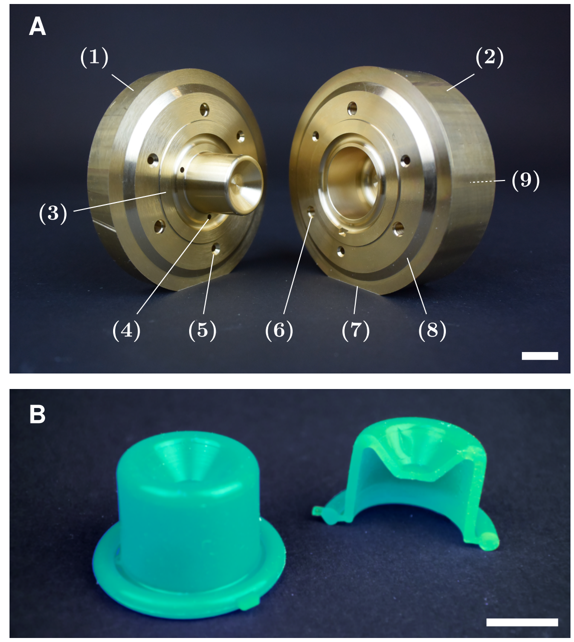

Fig. S3. Actuator molding. **A**, Brass mold for the A_4 actuator of the piano demonstrator. Indicated mold features are (1) male mold half, (2) female mold half, (3) parting surface, (4) spill hole, (5) alignment pin hole, (6) clamping bolt hole, (7) flat side, (8) parting chamfer, (9) threaded hole at the backside of both molds. **B**, Two samples of manufactured actuator. The right sample has been cut in half to show the cross-sectional profile (color enhanced to emphasize the profile). Scale bars are 10 mm.

that could affect the properties of the cured silicone.

S2.2. Actuator manufacturing procedure. For this research, six molds were made, one for each of the actuators in the piano demonstrator. The actuators were manufactured following the procedure below which was designed to minimize imperfections:

1. Clean both matching mold halves, the three alignment pins and the three sets of M3 bolts, nuts and washers with isopropanol. This removes contaminants that can introduce imperfections as they can locally inhibit curing of the silicone rubber.
2. Prepare the silicone rubber in a clean cup according to the provided instructions. For the actuators in the demonstrator, we use 8 g of PDMS (Sylgard 184, Dowsil) in the recommended weight ratio of prepolymer to cross linking agent of 10:1. To visually differentiate different actuator geometries, dip the tip of a skewer in silicone pigment (Silc Pig, Smooth-On) and mix it through the silicone until it has a homogeneous color. We verified experimentally that adding this small amount of pigment has no measurable effect on the actuator PV characteristic.
3. Degas the cup with the silicone for ten minutes inside a chamber connected to a vacuum pump (P6Z, Ilmvac, with ultimate pressure 2×10^{-3} mbar). This removes air bubbles from the mixture that can cause imperfections. To further minimize the risk of trapping air bubbles on filling the mold halves and assembling the mold, the silicone mixture should be sufficiently viscous. Therefore, place the silicone mixture briefly in an oven (45 s at 75 °C)

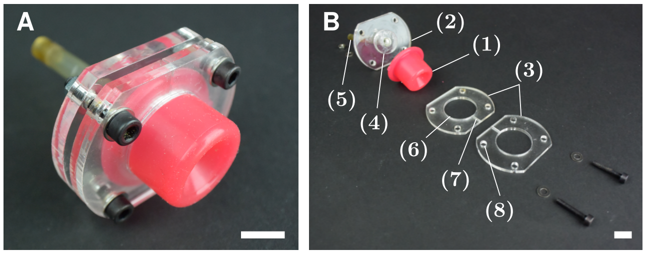

Fig. S4. Actuator clamping. A, Clamping assembly. **B**, Exploded view with indicated features (1) actuator, (2) base plate, (3) top plates, (4) cylindrical protrusion, (5) supply tubing, (6) top plate central hole, (7) slit to accommodate deformation, (8) holes for clamping bolts. Scale bars are 10 mm.

after degassing to reach the desired viscosity.

4. Pour the silicone mixture in the mold cavity. Filling all wells in both mold halves and placing them in a vacuum chamber for an additional five minutes minimizes the risk of trapping air bubbles during final assembly of the mold.
5. Assemble the mold. Place the female half on a table with the mold cavity facing up and insert the three alignment pins in the corresponding holes. In a fluent movement, flip the male half upside down and slide it over the alignment pins until both mold halves make contact. Next, insert three M3×35 bolts through the corresponding holes and tighten the M3 nuts to clamp the mold shut. Use a washer at both sides of every bolt to minimize damage from the nut and bolt to the mold. Finally, remove any excess silicone spilling from the mold.
6. Let the silicone cure. To decrease the recommended curing time of 24 hours, place the mold in a 75 °C oven for two hours and then let it cool down for 30 minutes at room temperature.
7. Open the mold. First remove any excess rubber that cured on the surface of the mold. Then remove the three nuts and bolts. Finally, slowly insert the shaft of a screwdriver in the chamfered parting plane of the mold while applying isopropanol for lubrication until the mold halves can be pulled apart.
8. Remove the actuator from the mold by gently working it free with blunt tweezers and applying more isopropanol. When it is released from the mold, use a precision knife to remove the excess rubber that cured in the overflow channels from the flange.

Fig. S3B shows one of the actuators manufactured following this procedure.

S2.3. Actuator clamping design. The clamping connects a conical shell actuator mechanically to a support structure and fluidically to a pressure supply line. Both types of connections impose requirements on the clamping design. Mechanically, we require that the clamping is stiff and constrains any motion of the bottom of the cylindrical wall with minimal prestrain of the actuator. This prevents loads on the support structure from influencing the actuator PV curve and can be easily modeled in simulations. Fluidically, we require an airtight connection since any leaks would distort the volume reading in the experimental PV curve measurement (see S3).

All those requirements are met by the clamping assembly shown in Fig. S4. It clamps the actuator (1) between a base plate (2) and two top plates (3) such that the actuator experiences both a radial and an axial clamping force. Both the 4 mm thick base plate and the 2 mm thick top plates are manufactured from acrylic glass sheets by lasercutting (Speedy 100R, Trotec Laser). The base plate features a 4 mm high cylindrical protrusion (4) with a central hole that provides an airtight fit for a piece of 6 mm OD polyurethane tubing (PUT6-C, Misumi) (5). The connection between the base plate, the protrusion and the tubing is further secured by applying cyanoacrylate glue (Loctite 460, Henkel adhesives) along the edge of the hole. Both top plates feature a central hole (6) and a cut (7). While the protrusion of the base plate fits loosely inside the actuator cavity, the central holes of the top plates are undersized with respect to the outer radius of the actuator. When the clamping is assembled, the cut in the top plates allows them to deform and accommodate the difference in radius. This permanent deformation imposes a radial clamping force on the flange with a small radial prestrain on the actuator. To secure the top plates and provide a way of connecting the clamping to other structures, sets of M3 bolts and nuts are inserted through oversized holes (8) in each of the acrylic parts and are tightened. This also compresses the toroidal ring axially which introduces an additional clamping force. Assembling the clamping with the cuts in the two top plates at opposite sides from each other results in the most uniform distribution of both the radial and axial clamping force.

S3. Pressure-volume curve characterization

S3.1. General setup. To obtain the pressure-volume characteristic of an actuator, we clamp the actuator as described in S2.3 and connect it to the setup shown in Fig. S5. This setup performs a stepwise inflation of the actuator (1) with a known volume and measures the resulting pressure in the actuator cavity. The volume in the actuator over time is determined and tracked by a LabVIEW program running on a personal computer (2). This program actuates a custom syringe pump consisting of an Arduino Uno running the LabVIEW Interface For Arduino and equipped with a motor shield (3), a stepper motor (QSH-4218-35-10-027, Trinamic) (4), a linear stage (LX-2001C, Misumi) (5) and a gastight glass syringe (model 1050 TLL, Hamilton company) (6). Driving the stepper motor in half-stepping mode results in a total volumetric resolution of 2.1 µL per step of the syringe pump. To ensure that the volume displaced by the syringe pump is equal to the increase in volume of the actuator cavity, all tubing (PUT6-C, Misumi) is stiff and the entire fluidic circuit is filled with water (see S3.2). A T-bore stopcock (7) attached to these tubes allows to connect other devices to the circuit.

With the change in actuator volume determined by the syringe pump, a piezoresistive pressure transducer (PR-21S, Keller Druckmesstechnik) (8) powered by a DC voltage supply (E0300-0.1, Delta Elektronika) (9) measures the pressure in the actuator cavity. To minimize pressure differences between the sensor and the actuator cavity due to hydrostatic and viscous effects, we place the sensor and actuator at the same height and use short tubes with an inner diameter of 4 mm in combination with low flow rates. Finally, an analog data acquisition card (NI 9215, National Instruments) (10) transmits the pressure signal to the LabVIEW program with a resolution of 15.2 Pa.

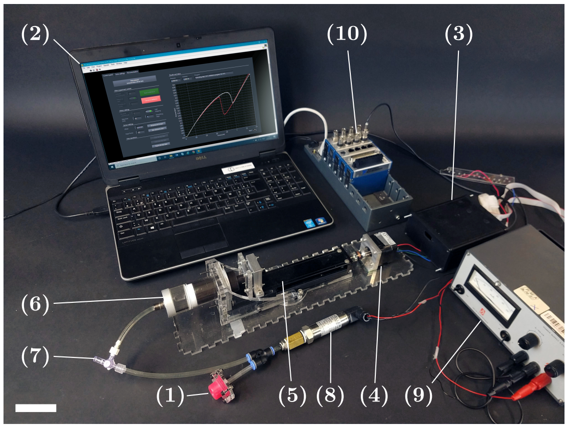

Fig. S5. Pressure-volume curve measurement setup. Experimental setup to measure the pressure of an actuator under volume controlled inflation with indicated features (1) actuator under test, (2) LabVIEW program running on a personal computer, (3) Arduino and stepper motor driver, (4) stepper motor, (5) linear stage, (6) glass syringe, (7) stopcock, (8) pressure transducer, (9) voltage supply and (10) data acquisition module. Scale bar is 50 mm.

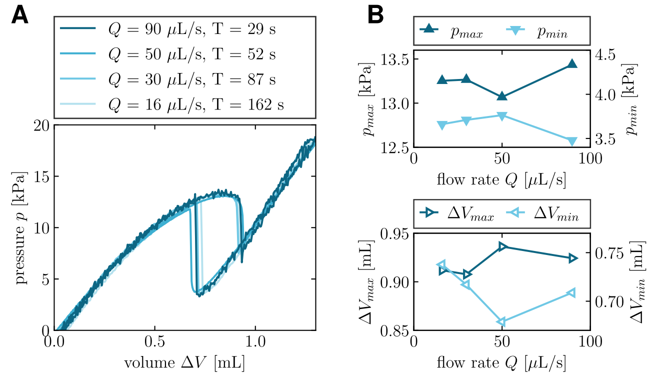

Fig. S6. Influence of flow rate. **A**, Pressure-volume curves measured on the same D_5 actuator with varying flow rate Q resulting in different measurement durations T. **B**, Influence of this varying flow rate on the isobaric and isochoric snapping thresholds. The left and right vertical axes give the value for the threshold on inflation and deflation, respectively. Both axes have the same scale but different offsets.

S3.2. General procedure. Several factors can skew the data of the measured pressure-volume characteristic. Therefore, all experiments are preceded by the following preparatory steps:

1. Close the clamping assembly under water and connect it to the measurement setup filled with water in advance.
2. Connect a syringe filled to 20 % of its capacity with water and no air inside to a piece of tubing filled with water branching off from the stopcock. Hold the syringe in a vertical position with the plunger facing upward and then pull the syringe plunger to create a partial vacuum in the fluidic circuit and trap air bubbles in the syringe. These first two steps serve to remove as much air as possible from the setup since the compression of pockets of air directs a part of the volume displaced by the syringe pump away from the actuator cavity, skewing the volume data.
3. Slowly release the plunger of the syringe, disconnect the tubing from the syringe and connect it to a reservoir of water at atmospheric pressure. This restores the pressure in the circuit to atmospheric pressure and the actuator to its stress-free configuration.
4. Using the LabVIEW program, retract the plunger of the syringe pump to draw 2 mL of water from the reservoir. The motion history of the plunger is then identical to that at the end of a cycle of inflation and deflation. Consequently, any hysteresis present in the syringe pump due to e.g. deformation of the sealing ring causes no difference between the volume in the syringe at the start and at the end end of the measurement cycles. We observe that this removes a shift of approximately 20 µL in the effective actuator volume between the start of the first and the second measurement cycle.
5. Wait 15 s and close the branch of the stopcock that leads to the reservoir so that the volume displaced by the syringe pump is directed exclusively towards the actuator.
6. Calibrate the offset on the pressure transducer signal such that it reads zero in this reference configuration.
7. Repeatedly inflate the actuator to a volume at least 50 % larger than the intended measurement range until the measured PV curves do not change significantly between subsequent measurements. This is a sign that the alignment of the polymer chains has stabilized and that the Mullins effect has subsided within the desired range of strain.

Next, a measurement of the actuator pressure-volume curve is initiated through the LabVIEW program. It inflates and deflates the actuator a number of times without pauses between subsequent strokes. During each of these strokes, the program drives the syringe pump to dispense volume in increments of 2.1 µL at regular time intervals. After every step of the stepper motor, the program stores the cumulative volume displaced by the syringe pump, the average value of the pressure transducer over that step and a timestamp. Together with a sound signal that is generated by the program at the start and end of every stroke of the syringe pump, these timestamps allow synchronization with footage of the actuator recorded by a camera (Nikon 1 V3).

The flow rate with which the actuator is inflated and deflated is 50 µL/s for all measurements. This value results from an experimental study where the flow rate is varied between 16 µL/s and 90 µL/s, which is the highest flow rate achievable with the setup (Fig. S6). In case viscoelastic material damping in the actuator of viscous losses of the flow through the tubes is significant, an increase in flow rate results in higher pressures on inflation and lower pressures on deflation. However, even for the maximum flow rate the difference between the pressure on the inflation and deflation stroke is negligible apart from the region where the isochoric snapping transitions cause hysteresis. Moreover, the difference in the isobaric snapping thresholds between experiments using different flow rates is insignificant compared to the level of noise on the signal. Similarly, there is no clear trend in the evolution of the isochoric snapping thresholds with flow rate The influence of viscoelastic effects on the PV characteristic is therefore negligible for the considered flow rates. However, for the highest flow rate of 90 µL, the integration interval for the pressure transducer is only 23 ms. This means that the averaging of the signal of that interval poorly attenuates the power grid frequency of 50 Hz such that the signal to noise ratio is poor. This ratio improves by decreasing the flow rate to 50 µL/s beyond which

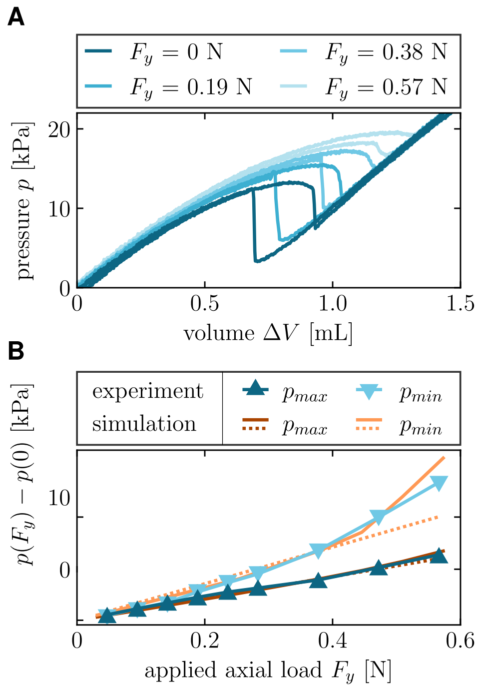

Fig. S7. Influence of external loading. **A**, Pressure-volume curves measured on the same D_5 actuator with varying magnitudes of a constant axial force F_y applied at the conical shell apex. **B**, Influence of the varying load on the isobaric snapping thresholds in the experiment (solid blue lines with triangular markers). The measured trends are modeled accurately by a dedicated finite element analysis (solid orange lines) and by a first order model based on simulation data (dashed orange lines) as described in S5.3.

improvement stagnates. To minimize measurement times, a flow rate of 50 µL/s is selected for all experiments. Finally, every actuator is measured five times and the recorded pressure values are averaged to attenuate noise on a longer time scale, e.g. the vibrations originating from stick-slip of the syringe seal.

S3.3. Measurements with external force. For the measurements of the influence of an external load on the pressure-volume characteristic of an actuator in Fig. S7, the considered load case is a force concentrated at the center of the conical shell cap with a constant magnitude F_y and direction parallel to its axis of revolution. In order to reproducibly load the actuator with such a force, the measurement setup from S3.1 is augmented with a lever made of a wooden board. To minimize the torque required to pivot the lever, it features a fixed axle going through its center of mass that rolls over a hard support surface. At either side of this axle, a site is marked at 120 mm from the fulcrum. One of these sites features a cylindrical peg with a radius of 1 mm that protrudes 15 mm from the board. At this site, the conical shell actuator is mounted upside down such that its axis of revolution is vertical and the peg makes contact with the conical shell cap. The other site marks the place on the lever for a weight that generates the external load for the actuator.

Before conducting an experiment with this setup, steps 1-7 from S3.2 are carried out without any load present on the actuator. Next, the following steps are taken to measure the pressure-volume curve under external loading:

8. Place the desired weight on one side of the lever and position the conical shell actuator above the peg at the other side of the fulcrum. The vertical placement of the actuator should be such that the lowest and highest angle of the lever with respect to the horizontal plane are equal in magnitude. This position minimizes the size of the horizontal component of the force on the actuator since it is proportional to the square of this angle.
9. Inflate the actuator up to the desired volume with the LabVIEW program.
10. Push the lever down to break contact between the peg and the conical shell, position it such that the peg touches the center of the conical shell and release the lever again. This step is required to relieve a shear load on the actuator and a misalignment between the point of origin of the force and the center of the conical shell cap. These effects result from the combination of the second order horizontal motion of the peg, the asymmetrical deformation of the conical shell and the high coefficient of friction between the silicone rubber actuator and the peg.
11. Deflate the actuator back to the starting volume with the LabVIEW program.
12. Repeat step 10 and then 9 up to 11 until five full measurement cycles have been completed.

S3.4. Measurements after cyclic loading. To measure the influence of strain-induced material degradation on the pressure-volume characteristic of an actuator, we subject it to thousands of identical high-amplitude inflation-deflation cycles. Since the measurement setup described in S3.1 is only capable of slow loading cycles, using this system for loading the actuator would lead to a prohibitively long experiment duration. Therefore, a separate loading system incorporating a proportional pneumatic valve (VEAB-L-26-D2-Q4-V2-1R1, Festo LTD) drives the repeated inflation-deflation cycles. The measurement procedure then consists of alternatingly performing a number of loading cycles with this loading system and then connecting the actuator to the measurement system described in S3.1 to characterize the PV characteristic.

Since the changes in the PV characteristic between subsequent measurements in this scheme can be small, potential sources of random variations between measurements should be avoided as much as possible. Therefore, we eliminate the full reclamping and recalibrating procedure given by steps 1-7 in S3.2 between measurements of the PV characteristic by connecting the loading system to the T-bore stopcock marked (7) in Fig. S5. It then suffices to calibrate the measurement system once and to turn the lever of the stopcock to connect the actuator either to the loading system or to the already calibrated measurement system. In this configuration, the actuator and the tube that connects it to the measurement setup should be filled with water at all times as air bubbles skew the measurements. However, the proportional valve in the loading system is rated for use with gases only. The loading system therefore includes a vertically mounted reservoir (CRVZS-0.75, Festo LTD) partially filled with water and air acting as an interface between air and water. The tube leading to the valve attaches to the top of the reservoir and the tube leading to the measurement system attaches to the bottom. In addition to the reservoir and the valve, the loading system includes a number of peripheral pneumatic and electronic components. The pneumatic components include a connection to a centralized supply of compressed air (5 bar), a pressure regulator (LRP-1/4-10, Festo LTD) and another reservoir (CRVZS-5, Festo LTD) to provide the valve with a constant supply of air pressurized at 4 bar. The electronic components include a voltage source (E0300-0.1, Delta Elektronika) to power the valve and a microcontroller board (Arduino Uno) to generate the pressure reference signal for the valve.

We use this setup for the measurements of the actuator made of Dragon Skin 30 in Fig. 2B. In that experiment, the applied loading signal is a trapezoidal signal with a rise and fall time of 1 s and a pause of 4 s between every flank such that the pressure signal and actuator deformation reach equilibrium in every cycle. The pressure signal varies between 0 kPa and 30 kPa which is well above the p_{max} of the actuator to accelerate the potential degradation. Due to visco-elastic effects, the equilibrium pressure measured at a fixed volume is lower for a measurement performed immediately following the repeated loading cycles than for a measurement performed some time later. This difference amounts to 0.5 kPa after 1000 successive cycles. This difference is significant and depends on the hold time during each loading interval, so for the sake of generality we only report values measured at least one hour after completing the indicated amount of loading cycles.

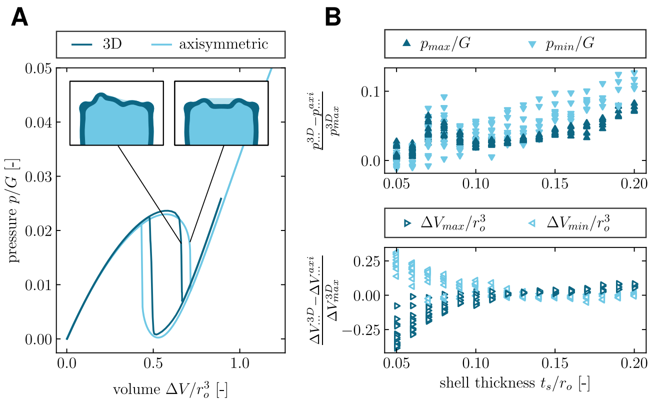

Fig. S8. Influence of simulation dimensionality. **A**, Pressure-volume curves of a conical shell actuator with θ_s = 45° and t_s/r_o = 0.1 resulting from a fully three-dimensional and an axisymmetric simulation. Insets show a cross-sectional view of the actuator deformation at the onset of the isochoric snap on inflation for both simulations. **B**, Difference between the values for the snapping thresholds (isobaric thresholds at the top, isochoric thresholds at the bottom) obtained from tree-dimensional and axisymmetric finite element simulations for every data point in the geometric parameter study. The difference is normalized by the snapping threshold on inflation. Different data points for the same shell thickness t_s/r_o correspond to simulations with different cone angles θ_s.

S4. Pressure-volume curve simulation

S4.1. Finite element model dimensionality. All numerical simulations of the conical shell actuator deformation are carried out with the finite element modelling software Abaqus (Abaqus/CAE 2020, Dassault Systèmes). The simulation geometry is a 360° revolution of the cross-section in Fig. S1 around the axis of symmetry. While the deformation of the conical shell actuator is axisymmetric for a large part of the inflation and deflation cycle, it becomes unstable near the snapping thresholds the deformation and transitions to an asymmetric mode. Moreover, this asymmetry can occur in different planes on inflation and deflation so the deformation has no global symmetry. Therefore a simulation with an axisymmetric description of the actuator deformation yields inaccurate values for the snapping thresholds. As shown on Fig. S8A, an axisymmetric simulation results in a delayed occurrence of the isochoric snapping transitions. The corresponding error on the isochoric snapping thresholds compared to the values from a three-dimensional simulation exceeds 10 % in a large part of the $(\theta_s, t_s/r_o)$-domain (Fig. S8B). A similar error is made in the value for the isobaric snapping thresholds on deflation. Therefore, we only report data obtained from a 360° three-dimensional model even though it consumes an order of magnitude more memory and processing time than an axisymmetric simulation.

S4.2. Geometric imperfections. While a fully axisymmetric deformation trajectory of the conical shell is physically unstable, it can be numerically stable if the geometry in the base state is perfectly axisymmetric. In order to obtain physically accurate asymmetric deformations in the 360° three-dimensional simulation, non-axisymmetric imperfections breaking all symmetry have to be introduced. Therefore, in all reported simulations the stress-free geometry of the actuator is obtained by applying an asymmetric displacement field to the axisymmetric base geometry of Fig. S1.

In the literature on shell buckling, this displacement field usually corresponds to a superposition of buckling modes of the structure with a small scale factor applied (35). For most structures, these buckling modes can be found cheaply as the eigenmodes of the stiffness matrix in the unloaded state. This is possible because these structures do not deform significantly prior to buckling such that the stiffness matrices at the moment of buckling and in the unloaded state are approximately equal. However, for the conical shell actuator, the structure deforms significantly prior to buckling which means that it has to be preloaded to that configuration before solving the eigenvalue problem. This requires a time intensive simulation just to obtain the imperfections that should be applied to obtain an accurate deformation of the actuator. Moreover, due to the highly nonlinear behavior of the actuator, a buckling mode analysis in Abaqus regularly finds modes that are not physically accurate but concern only a small number of nodes in the cylindrical wall. This means that performing buckling modes is not an efficient and reliable method to generate the imperfections for the 240 simulations in the geometric parameter study.

Instead, we perform a buckling analysis on a single actuator and describe the qualitative features of the buckling modes (Fig. S9B) by three imperfection fields (Fig. S9A) that are applied in every simulation reported in this research. These buckling modes are obtained for a D_4 conical shell actuator (geometric parameters in Table S5) with a perfectly axisymmetric geometry. The buckling analysis is preceded by a static step with artificial damping in which the actuator is preloaded to the configurations prior to the isochoric snapping transitions on inflation and deflation. Based on the resulting buckling modes, each imperfection field is defined as a C^1 continuous vector function with as inputs the coordinates of the undisturbed geometry. These three functions have zero magnitude beyond the outer edge of the conical shell.

The first imperfection field corresponds to the global deformation of the conical shell in the majority of the buckling modes. This global deformation is characterized by a rotation

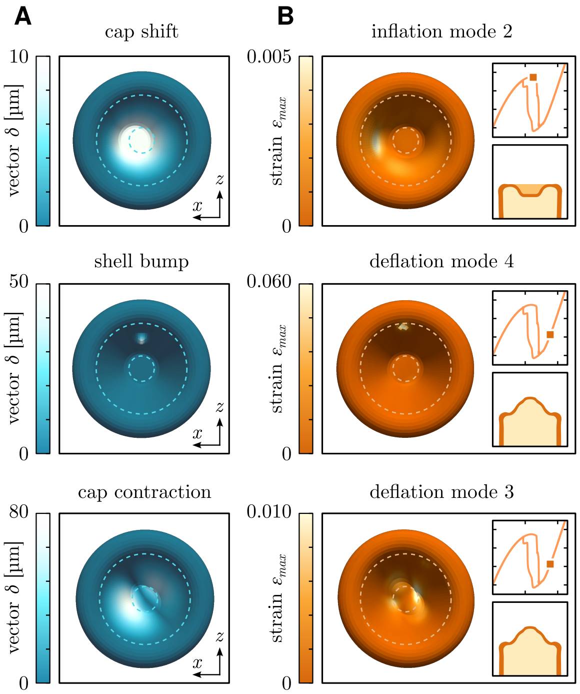

Fig. S9. Imperfections seeded in the simulated geometry. In every finite element simulation, the axisymmetric geometry of the actuator is disturbed by three displacement fields to facilitate convergence. These fields are an offset of the central cap, a localized bump and a contraction of the cap and each of them is a qualitative representation of a buckling mode of the conical shell. **A,** Geometry of the D_4 actuator after applying each of the imperfections. The displacement fields have been scaled by a factor of 100, 20 and 10, respectively, to improve visualization. The color bar indicates the magnitude of the displacement field and the dashed lines are the boundaries of the conical shell region of the actuator in the axisymmetric reference geometry. **B,** Corresponding buckling modes of the D_4 actuator applied as deformation fields to the undeformed geometry. The color bar indicates the maximum principal strain. The insets show the deformation of the actuator (bottom) and the point on the pressure-volume curve (top) to which the actuator is preloaded before applying the buckling analysis.

of the central cap around an axis perpendicular to the axis of axisymmetry. It is caused by a difference in bending curvature of the conical shell at both sides of the cap as shown by the distribution of the principal strain for the second buckling mode on inflation in Fig. S9B. In order to induce this differential curvature, the first imperfection field shortens one side of the shell and elongates the opposite side by shifting the actuator cap. Concretely, the deformation field displaces the cap by ϵ_s along the x-axis of the coordinate system shown in Fig. S9A. The displacement is uniform within the cap and tapers off to zero following the function

$$\vec{\delta}_s = \epsilon_s \cdot \vec{e}_x \cdot \begin{cases} 1 & \rho \leqslant 0 \\ (1+\cos\pi\rho)/2 & 0 < \rho \leqslant 1 \\ 0 & \rho > 1 \end{cases}, \qquad [2]$$

where ρ is the radial coordinate r normalized to go from 0 at the inner edge of the conical shell segment of the actuator (see Fig. S1) to 1 at its outer edge.

The second important feature in different buckling modes is the concentration of stress in a relatively small area on the conical shell as in the fourth buckling mode on deflation in Fig. S9B. Physically, this stress concentration occurs at folds in the shell with a high curvature for thin and steep shells. To facilitate the nucleation of these folds, the second imperfection field describes a localized bump normal to the conical shell. The bump has a maximal elevation of ϵ_b and its profile is defined by

$$\vec{\delta}_b = \epsilon_b \cdot e^{-(20\phi/\pi)^2} \cdot (1-\cos 2\pi\sigma)/2 \cdot \vec{n}, \qquad [3]$$

where $\vec{n}$ is the normal vector of the closest point on the upper surface of the conical shell. σ is proportional to a coordinate measured along the cone angle as $r\cos\theta_s + y\sin\theta_s$ and normalized to go from 0 to 1 between the inner and outer edge of the conical shell. The imperfection is only applied when $0 \leqslant \sigma < 1$ and $0 \leqslant \rho < 1$. Finally, ϕ is the angular distance to the positive yz-plane. This places the apex of the bump in the yz-plane and therefore breaks the only symmetry plane remaining after applying the offset of the cap.

A third feature that appears in different buckling modes is the folding of the cap around an axis in the xz-plane as in the third buckling mode on deflation in Fig. S9B. This folding likely results from an increasing uniform radial compression of the central cap when the conical shell flattens. For a perfectly circular cap there is no preferential direction for buckling which can stall the simulation. Therefore, the third imperfection contracts one axis of the central cap by a factor ϵ_c by applying the deformation field

$$\begin{aligned}\vec{\delta}_c = \epsilon_c \cdot r \cdot &\left(\cos(\phi+\pi/5) + \cos^2(\phi-\pi/20)/4\right)\cdots \\ &\cdot\left(\cos\pi/5\cdot\vec{e}_x + \sin\pi/5\cdot\vec{e}_z\right)\cdots \\ &\cdot\begin{cases} 1 & \rho \leqslant 0 \\ (1+\cos\pi\rho)/2 & 0 < \rho \leqslant 1 \\ 0 & \rho > 1 \end{cases}.\end{aligned} \qquad [4]$$

The resulting shape of the cap is egg-like. This shape breaks symmetry and provides a single location for stress concentration at the edge of the cap analogously to the bump imperfection. The remaining symmetry axis lies in the xz-plane and encloses an angle of $\pi/5$ radians with the x-axis. This axis does not coincide with any other axis of symmetry and therefore ensures that there is always a preferential direction for the redistribution of stresses as the shell buckles.

The magnitudes of the three imperfection modes are constant for all simulations. They are determined as approximately the smallest value for which all simulations converge with the physically correct deformation. Following this criterion, we set $\epsilon_s = r_o/1000 = 10\,\mu\text{m}$, $\epsilon_b = r_o/200 = 50\,\mu\text{m}$ and $\epsilon_c = 0.025$. To assess the influence of the precise magnitude of these imperfections on the snapping thresholds, we perform a series of simulations where either ϵ_s, ϵ_b or ϵ_c is changed while keeping all other simulation parameters constant (Fig. S10). These imperfection sensitivity studies confirm that the selected magnitudes are small enough such that they do not introduce a significant bias. They also show that the isobaric snapping threshold on deflation consistently has the highest imperfection sensitivity of all snapping thresholds and that it is most sensitive to ϵ_s over the examined parameter domain. However, this sensitivity is negligible compared to the imperfection sensitivity of spherical shells. In spherical shells a deviation of $100\,\mu\text{m}$ on a $1\,\text{mm}$ thick shell causes a change in the snapping threshold by over $10\,\%$ (35), but for the conical shell actuator the same imperfection magnitude leads to an error of at most $1\,\%$ on either snapping threshold. Moreover, for the machines with which the brass molds are manufactured (see S2.1) the

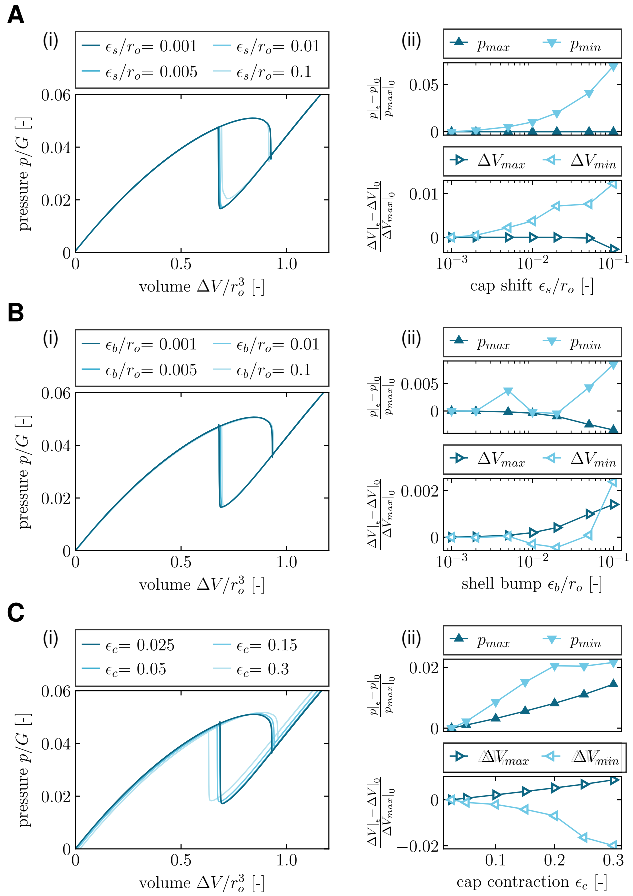

Fig. S10. Geometric imperfection sensitivity. A, Influence of the magnitude of the cap shift imperfection ϵ_s on the behavior of the D_5 actuator. (i) Simulated pressure-volume curves with varying imperfection magnitude ϵ_s. (ii) Change in the isobaric and isochoric snapping thresholds with respect to the simulation with the smallest imperfection magnitude. All values are normalized with the snapping threshold on inflation. **B,C**, Influence of the shell bump imperfection magnitude ϵ_b and cap contraction magnitude ϵ_c on the pressure-volume curve. For the simulations in each of the three subfigures, the two imperfection magnitudes that are not varied explicitly are kept constant at default values $\epsilon_s/r_o = 0.001$, $\epsilon_b/r_o = 0.005$ and $\epsilon_c = 0.025$.

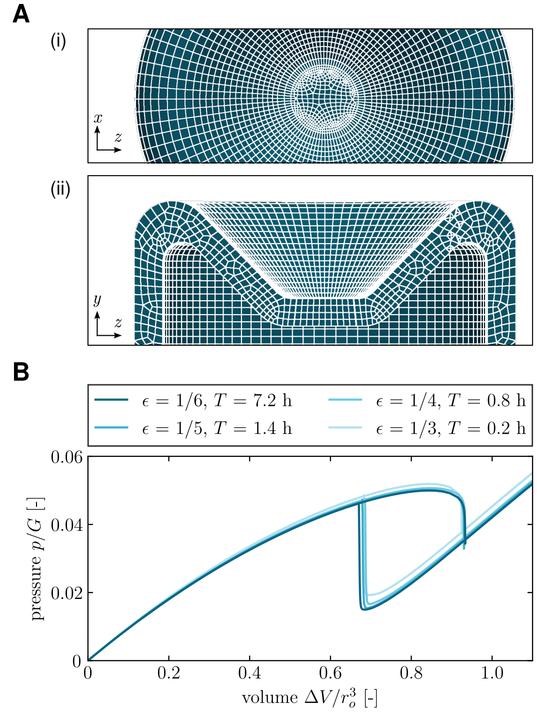

Fig. S11. Finite element mesh. A, Top view (i) and cross-sectional side view (ii) of the finite element mesh of a conical shell actuator. The coordinate system is as in Fig. S9. **B**, Simulated pressure-volume curves for the D_5 actuator with varying mesh sizes ϵ. ϵ is defined as the height of the elements in the central cap region divided by the shell thickness t_s. T is the total wall time of the three-dimensional simulation in hours.

accuracy is in the range of 10 µm and the tolerances on the alignment pins and holes (diameter 2H7) are in the same range. Therefore, any geometric imperfections in the molds do not cause a deviation from the finite element model.

S4.3. Mesh generation. In Abaqus, the imperfect geometry is generated by a python script. It first creates a mesh of the perfectly axisymmetric geometry, then calculates the sum of the local $\vec{\delta}_s$, $\vec{\delta}_b$ and $\vec{\delta}_c$ vectors and finally updates the mesh node coordinates with the Abaqus `editNode` command. The mesh of the axisymmetric geometry consists exclusively of hexahedral C3D8RH elements and is generated by the native Abaqus mesher operating in sweep mode on two partitions of the geometry. The first partition contains every region apart from in the central cap, and it features a mesh of concentric rings with a constant amount of elements. Each ring has the same amount of elements and therefore the element density decreases with the distance from the central axis (Fig. S11A). This gradient in element density is well suited to the deformation of the actuator as the deformation near the center of the actuator has high spatial frequencies, while the deformation of the cylindrical wall is characterized by low spatial frequencies. The second partition with only the central cap features a mesh without a concentric structure. This distribution suppresses the occurrence of numerical artefacts at the pole of the actuator, resulting in faster convergence and more accurate simulation results than in the case of a fully concentric mesh.

The speed and accuracy of a simulation also depend on the size of the elements in the mesh. Larger elements lead to a lower element count and hence faster simulations, but a certain amount of elements is needed especially across the thickness of the shell to accurately capture the bending stress distribution. We base the element size for all simulations based on a mesh size study performed on the D_5 actuator. Fig. S11 shows that including more than four elements across the shell thickness only marginally improves the accuracy of the simulated PV-curve as the isochoric snapping thresholds shift by less than 3 %. At the same time, the simulation duration then at least doubles. Therefore, the mesh size in all simulation is set to one fourth of the thickness of the conical shell.

S4.4. Material model. Since the silicone material of the conical shell actuator is nearly incompressible and experiences strains in the order of tens of percent, it can not be described accurately by a linear material model based on Hooke's law. Instead, a hyperelastic material model is required. Several formulations for the strain energy density of such materials exist with varying behaviors at increasing strains. In Fig. S12, we repeat a simulation of a conical shell actuator for four such material models. Three of the models have multiple parame-

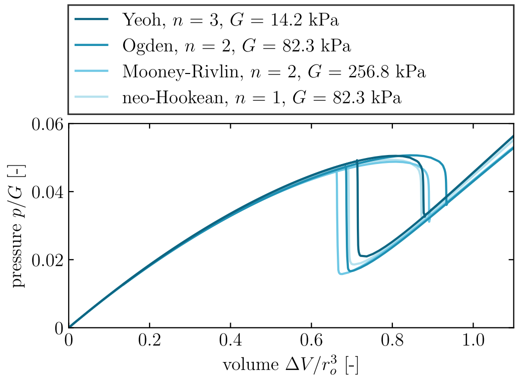

Fig. S12. Hyperelastic material models. Simulated pressure-volume curves for the D_5 actuator for different hyperelastic material models with initial shear modulus G. Each material model is obtained by fitting to experimental data from uniaxial tensile tests for different materials (42). On rescaling the pressure data by G, the curves largely coincide regardless of the number of fitted model coefficients n.

μ_1	α_1	μ_2	α_2
170 kPa	3.5	−88 kPa	0.48

D_1	D_2	ρ	α	β
0 Pa^{-1}	0 Pa^{-1}	1080 kg/m^3	0.018 s^{-1}	0.018 s

Table S3. Abaqus material model parameters

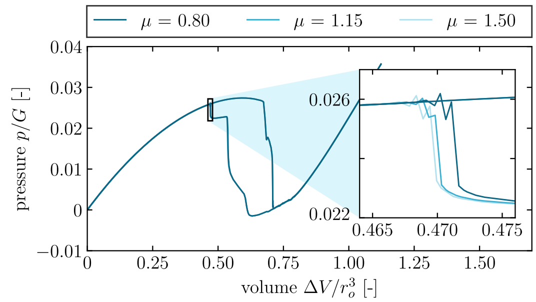

Fig. S13. Self-contact friction model Simulated pressure-volume curves for the D_4 actuator for different coefficients of static friction μ in the self-contact interaction. The curves fully coincide on inflation as there is no self-contact. On deflation, self contact occurs during and between the isochoric snapping transitions. Different μ shift these transitions as shown on the enlarged section of the pressure-volume curve in the inset, but this shift is negligible.

ters which are fitted to experimental data for uniaxial tension of Ecoflex 00-30 (for the Yeoh model), Dragon Skin 30 (for the Ogden model) and Sortaclear 40 (for the Mooney-Rivlin model) using the Soft Robotics Materials Database tool (42). In all cases, we limit the datapoints for fitting to strains up to 60 % as higher strains do not occur in our simulations. The results show that after scaling the pressure data with the initial shear modulus of the material G, the three models give similar results for the pressure-volume curve. Moreover, a one-term neo-Hookean material model with the same G as the Ogden model fully captures the behavior of the actuator even though it is a poor fit for the uniaxial test data. This shows that, since the strains in the conical shell actuator remain relatively limited, the influence of the material on p_{max} and p_{min} can be modeled as uniform scaling with G.

Even so, the variation on especially the isochoric snapping thresholds in Fig. S12 is significant with a relative standard deviation of over 5 %. When the results of the geometrical parameter study are extrapolated to different materials, this can lead to errors on the absolute values of the snapping thresholds. To prevent errors in the relative order of thresholds for different actuators, we therefore recommend a safety margin of 10 % between the thresholds of different actuators that should follow a desired inflation and deflation sequence. Moreover, we select the Ogden model for Dragon Skin 30 as the basis for our parameter study because on average it has the smallest deviation from the snapping thresholds for the three other material models Fig. S12. The strain energy density is then formulated as

$$U = \frac{2\mu_1}{\alpha_1^2}\sum_{i=1}^{3}(\lambda_i^{\alpha_1} - 1) + \frac{2\mu_2}{\alpha_2^2}\sum_{i=1}^{3}(\lambda_i^{\alpha_2} - 1)\,, \qquad [5]$$

with λ_i the three principal stretches. The values for the model parameters given in Table S3.

Other material parameters include D_1 and D_2 which define the material compressibility. For a lack of compressibility data on Dragon Skin 30, the parameters are set zero. This defines the material as practically incompressible and requires the use of elements with a hybrid formulation for hydrostatic pressure to ensure convergence. Finally, since we perform dynamic simulations, a complete material model includes inertia and damping. For the density ρ of Dragon Skin 30, we use the value provided in the material datasheet. For damping, rigorously obtained data on the viscoelastic properties of silicone rubber are rare so we use Rayleigh damping with the sole purpose of numerically stabilizing the isochoric snapping transitions in the simulation. Since the coefficients α and β have no physical significance, their value should be as low as possible in order not to distort the simulation results. Therefore, we obtain the values for α and β in Table S3 by slowly increasing their values until simulations converge over the full range of the $(\theta_s,\, t_s/r_o)$ parameter space under consideration. To minimize the effect of the damping parameters on the simulated PV curve, we select a large simulation time scale (see S4.5).

S4.5. Boundary conditions and loading. Both in the physical actuator and in the finite elements model, three mechanisms constrain the allowed deformation of the conical shell actuator. First, in the physical actuator the bottom of the cylindrical wall features a downward extension, a flange, and a toroidal ring which are clamped as described in S2.3. In the finite element model, this geometry and the stiff clamping assembly is not modeled, but replaced by a fully fixed boundary condition for of all nodes on the bottom face of the cylindrical wall. Additionally, the pressure-volume curve of physical actuators is measured under volume control so any deformation of the actuator has to respect the imposed cavity volume. The corresponding boundary condition in the finite elements model is a hydraulic fluid cavity interaction defined on the inner surface of the actuator. Finally, in some physical actuators the motion of the conical shell is constrained by self-contact. This mostly occurs for thick and deep shells when folds appear on the inside of the conical shell as the actuator approaches the isochoric snapping transition on deflation. In the finite element simulations, a self-contact interaction on the inner surface avoids self-intersection of the mesh. The contact interaction

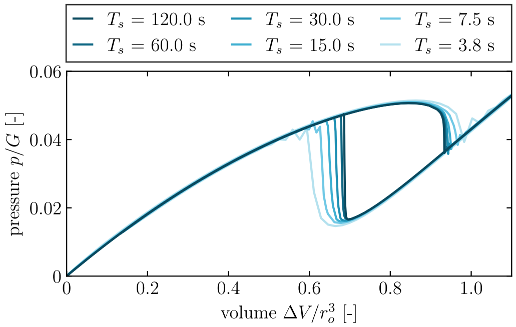

Fig. S14. Simulation time scale. Simulated pressure-volume curves for the D_5 actuator for different in-simulation step durations T_s for both inflation and deflation. Smaller step times lead to a larger contribution of inertial and damping effects that skew mainly the isochoric snapping thresholds when T_s is smaller than 60 s

has an exponential model for the normal behavior and models the tangential behavior as static friction. Data on the exact coefficient of friction of silicone rubbers in self-contact is rare in literature, so we set $\mu = 1.15$ which is a conservative estimate for the static friction coefficient between soft vulcanized rubber and glass (43). However, the exact value for the friction coefficient has a negligible influence on the PV characteristic as shown by the simulations of the D_4 actuator on Fig. S13. In these simulations, contact governs the isochoric snapping transitions on deflation but these threshold change by less than 0.5 % when the friction coefficient changes from 0.8 to 1.5 Therefore, we assume that friction is modelled accurately.

The pressure-volume characteristic of the actuator under volume control is simulated in two implicit dynamic steps. In one step, the actuator is inflated from its stress-free configuration until the shell snaps to the high-volume state. The maximum volume in this step is determined by first performing an approximate axisymmetric simulation, extracting the volume reached directly after the isobaric snapping transition on inflation and increasing it by a margin of 5 %. The other step decreases the actuator volume until the actuator has snapped back to the low-volume state and the volume is 5 % lower than the volume after isobaric snapping on deflation in the axisymmetric simulation. This gradual volume-controlled inflation and deflation is simulated by applying a volumetric flux to the hydraulic fluid cavity interaction using the `Fluid flux` keyword in the simulation input file. Within the bulk of each step, the flux has a constant amplitude but it ramps up and down to zero at the start and end of the step to avoid excessive accelerations. Moreover, enabling the flags for nonlinear geometry, moderate dissipation and adaptive time stepping flags in the step definition facilitates relatively fast convergence in the presence of the discontinuous isochoric snaps.

The modeled duration of each step is determined by the time scale study depicted in Fig. S14. It shows that for time step durations of less than 60 s, the isochoric snapping transitions are less steep than in quasi-static simulations with a higher duration. The reason is that a decrease in step duration corresponds to an increase in the applied volumetric flux so more volume is added or removed while the shell deforms with finite speed during the snapping transition. Another effect is that for a short step duration the velocity of the deformation is higher so damping plays a larger role and increases the hysteresis loop. However, the figure shows that by selecting a step duration of 120 s, we limit all these effects and obtain the quasi-static pressure-volume curve. With this step duration, there is a large difference in the velocity of the actuator shell between the quasi-static deformation and the highly dynamic isochoric snapping transitions. As a compromise between the wall time of the simulation and the time step resolution during the snapping transitions, an adaptive time incrementing scheme is used with the maximum step size limited to 1.8 s.

S5. Simulation data processing

S5.1. Isobaric snapping characteritics. The process of defining and submitting every conical shell actuator simulation is automated with a python script run from the Abaqus interface. When a simulation finishes, the script also creates a file in the JavaScript Object Notation format containing the simulation parameters and results. These results include data for the cavity pressure and volume. In Abaqus, these quantities are tracked by the cavity variables `PCAV` and `CVOL`, respectively. While the inflation of the actuator in the simulation is volume-controlled, it is possible to identify the points at which isobaric snapping transitions would occur if pressure would be controlled instead. The reason is that the isobaric snapping transitions are triggered at the first local maximum in pressure encountered during inflation and the first local minimum in pressure on deflation. The end point of such an idealized snapping transition has the same pressure as that threshold but a higher volume in the case of inflation and a lower volume in the case of deflation.

To improve the accuracy beyond the time step resolution of the simulation, we obtain the snapping threshold by fitting a quadratic polynomial through data points surrounding the local extremum and calculating the apex of this polynomial. Next, we linearly interpolate the actuator volume `CVOL` at both the start and end point of each isobaric snapping transition and take the difference to find the jump in volume $\Delta V|_{p_{max}}$ and $\Delta V|_{p_{min}}$ on inflation and deflation, respectively (Fig. S15A). The vertical strokes $\Delta y|_{p_{max}}$ and $\Delta y|_{p_{min}}$ result in the same manner from the data for the vertical displacement of the shell apex monitored in Abaqus as `U2` (Fig. S15B). Finally, the released energies on inflation and deflation $\Delta U|_{p_{max}}$ and $\Delta U|_{p_{min}}$ are equal to the sum of the change in elastic energy of the actuator ΔU_{in} and the fluidic work performed by the pressure supply during the snapping transition. The former is given by the difference in the Abaqus variable `ALLIE` (Fig. S15C) and the latter by the product $p_{max} \cdot \Delta V|_{p_{max}}$ for inflation and $p_{min} \cdot \Delta V|_{p_{min}}$ for deflation.

S5.2. Isochoric snapping characteristics. Since simulations occur under volume control, isochoric snapping transitions do not manifest as local extrema in the pressure-volume curve but as rapid volume-preserving motions of the actuator. While in most cases this motion results in a clearly identifiable vertical flank in the PV curve, this is not the case in all of our simulations. The reason is that between two subsequent time increments covering the snap, either the difference in pressure is small or the difference in volume is large. The former case occurs for actuators with large θ_s and t_s/r_o that snap between two configurations with distinct deformation modes but similar equilibrium pressures. The latter case occurs when the solver converges quickly despite the discontinuity, such that the time

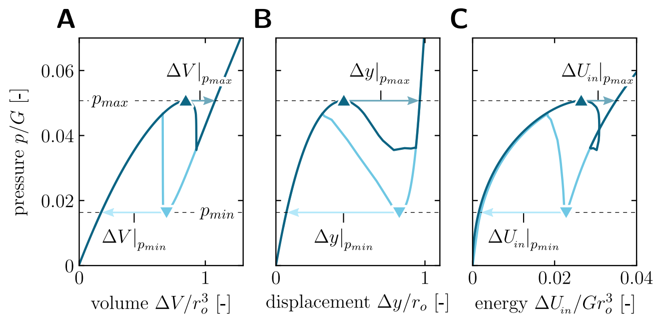

Fig. S15. Extracted characteristics for isobaric snapping. A, Simulated pressure-volume curve for the D_5 actuator under volume control showing the isobaric snapping thresholds p_{max} and p_{min} and the isobaric snapping transitions on inflation (right pointing arrow) and deflation (left pointing arrow). The change in internal actuator volume during these snapping transitions is $\Delta V|_{p_{max}}$ and $\Delta V|_{p_{min}}$, respectively. **B**, Relation between the vertical displacement of the shell apex Δy and the actuator pressure for the same simulation. The arrows indicate the same transitions as on subfigure A and represent a stroke of $\Delta y|_{p_{max}}$ and $\Delta y|_{p_{min}}$. **C**, Relation between the internal elastic energy ΔU_{in} and the actuator pressure for the same simulation. The change in the internal energy during the isobaric snapping transitions is $\Delta U_{in}|_{p_{max}}$ and $\Delta U_{in}|_{p_{min}}$ and is used to calculate the released energy.

increment is not decreased automatically and the constant volumetric flux results in a large increase in volume. Because of these limitations, there is no precise way of identifying the start and end point of the isochoric snaps from the pressure and volume data alone.

Regardless of the difference in pressure or the size of the time increment, however, every isochoric snapping transition results in an increase in the energy dissipated by material damping over the entire model `ALLVD` as this dissipation is required to stabilize the solution. Because of the viscous damping and because of inertia, the rate at which this dissipation occurs is spread as a peak over time where the first base of the peak indicates the effective start of the snapping transition. To find the thresholds that mark the start of these transitions, we first identify all peaks in the dissipation rate that mark distinct snapping transitions. These peaks must satisfy the following conditions:

- The peak dissipation rate needs to be sufficiently high compared to the average dissipation rate in the volume interval before the isobaric snapping threshold is reached. In that interval, the deformation is unconditionally stable so the deformation is always quasi-static. Since in irreversible snapping transitions the average velocity is an order of magnitude higher than in fast but reversible motions, we only retain peaks that are ten times higher than the quasi-static level.
- At both sides of the peak, there needs to be a continuous interval over which the volume changes significantly while the dissipation rate remains sufficiently low. Such a dwell in energy dissipation is only assigned to the peak if between the dwell and the peak, the dissipation rate never exceeds the dissipation rate at that peak. These dwells then correspond to the quasi-static branches between which the transition occurs. If there are multiple peaks that are not separated by such a dwell, in practical applications it is impossible to reliably pause the inflation at the intermediate branch. Therefore, the snapping transitions are practically indistinguishable and we only retain the one with the highest peak dissipation rate. Quantitatively, we put the threshold for the maximum dissipation rate in the dwell on half the dissipation rate of the peak and the threshold for the minimum width of the dwell on 2 % of the full volume scale in the simulation.
- The total energy dissipated between the two dwells flanking the peak needs to be sufficiently high. This filters out short-lived peaks due to simulation artefacts caused mainly by poor stabilization of the self-contact that occurs in actuators with a high cone angle. Assuming that all true snapping transitions dissipate energy in the same order of magnitude, we reject any peak with a total dissipated energy less than one-tenth of the maximum dissipated energy over all peaks.

Next, we identify the start and end point of each isochoric snap as the points at either side of the corresponding peak where the energy dissipation rate is sufficiently low. These points are the volume at which the dissipation rate drops below the threshold of ten times the average quasi-static dissipation rate. In case this threshold value is not reached in between two consecutive snapping transitions, the point at which the dissipation rate reaches its minimal value between the two peaks is selected instead. For each peak, two points meet those conditions. The one that occurs before the peak is the isochoric snapping threshold and the one that occurs after it marks the end of the snapping transition.

We interpolate `PCAV` at both points and take their difference to obtain the change in pressure associated with the isochoric snap $\Delta p|_{\Delta V_{max}}$ for inflation and $\Delta p|_{\Delta V_{min}}$ for deflation. These changes in pressure occur at nearly constant volume, so the source that drives the inflation does not perform work during these snapping transitions. Therefore, the released snapping energies $\Delta U|_{\Delta V_{max}}$ and $\Delta U|_{\Delta V_{min}}$ are the plain differences in `ALLIE` over the course of the snaps. Finally, since the deformation of the shell at the start of the isochoric snaps is asymmetric, there is no straightforward definition for the stroke of the actuator so we do not report data on it.

S5.3. Load-sensitivity of isobaric snapping thresholds. As shown experimentally in Fig. S7, a constant external force F_y parallel to the axis of revolution of the actuator and applied in the center of the conical shell modifies the pressure-volume characteristic. This effect can be captured accurately using the finite element framework of S4 with the addition of a preloading step prior to inflation. In this initial step, F_y is applied as a uniformly distributed force acting on every mesh node covering the actuator central cap. The magnitude of F_y ramps up linearly over the step time of 30 s. Next, the magnitude is fixed for the rest of the simulation and inflation and deflation take place as described in S4.5.

Instead of performing a dedicated simulation to assess the effect of F_y on the isobaric snapping thresholds p_{max} and p_{min}, this relation can also be estimated by processing the data of a simulation without an external load. This follows from the energy balance of a quasi-static inflation of the actuator in the presence of F_y:

$$\int p\, dV = U_{int}(V) + F_y y(V), \qquad [6]$$

where U_{int} is the internal elastic energy in the actuator and y is the axial displacement of the shell apex. Compared to the case without external loading, a load F_y produces a different

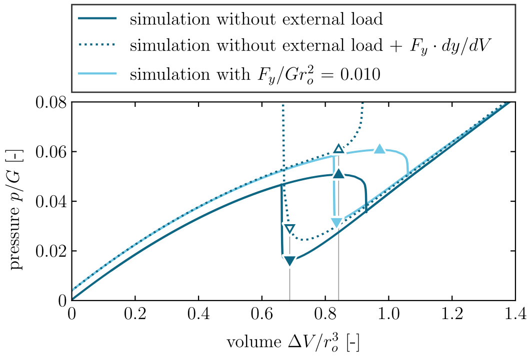

Fig. S16. Estimating load sensitivity. Simulated pressure-volume curves for the D_5 actuator conducted with and without loading with an external force F_y concentrated at the shell apex. The change in pressure between both simulations is estimated by the product of F_y and the derivative of the vertical apex displacement y with respect to volume V in the simulation without external load.

equilibrium deformation for the same volume. Therefore, the functions $U_{int}(V)$ and $y(V)$ depend on the constant load F_y. However, if it is assumed that the effect of F_y on the deformation is negligible, the pressure in the presence of an external load can be expressed as

$$p(V, F_y) \approx \left.\frac{dU_{int}}{dV}\right|_{F_y=0} + F_y \left.\frac{dy}{dV}\right|_{F_y=0}. \quad [7]$$

The derivatives dU_{int}/dV and dy/dV can be calculated from an Abaqus simulation without external load by taking the finite differences of `ALLIE` and `U2` with respect to `CVOL`. Moreover, for such a simulation the last term in equation 6 disappears. This means that the fluidic work is fully converted to the internal energy such that equation 7 simplifies to:

$$p(V, F_y) \approx p(V, 0) + F_y \left.\frac{dy}{dV}\right|_{F_y=0}, \quad [8]$$

and the sensitivity of the internal pressure to external loading to

$$\frac{dp}{dF_y} \approx \left.\frac{dy}{dV}\right|_{F_y=0}. \quad [9]$$

Fig. S16 compares the pressure-volume curve of an actuator obtained from a simulation with an external load to the estimation obtained from applying equation 8 to a simulation without loading. The estimation aligns well with the dedicated simulation for low volumes and accurately predicts the offset in pressure at the initial volume. It also correctly predicts a low sensitivity to F_y at high volumes where the high internal pressure increases the stiffness against loading. Moreover, the isobaric snapping thresholds in the presence of F_y are well approximated by evaluating equation 8 at the volumes at which isobaric snapping initiates in the simulation without loading ($\boldsymbol{\Delta}$ and $\boldsymbol{\nabla}$ on Fig. S16).

However, the accuracy of the estimation for the isobaric snapping thresholds produced by equation 8 is limited by two factors. On the one hand, the accuracy decreases for high external loading because the estimation does not take into account the influence of F_y on the actuator deformation. This reduces the accuracy of the linear estimation for high loading. For the isobaric snapping thresholds the resulting error only becomes pronounced towards the end of the range where F_y does not eliminate isobaric snapping (Fig. S7B). Therefore, for practical purposes this effect is of minor importance. On the other hand, for intermediate volumes equation 8 produces inaccurate results even when F_y is small. In particular, the estimation diverges from the dedicated simulation near the isochoric snapping transitions. This is caused by the rapid displacement of the shell apex while the volume remains approximately constant such that dy/dV increases dramatically. As a result, the slope of dy/dV at the volume where isobaric snapping initiates is large if that volume is close to the isochoric snapping threshold. A small change in the estimation for the volume corresponding to p_{max} or p_{min} then causes a large change in the prediction for the isobaric snapping threshold under loading. In this case, the result of equation 8 is highly sensitive to numerical noise in the simulation and not practically usable. This issue only becomes significant for a small part of the performed simulations and only for p_{min}, so in the majority of cases equation 8 is sufficiently accurate for the design of conical shell actuators.

S6. Inverse design

S6.1. Selection chart generation. For the purpose of designing conical shell actuators with certain values for the isobaric snapping threshold on inflation p_{max}, we plot a selection chart consisting of contours of p_{max}/G over the (θ_s, t_s/r_o)-domain. All points that lie on such a contour correspond to actuator geometries with the same p_{max}/G, so overlaying these contours on contours of other actuator properties allows to find the geometry that has a particular value for the snapping threshold and for the other property (see S6.2). We generate these contours based on data resulting from a geometric parameter study. The parameter study runs a finite element simulation for every sample of (θ_s, t_s/r_o) in a rectangular grid where θ_s varies from 20° to 55° in steps of 2.5° and t_s/r_o varies from 0.05 to 0.2 in steps of 0.01.

In this study, only samples with a high θ_s and a low t_s/r_o feature an isobaric snapping transition. This means that the contours of p_{max} only exist in that region of the domain. To find the precise boundary of this region, we track the values of both the snapping thresholds p_{max} for inflation and p_{min} for deflation. The difference $p_{max} - p_{min}$ is a continuous function that is always positive within the region and that tends to zero on approaching the region boundary. Beyond that boundary, the function is undefined but it is possible to extrapolate the positive values that exist within the region to negative values outside of the region. We achieve this by linearizing $p_{max} - p_{min}$ at each sample point that lies just within the region boundary and evaluating these linear functions at the points on the (θ_s, t_s/r_o)-grid that lie outside the boundary. This leads to multiple estimates for each sample point that lies outside the boundary, which we combine by taking the arithmetic mean. With positive values inside the region and negative values outside it, the marching squares algorithm finds the contour line where $p_{max} - p_{min}$ crosses zero. This interpolated contour line then marks the continuous boundary of the region in (θ_s, t_s)-space where isobaric snapping exists and p_{max} and p_{min} have a value.

To obtain smooth contours of the characteristic over this entire region, we fill this region with a fine triangular mesh of which the maximum edge length is 4 % of the domain width. Then, we apply a spatial gaussian filter with a small standard

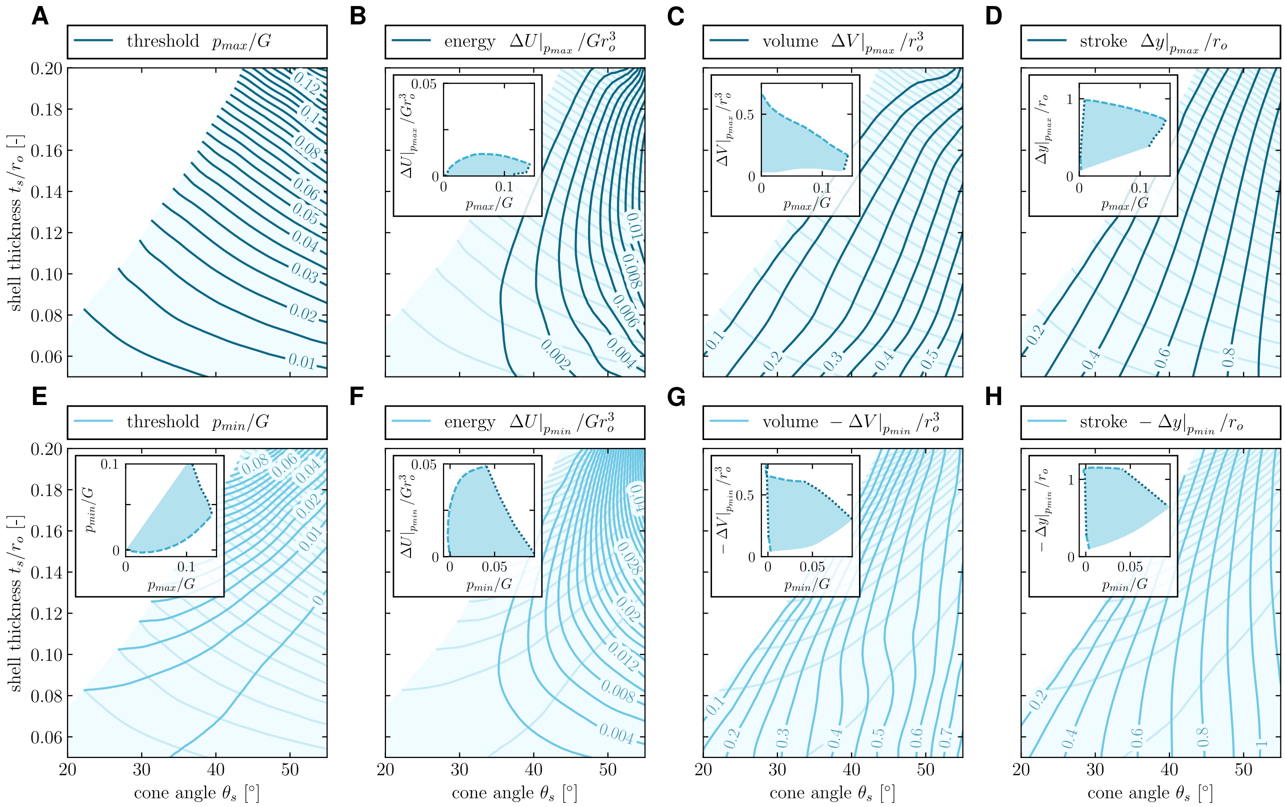

Fig. S17. Design for isobaric snapping characteristics. **A**, Contours of the isobaric snapping threshold on inflation p_{max} in the (θ_s, t_s/r_o)-plane. The levels indicated by the labels on the contour lines are normalized by G, the initial shear modulus of the actuator material. **B**, contours of the energy release associated with the isobaric snapping transition on inflation $\Delta U|_{p_{max}}$ normalized by Gr_o^3, where r_o is the outer radius of the actuator. These contours are overlaid on the contours of p_{max}/G duplicated from subfigure A (lightly colored). The inset show the range of combinations of p_{max}/G and $\Delta U|_{p_{max}}/Gr_o^3$ covered by the simulated data set. Bold lines at the edge of the range indicate data points at the edge of the values that were considered for θ_s (dashed lines) and t_s/r_o (dotted lines) in the data set. At these lines, the range can therefore be extended by considering values for θ_s and t_s/r_o beyond the bounds used data set. At the other edges, the range might be extended by refining the step size of the data set. **C, D**, contours of, respectively, the change in volume and vertical apex displacement (i.e. the stroke) between the start and end of the isobaric snapping transition on inflation. **E**, contours of the isobaric snapping threshold on deflation p_{min}. **F, G, H**, contours of, respectively, the energy release, the change in volume and the stroke associated with the isobaric snapping transition on deflation. They are overlaid on the contours of p_{min}/G from subfigure E. For the change in volume and displacement of the apex, the absolute value is reported.

deviation of 5 % the domain width to the simulated data for p_{max} to remove numerical noise. Next, we evaluate the value for p_{max} at every mesh node by cubic interpolation on the filtered sample data. Finally, we generate the contours using the `tricontour` function from the python matplotlib library. For other actuator characteristics associated with isobaric snapping, such as the threshold on inflation, the stroke of the snapping motion or the energy released during snapping, we use the same region mesh. For characteristics associated with isochoric snapping, the boundary of the region where they exist is found as the zero contour line of the difference in isochoric snapping thresholds. Because the identification of the isochoric snapping thresholds is more sensitive to numerical noise, a stronger spatial filter with a standard deviation of 7 % the domain width is used, but apart from that the same protocol is followed.

S6.2. Design for two target variables. Given a pair of desired isobaric snapping thresholds $\bar{p}_{max}$ and $\bar{p}_{min}$, the method described in section S6.1 allows to generate the matching contours of p_{max} and p_{min} in the (θ_s, t_s/r_o)-domain. If these contours intersect, the x- and y-coordinate of the intersection point correspond to the values for θ_s and t_s/r_o that yield the desired set of thresholds. Instead of generating the precise contours for $\bar{p}_{max}$ and $\bar{p}_{min}$ for a specific application, it is also possible to use a generic plot of contours at selected levels as shown in Fig. S17E. The procedure to find the required cone angle and thickness for a desired $\bar{p}_{max}$ and $\bar{p}_{min}$ is then as follows:

1. Generate a dimensionless representation of the desired isobaric snapping thresholds $\bar{p}_{max}$ and $\bar{p}_{min}$ by dividing both values by the initial shear modulus G of the intended actuator material.
2. Verify that $\bar{p}_{max}/G$ and $\bar{p}_{min}/G$ lie within the domain of possible combinations spanned by the data set (see the inset on Fig. S17E). If this is not the case but the data set contains points with the same $\bar{p}_{max}/\bar{p}_{min}$, a different material with a higher G can be selected. Otherwise, the data set can be extended beyond the boundaries indicated by the dashed and dotted lines by performing simulations with values for θ_s and t_s/r_o outside the currently considered domain, respectively.
3. Identify the contours on Fig. S17A and E that correspond to the desired values for $\bar{p}_{max}/G$ and $\bar{p}_{min}/G$, respectively.
4. Find the intersection point of the selected contours. In case no contours are reported with the exact values for

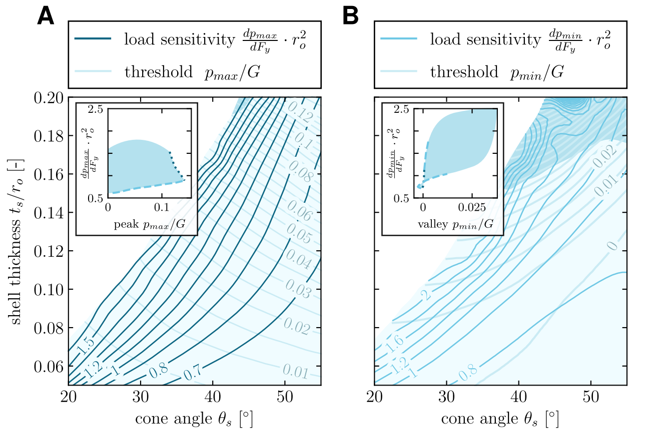

Fig. S18. Design in the presence of external loading. Contours of the sensitivity of the isobaric snapping thresholds to an external load F_y applied to the shell apex. The same conventions as in Fig. S17 are used. **A**, Contours of the load-sensitivity of the isobaric snapping threshold on inflation dp_{max}/dF_y in the (θ_s, t_s/r_o)-plane. They are overlaid on the contours of p_{max}/G duplicated from Fig. S17A (lightly colored). **B**, Contours of the load-sensitivity of the isobaric snapping threshold on deflation overlaid on the contours of p_{min}/G from Fig. S17E. In the darker shaded area of parameter space, the contours are extremely sensitive to numerical noise so can not be used for tuning.

$\bar{p}_{max}/G$ and $\bar{p}_{min}/G$, round them to the nearest contour levels. The exact intersection point can then be found through interpolation, but since we observe that the model prediction error is of the same magnitude as the difference in level between subsequent contours, interpolation only leads to a marginally more accurate result at best.

5. The cone angle θ_s for the actuator with the desired combination ($\bar{p}_{max}$, $\bar{p}_{min}$) corresponds to the horizontal coordinate of the intersection point. Multiplying the vertical coordinate with the desired outer radius r_o of the actuator yields the shell thickness t_s.

The same procedure applies to the design of conical shell actuators for other characteristics associated with the isobaric snaps such as the release of energy or the changes in volume and apex displacement using the other sets of contours in Fig. S17. For example, with the contours in Fig. S17D it is possible to design a conical shell actuator with a particular p_{max} and associated snapping stroke $\Delta y|_{p_{max}}$. Similarly, Fig. S18 presents the contours of the sensitivity of the snapping thresholds p_{max} and p_{min} to an external load F_y. It facilitates the design of an actuator with different snapping thresholds based on the load it experiences. However, these contours are obtained using the approximative equation described in S5.3 so their accuracy is limited. For actuators with a high t_s/r_o in particular, the estimate for the load sensitivity is highly dependent on the volume at which the isobaric snapping threshold is reached. The load sensitivity then changes significantly if it is evaluated at the simulated time increment right before or after the increment at which the isobaric snapping threshold is reached. In Fig. S18, the darker shaded area represents the region where the difference in the estimate varies with more than 50 % between those time increments. The contours in this area are shaped primarily by numerical noise rather than by physical trends so can not be used for design. Apart from this limitation, all contours in Fig. S17 and Fig. S18 can be combined to design conical shell actuators with arbitrary combinations of characteristics.

Finally, the procedure also extends to characteristics associated with the isochoric snapping transitions. However, conical shell actuators with high θ_s and low t_s/r_o feature two distinct isochoric snaps with significant energy release on both inflation and deflation. On the plots in Fig. S19, this manifests as contours splitting in two branches at the point where a second snap develops. As for the non-branching contour plots of Fig. S17, all branches of a contour connect actuator geometries with the same snapping characteristic. For the branch marked by the solid line, on the one hand, this characteristic is associated with the last isochoric snap to occur on inflation or deflation. For the dashed line, on the other hand, this characteristic is associated with the penultimate isochoric snap.

This means that either plot in Fig. S19 also serves to design actuators with particular characteristics for the two isochoric snaps on either inflation or deflation. For example, the parameters that result in a PV-curve with thresholds $\overline{\Delta V}^a_{max}$ and $\overline{\Delta V}^b_{max}$ for, respectively, the last and penultimate transition on inflation correspond to the intersection point between the solid contour at level $\overline{\Delta V}^a_{max}$ and the dashed contour at level $\overline{\Delta V}^b_{max}$ on Fig. S19A. For practical applications, an actuator designed in this way only has added value if it is possible to reliably trigger the first transition without triggering the last (see also S5.2). Since this is only possible if the difference between the two snapping thresholds is sufficiently large, the darker shaded areas on Fig. S19 indicate the regions where that difference exceeds 5 % of r_o^3. Finally, there also exists a region at the highest values for θ_s and the lowest values for t_s/r_o where three distinct snapping transitions occur on deflation. The energy release for that third transition is significantly lower than for the other two snaps, however, so we do not report data for this third snapping transition in Fig. S19.

S6.3. Design for up to four target variables. In the procedure for designing a conical shell actuator with a particular combination of isobaric snapping thresholds in S6.2, the initial shear modulus G of the actuator material is a given. If there is no restriction on the actuator material apart from the one in step 2 of that procedure, however, G can vary over a large continuous range. In practise, this is possible by varying mixing ratios between prepolymers and cross-linking agents, the fraction of an additional silicone thinner and the curing temperature profile. G then becomes an additional degree of freedom for designing conical shell actuator for certain target characteristics of the PV-curve.

The same combination of θ_s and t_s/r_o can realize an infinite number of different PV curves with different values for p_{max} and p_{min} by varying the material from which an actuator with that geometry is made. However, the influence of G on these characteristics is limited to scaling all pressure values linearly, as shown analytically by a dimensional analysis, numerically by the study in S4.4 and experimentally by the measurements in Fig. 2A. This means that the ratio p_{min}/p_{max} for all characteristics generated by the same geometry but different materials is constant. Consequently, a PV-curve with a target value for ($\bar{p}_{max}$, $\bar{p}_{min}$) can be achieved by any actuator geometry where the ratio of the isobaric snapping thresholds equals $\bar{p}_{min}/\bar{p}_{max}$ when G is chosen appropriately. Graphically, all these designs lie on the contour line of p_{min}/p_{max}

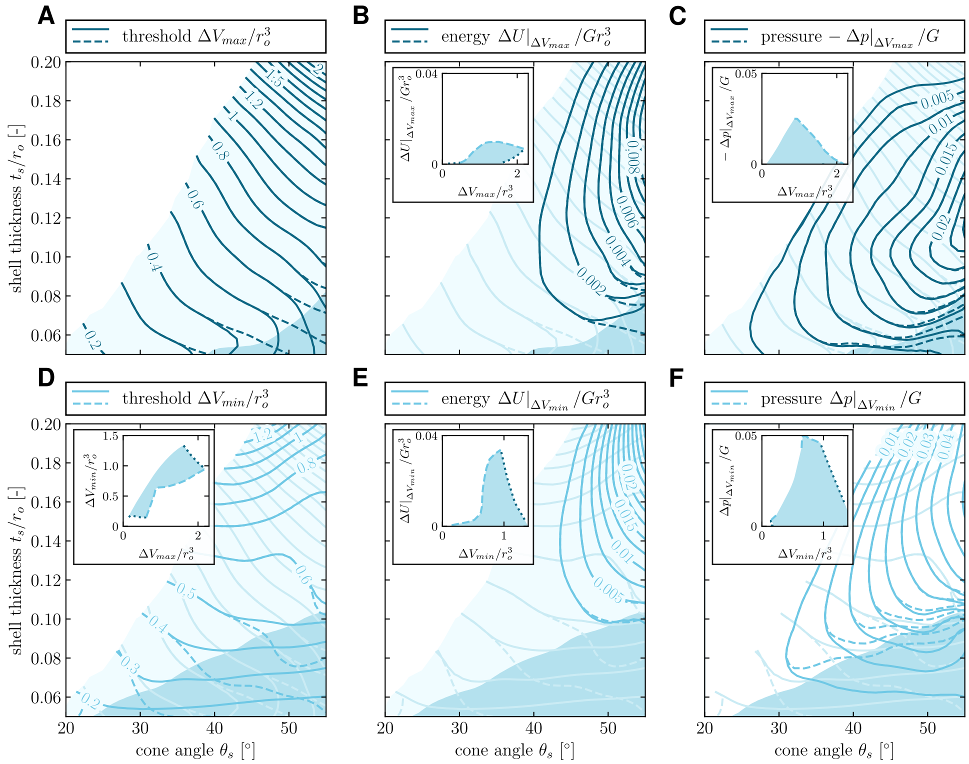

Fig. S19. Design for isochoric snapping characteristics. Contours of the isochoric snapping transition characteristics in the (θ_s, t_s/r_o)-plane using the same conventions as in Fig. S17. **A**, Contours of the isochoric snapping threshold on inflation ΔV_{max}. The darker shaded area indicates the region of the (θ_s, t_s/r_o)-plane where the actuator has multiple of those transitions that are practically distinguishable. In that case, the solid contour lines refer to the last transition that occurs on inflation, and the dashed contour lines refer to the penultimate transition on inflation. **B, C**, Contours of the energy release and the change in pressure associated with the isochoric snapping transition on inflation. These contours are overlaid on the contours of $\Delta V_{max}/r_o^3$ duplicated from subfigure A (lightly colored). **D**, Contours of the isochoric snapping threshold on deflation ΔV_{min}. Here, the solid and dashed lines refer to the last and the penultimate isochoric snapping transition on deflation, respectively. **F, G**, Contours of, respectively, the energy release and the change in pressure associated with the isochoric snapping transition on deflation. They are overlaid on the contours of $\Delta V_{min}/r_o^3$ from subfigure D.

with level $\bar{p}_{min}/\bar{p}_{max}$ in the (θ_s, t_s/r_o)-domain. A contour plot of p_{min}/p_{max} (Fig. S20A) is therefore a graphical tool to design actuators where G is a degree of freedom in the design process. The same reasoning applies to other actuator metrics expressed in terms of pressure such as the jumps in pressure associated with isochoric snaps (Fig. S19B and C). Moreover, a similar reasoning as for G holds for the outer radius of the actuator r_o. It is considered a given in S6.2 but as a design variable, it scales all actuator metrics related to displacements with r_o and all metrics related to volume with r_o^3. Therefore, contour plots of ratios of metrics expressed in volume or displacement allow to graphically design actuators with r_o as an additional degree of freedom.

Contours of a ratio of pressures can be superimposed on any other contour in Fig. S17 or Fig. S19 that does not involve G as normalization parameter to design an actuator for three target variables. It can also be superimposed on contours of ratios of volumes or displacement to design an actuator for four target variables. This allows to find θ_s, t_s, G and r_o for an actuator with for example a desired set of isobaric and isochoric snapping thresholds ($\bar{p}_{max}$, $\bar{p}_{min}$) and ($\overline{\Delta V}_{max}$, $\overline{\Delta V}_{min}$), respectively, using the following procedure:

1. Compute the desired ratios of the isobaric and isochoric snapping thresholds $\bar{p}_{min}/\bar{p}_{max}$ and $\overline{\Delta V}_{min}/\overline{\Delta V}_{max}$.
2. Verify that both ratios lie within the domain of possible combinations spanned by the data set (Fig. S20C). If this is not the case, the data set can be extended with simulations with values for θ_s and t_s/r_o beyond the currently considered domain.
3. On Fig. S20B, identify the contours that correspond most closely to $\bar{p}_{min}/\bar{p}_{max}$ and $\overline{\Delta V}_{min}/\overline{\Delta V}_{max}$ and find the (θ_s, t_s/r_o)-coordinate of the intersection between the two contours.
4. Mark the same (θ_s, t_s/r_o)-coordinate on Fig. S17A. Interpolate the contours of p_{max}/G to find the level ϕ for which the contour would go through the marked point. G is then obtained as $\bar{p}_{max}/\phi$ Similarly, mark (θ_s, t_s/r_o) on Fig. S19A, find the level ψ for which $\overline{\Delta V}_{max}/r_o^3$ goes through that point and find r_o as $\sqrt[3]{\overline{\Delta V}_{max}/\psi}$. Multiply this value for r_o by the vertical coordinate of the marked point to find t_s. The horizontal coordinate of the marked point corresponds to θ_s.

For the contours of $\Delta V_{min}/\Delta V_{max}$ on Fig. S20B we do not consider all isochoric snapping transitions that occur on inflation and deflation as that would lead to a high number of branching lines that would render the graph illegible. Instead, for both ΔV_{max} and ΔV_{min} we select the last isochoric snap that occurs on inflation and deflation, respectively. Nevertheless, for a small range of combinations for p_{min}/p_{max} and $\Delta V_{min}/\Delta V_{max}$ indicated by the darker region in Fig. S20C, the matching contours intersect twice. This means that for those values, two actuators with different geometries and

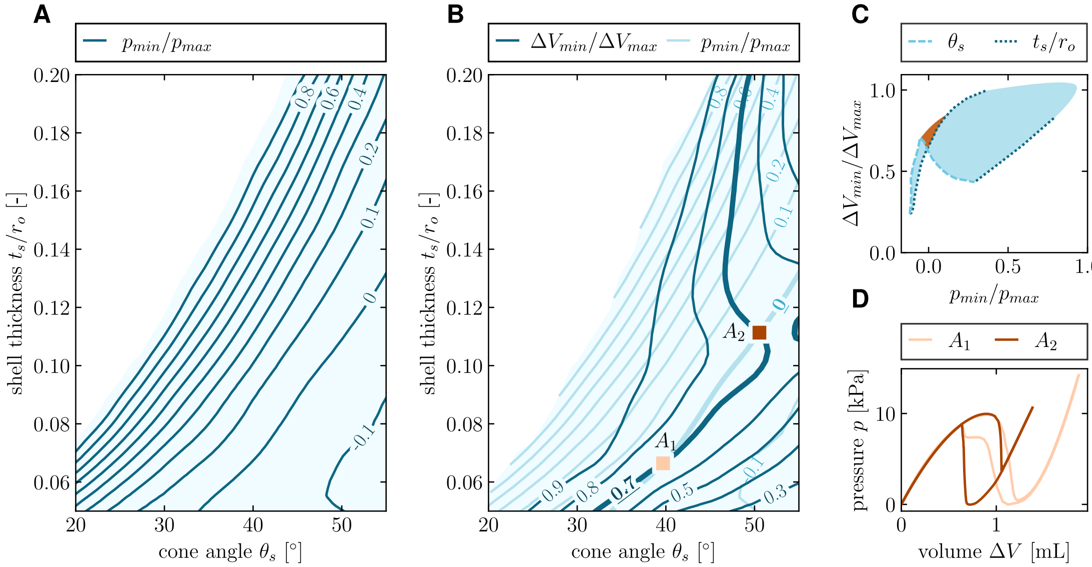

Fig. S20. Simultaneous design for isobaric and isochoric snapping characteristics. **A**, Contours of the ratio between the isobaric snapping thresholds p_{min}/p_{max} in the $(\theta_s, t_s/r_o)$-plane. **B**, Contours of the ratio between the isochoric snapping thresholds $\Delta V_{min}/\Delta V_{max}$ overlaid on the contours of p_{min}/p_{max} duplicated from subfigure A. ΔV_{min} and ΔV_{max} refer to the last isochoric snapping transitions to occur on deflation and inflation, respectively. The highlighted contours for $p_{min}/p_{max} = 0$ and $\Delta V_{min}/\Delta V_{max} = 0.7$ have two intersection points A_1 and A_2, with different geometric parameters but similar PV curves. **C**, Range of combinations of p_{min}/p_{max} and $\Delta V_{min}/\Delta V_{max}$ covered by the numerical parameter study. As in Fig. S17, bold lines indicate data points on the boundary of the considered range for θ_s (dashed lines) or t_s/r_o (dotted lines) in the numerical parameter study. At these lines, the range of combinations can be extended by considering simulations with more extreme values for θ_s and t_s/r_o, respectively. In the region colored in dark orange, multiple geometries realize the same combination. **D**, PV curves for geometries A_1 and A_2 from subfigure B, with the actuator outer radius r_o and material shear modulus G determined such that A_1 and A_2 have the same isobaric and isochoric snapping thresholds (values in Table S4).

materials exist that feature similar PV-curves with identical values for both the isobaric and isochoric snapping thresholds but differ in the stroke or energy of the snapping transitions. Fig. S20D shows a pair of such actuators with the values for the target thresholds as well as the actuator parameters given in Table S4.

Table S4. Design procedure input and output for designing actuators A_1 and A_2 on Fig. S20

procedure input		procedure output		
variable	target	variable	value for A_1	value for A_2
p_{min}	0 kPa	θ_s	39.7°	50.5°
p_{max}	10 kPa	t_s/r_o	0.066	0.111
p-ratio	0	p_{max}/G	0.0090	0.0356
ΔV_{min}	0.7 mL	G	1.12 MPa	0.28 MPa
ΔV_{max}	1 mL	$\Delta V_{max}/r_o^3$	0.397	0.804
ΔV-ratio	0.7	r_o	13.6 mm	10.7 mm

S7. Qualitative sequence design

S7.1. Formal description of a sequence. In order to design a system that performs a certain task, the task must first be formalized. Here, we abstract a task as a series of actions performed in a given sequence. These actions can correspond to for example an actuator undergoing a snapping transition or reaching a certain volume or deformation. Every distinct action that is relevant for the task receives a unique symbol and all those symbols are gathered in the alphabet $\mathcal{A}$. The task is then represented as a sequence of those actions by the combinatorial word $w = a_1a_2a_3\cdots$. Every letter a_i in word w is a symbol from alphabet $\mathcal{A}$ and multiple a_i can refer to the same symbol, meaning that the same action is carried out at different points in the sequence (44). For the application of playing "Ode to Joy" on a piano keyboard, an action corresponds to playing a note so $\mathcal{A} = \{D_4, G_4, A_4, B_4, C_5, D_5\}$ contains the involved notes and $w = B_4B_4C_5D_5D_5C_5B_4A_4\cdots$ encodes the melody.

In the proposed underactuated system of hysteretic elements, there is one input pressure signal p_c while there are multiple independent state variables. These states are the cavity volumes of the different conical shell actuators that are connected to the common input. In such a system, the actions in a word w correspond to the state variables reaching certain variables. For the qualitative design of an underactuated system, only the relative order of these actions is relevant and the time intervals between subsequent actions can be disregarded. The reason is that we consider quasi-static systems where the actuators react instantaneously to the common input signal p_c. In that case, the timing between subsequent actions can be varied by changing the time scale of the input signal alone so the timing is independent of the design of the actuators. Therefore, w suffices to determine the design of the actuators even though abstracting the task as the combinatorial word w only retains information on the relative order of the different actions.

S7.2. Finding the sequences that a given system can produce. Within the framework of S7.1, the task of playing music on a piano is abstracted as the sequence w in which notes are played. In our setup, a note is played whenever an actuator undergoes the isobaric snapping transition on inflation at p_{max} (Fig. S21A). Therefore:

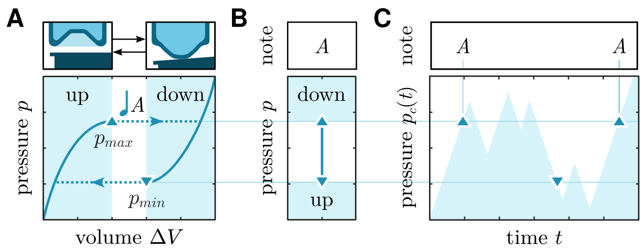

Fig. S21. Operation of a single actuator playing a note. **A**, Diagram of an actuator pressure-volume curve with the isobaric snapping thresholds on inflation (p_{max}, ▲) and deflation (p_{min}, ▼) and the region where the key is in the up and the down state. The pictograms at the top show the physical configuration of the actuator and the piano key in both states. The note symbol shows at which point the note is played. **B**, Simplified notation of the actuator characteristic in subfigure A that only retains the values of p_{max} and p_{min}. **C**, Response of the actuator to a pressure signal $p_c(t)$. ▲ marks the isobaric snap on inflation at p_{max}, which is accompanied by the playing of the note A as shown in the timeline above the graph. ▼ marks the isobaric snap on deflation at p_{min}.

Rule 1. Playing a note

If actuator A is in the retracted state and $p_c(t) > p_{max,A}$, then and only then note A is played.

After this event, the played note quickly fades out and any variation in $p_c(t)$ that stays above $p_{min,A}$ produces no additional note A. Note A can only be played again by first dropping p_c below $p_{min,A}$ to retract the actuator and then increasing p_c again to satisfy Rule 1 (Fig. S21C). Since p_{max} and p_{min} are the only parameters that affect this qualitative behavior, in subsequent figures we simplify the PV curves of all actuators to one dimensional plots that only show the values of p_{max} (▲) and p_{min} (▼) (Fig. S21B).

Given a fixed distribution for the values of p_{max} and p_{min}, different sequences can be produced by varying $p_c(t)$. However, for every possible distribution of p_{max} and p_{min} in a system with three or more actuators, there exist some sequences which can never be played regardless of the $p_c(t)$ that is applied. The reason is that for any three actuators in the system, they can be labeled as A, B and X for which it is impossible to first play A and later B without triggering the undesired note X in between. For example, Fig. S22A shows a system where notes A and B are played by actuators with thresholds $p_{max,A} < p_{max,B}$ and $p_{min,A} < p_{min,B}$. To play note A and then note B in this configuration, $p_c(t)$ must go from $p_{max,A}$ to $p_{max,B}$. Therefore, if an actuator exists with a p_{max} in between $p_{max,A}$ and $p_{max,B}$ and it is in the retracted state at the moment when A is played, by Rule 1 it will be played before B is played. This is the case for actuators X and Y in the beginning of the sequence pictured in Fig. S22A because the starting pressure is lower than any of their p_{min} thresholds. As a result, the sequence AXY is produced instead of the desired sequence AB. To avoid playing the undesired notes, actuators X and Y must be in the extended state before A is played. For Y this can be achieved. A then needs to be retracted by dropping the pressure to $p_{min,A}$ but keeping it above $p_{min,Y}$ so Y stays in the extended state when A is played as shown in Fig. S22A. However, since $p_{min,A} < p_{min,X}$, this action always retracts X first so X always interrupts the sequence AB. Since $p_{min,B}$ does not factor into this reasoning, the same scenario occurs in a system with $p_{max,A} < p_{max,B}$ and $p_{min,A} > p_{min,B}$ as shown in Fig. S22B. This results in the following system behavior:

Rule 2. Unavoidable notes I

If $p_{max,A} < p_{max,X} < p_{max,B}$ and $p_{min,A} < p_{min,X}$, then any played sequence starting with A and ending with B unavoidably contains X in between.

The other possible situation is that $p_{max,A} > p_{max,B}$. In this case, B and any other actuator with a p_{max} lower than $p_{max,A}$ is always in the extended state when A is played. Therefore, B can only be played after A when $p_c(t)$ is subsequently decreased below $p_{min,B}$ to retract B. If $p_c(t)$ is ramped up immediately after this point, all actuators such as Y on Fig. S22C-D with a $p_{max,Y} < p_{max,B}$ and a $p_{min,Y} < p_{min,B}$ remain in the extended state so Rule 1 is not triggered for them before B is played. Actuators such as X, on the other hand, have $p_{min,X} > p_{min,B}$ so they are reset to the retracted state before B such that:

Rule 3. Unavoidable notes IIa

If $p_{max,X} < p_{max,B} < p_{max,A}$ and $p_{min,B} < p_{min,X}$, then any played sequence starting with A and ending with B unavoidably contains X in between.

When A and B refer to the same actuator, the above reasoning stays valid and X is played within any two occurrences of A:

Rule 4. Unavoidable notes IIb

If $p_{max,X} < p_{max,A}$ and $p_{min,A} < p_{min,X}$, then any played sequence starting with A and ending with another occurrence of A unavoidably contains X in between.

To formalize these rules, we note that the conditions of rules 2-4 put no constraints on the relative values of p_{max} of one actuator with respect to p_{min} of another actuator or on the absolute values of p_{max} and p_{min}. Therefore, the range of sequences that can be played by a system only depends on the relative order among the p_{max} of its actuators on the one hand, and the order among their p_{min} on the other hand. Formally, these relative orders are represented by the words s_{max} and s_{min} which both feature one letter for each actuator in $\mathcal{A}$ in the order of ascending p_{max} and p_{min}, respectively. As an example, for the system shown in Fig. S22A, the snapping threshold orderings are $s_{max} = AXYB$ and $s_{min} = YAXB$. We also introduce the notation w_A^B to mean any subword of word w that starts with letter A and ends with letter B later in w but does not contain any A or B in between. For example, in the sequence $w = ABCABAC$, w_A^C refers to ABC and AC but not to $ABAC$. w_A^A refers to any subword of w starting with A and ending with another A without any A in between, so in the example w_A^A refers to BC and B. With these constraints, rules 2-4 can be rephrased as:

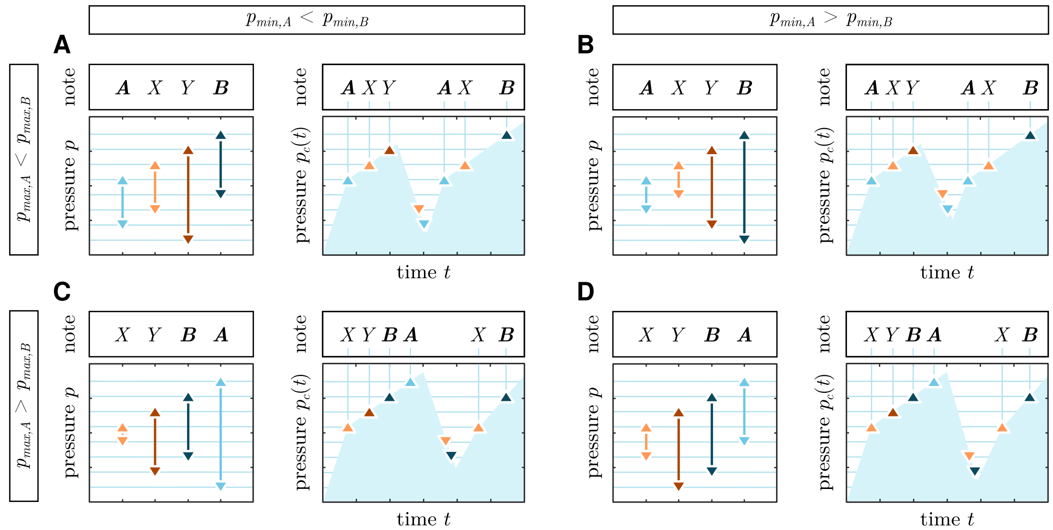

Fig. S22. Constraints on playable sequences. **A-D**, All possible arrangements of peaks and valleys for a pair of actuators A and B. In every case, actuator X always is an example of a note that will always be played directly between A and B. Actuator Y is an example of a note that can appear directly between A and B but is avoidable when it is played before A and is not reset to down state when A is reset. For each subfigure, the left panel shows the condensed representation of the pressure-volume curves of the actuators with their names above as in Fig. S21B. The right panel shows the pressure signal $p_c(t)$ required to play A and then B without Y appearing in between and shows the inevitability of playing X. As in Fig. S21C, the letters above the plot indicate the sequence of notes that is played over time.

Rule 5. Sequences that can be played

A sequence w can be played with orders s_{max} and s_{min} if and only if for every set of notes A, B and C (B and C can be identical) ordered as $A < B \leqslant C$ in s_{max}

(I) all subsequences w_A^C contain B if $A < B$ in s_{min}
(II) all subsequences w_C^B contain A if $A > B$ in s_{min}

For a system with only two actuators, in Rule 5 B and C are taken to be the same. If $s_{min} = AB$, only case (I) remains which is trivially satisfied because any w_A^B contains B by definition. A system with $s_{max} = AB$ and $s_{min} = AB$ can therefore play any sequence w of arbitrary length that only contains two distinct notes. For any system with three or more actuators, however, every set of three distinct actuators A, B, C will lead to a case (I) or (II) that is nontrivial so it is impossible to play sequences containing AC or CB, respectively. Therefore, no single system is capable of playing all sequences containing three or more distinct notes and an algorithm is required to find a system that can play a particular sequence.

S7.3. Finding the systems that can produce a given sequence. The problem of finding the threshold orderings s_{max} and s_{min} that satisfy Rule 5 belongs to the class of constraint satisfaction problems (45, 46). In this framework, the variables are the positions in s_{max} and s_{min}, the domain of each of those variables is the set of actuators $\mathcal{A}$ and the constraints are those imposed by 5. A simple algorithm to find all solutions to the constraint satisfaction problem is to loop over all possible combinations of threshold orderings s_{max} and s_{min} and for each combination check whether the constraints are satisfied or not. This check can be performed efficiently by first constructing a binary lookup table which for any three actuators A, B and C indicates whether w_A^B contains C or not. Verifying Rule 5 then amounts to evaluating a value in this lookup table for every set of actuators ordered as $A < B \leqslant C$ in s_{max}. If a system contains h hysteretic actuators, this amounts to a total of $h(h-1)(h-2)/3!$ evaluations involving three actuators and $h(h-1)/2!$ evaluations involving two actuators. For example, a system with $h = 8$ requires a total of 84 evaluations to confirm that a given system can play a given sequence. This number of evaluations takes in the order of 10 µs for a python script running on an Intel Core i7-8650U processor clocked at 1.9 GHz. Even though the time to check a single (s_{max}, s_{min}) is low, looping over all possible combinations requires performing this procedure $(h!)^2$ times. For $h = 8$, this already results in a run time of over 10 hours so a brute force approach is not practical for designing systems that can play complex sequences.

More efficient solving strategies have been developed in the field of constraint programming. These strategies can be classified as complete or incomplete, where incomplete solving strategies are generally efficient in finding solutions, but it is not guaranteed that they find one if one exists (45). Since in many practical cases the amount of orderings that can produce a certain sequence is rare (see S7.5), we opt for a complete solving strategy that is guaranteed to find all possible solutions or proves that no solution exists. The core principle of most complete solvers is a backtracking scheme in which a solution is constructed variable by variable and checked in every intermediate state. In case such an intermediate solution does not satisfy the constraints, all solutions that can be constructed from this partial solution are eliminated without evaluating them all. Complete solvers differ in the degree of inference they do on the constraints to detect whether a partial solution inevitably leads to a conflict later on or not. A higher degree of inference reduces the number of partial solutions that needs to be evaluated but increases the effort to evaluate each partial solution (45).

This trade-off in performance is well documented for problems where every constraint involves one or two variables but not in case more variables are involved. The problem of finding a system that satisfies Rule 5 falls in the latter category because every constraint on a triplet (pair) of notes involves three (two) positions in s_{max} and two positions in s_{min} for a total of five (four) variables per constraint. For the lack of documentation on solver performance in this case, we limit our algorithm to a basic backtracking strategy without algorithmic inference. However, a general guideline to increase the efficiency of solvers is to reduce the number of variables involved in each constraint (46). Therefore, in our algorithm we construct the solution in such a manner that some constraints only involve three variables. Although the efficiency of the solver can be improved by applying techniques established in constraint programming on this transformed set of constraints, the efficiency of our algorithm is already sufficient for most practical applications (see S7.5).

In our formulation, the backtracking algorithm builds s_{max} one actuator at a time. At the start of the algorithm, s^0_{max} is empty. In any subsequent recursion step i, s^i_{max} is a partial ordering containing i actuators. The algorithm then selects one actuator from alphabet $\mathcal{A}$ covering w that is not part of s^i_{max} and adds it to the end of that ordering to form s^{i+1}_{max}. Next, the algorithm verifies whether or not sequence w is compatible with s^{i+1}_{max}. If this is the case, a new recursion step starts to generate s^{i+2}_{max}. Otherwise, the actuator that was newly added to the ordering is replaced by another actuator that is not included in s^i_{max} and recursion continues with the updated s^{i+1}_{max}. This process continues until the partial ordering contains all actuators in $\mathcal{A}$ or until all valid candidate actuators are exhausted over all recursion steps.

With this approach, a large range of possible s^i_{max} can be eliminated without evaluating them all. The reason is that building s_{max} exclusively by appending actuators guarantees that if s^i_{max} fails to produce the desired sequence w, inevitably all s_{max} and s_{min} that can be generated from this partial ordering fail to produce w as well. This follows from further analysis of case (I) and (II) in Rule 5. Case (I) states that for every $A < B < C$ in s_{max} and $A < B$ in s_{min}, all w^C_A must contain B. This also means that w^B_A can never contain C. If this would not be the case and w^B_A would contain C between A and B, then it would contain a subword w^C_A which by case (I) would contain B. w^B_A would then contain another B in between A and B, which directly contradicts the definition of w^B_A in S7.2. An analogous reasoning shows that case (II) also means that for any $A > B$ in s_{min}, w^B_A can never contain C. Therefore, Rule 5 imposes the following constraint on s_{max} regardless of the order of A and B in s_{min}:

Rule 6. Constraint on s_{max}
If a system with $A < B < C$ in s_{max} plays a sequence w, then no subsequence w^B_A can contain C.

Consequently, any subsequence starting with an actuator A and ending with an actuator B with a higher p_{max} can only include actuators with a lower p_{max} than B. In the algorithm, actuators are added to s^i_{max} exclusively in the last position. Therefore, any actuator C that is not part of s^i_{max} will occupy a higher position in s_{max} than any actuator A that is part of s^i_{max}. This means that as soon as Rule 6 fails for an A and B in a partial s^i_{max} because w^B_A contains a C not part of s^i_{max} at that point, then it also fails for any s^{i+1}_{max}. This result is independent of the order in which actuators are added to s^i_{max} or the possible ordering of all actuators in s_{min}. By the same logic, Rule 6 limits the range of candidate actuators that can be appended to s^i_{max} and for which recursion should continue:

Rule 7. Eliminating candidates to append to s^i_{max}
If actuator $B \notin s^i_{max}$ and there exists an $A \in s^i_{max}$ such that a w^B_A contains other actuators $C \notin s^i_{max}$ then B can not be appended to s^i_{max} to form s^{i+1}_{max}.

If in an iteration $j > i$ all actuators C that trigger Rule 7 for an actuator B and s^i_{max} have been incorporated in s^j_{max}, B can then be appended in the next recursion step.

Whenever an actuator is added to s^i_{max}, it generates a number of constraints on the relative positions in s_{min} of this actuator and every other actuator in the partial ordering. These constraints follow directly from logically inverting Rule 5:

Rule 8. Constraints on s_{min}
If actuator B is added at the end of s^i_{max} to form s^{i+1}_{max}, then sequence w can only be played if for all $A \in s^i_{max}$, $C \notin s^{i+1}_{max}$: (I) $A > B$ in s_{min} if any w^C_A does not contain B (II) $A < B$ in s_{min} if any w^B_C does not contain A

Since every constraint involves only one actuator C of which the position in s^i_{max} is not yet fixed, the constraints that are generated for B are independent of the order in which these actuators C are added to form the final s_{max}. Therefore, if the constraints in Rule 8 that are generated in recursion step i contradict the constraints generated in previous recursion steps, then s^i_{max} can never lead to a valid solution. As is the case for Rule 7, B can then be eliminated as a candidate to be appended to s^i_{max} until all actuators C that cause the conflicting constraints have been added to s_{max} first.

To check for such conflicts, the algorithm maintains a set S_A for every actuator A in $\mathcal{A}$ which contains all actuators that have to occupy a higher position than A in the final s_{min}. Whenever Rule 8 generates a new constraint $A < B$, S_A is extended with B. Moreover, S_A is extended with every element C in S_B because $A < B$ and $B < C$ imply that $A < C$. A conflict then occurs if S_A comes to contain A as that would mean that $A < A$. If a complete s_{max} is reached without the appearance of such conflicts, the algorithm generates all possible orderings s_{min} that are compatible with the constructed set of constraints. It constructs these orderings by recursively adding actuators to a partial s^i_{min} defined in the same way as s^i_{max}. In every recursion step, it selects an actuator A that does not appear in S_B for all B not in s^i_{min}, and adds it to the end of s^i_{min} to find s^{i+1}_{min}. When the algorithm terminates, it has found all possible orderings s_{max} and s_{min} with which a sequence w can be played. The complete algorithm is summarized in Algorithm 1

S7.4. Finding the input signal that produces a given sequence. The algorithm described in the previous section finds possible

Algorithm 1 Qualitative sequence design algorithm

Input $\underline{w}$: desired sequence of snap-through events.
Input $\underline{s^i_{max}}$: optional ordering of thresholds on inflation. Empty by default.
Input $\underline{S^i}$: optional lookup table with S_A for every actuator A. All S_A are empty by default.
Output $\underline{\phi}$: all (s_{max}, s_{min}) that can play w where the first i elements of s_{max} are given by s^i_{max}.

```
function getOrdering(w, [s^i_max, S^i])
    φ ← ∅
    // terminate recursion
    if s^i_max contains all actuators that appear in w then
        for s_min compatible with S^i do
            φ ← φ ∪ (s^i_max, s_min)
    // find candidates to add to s^i_min
    for B ∈ w and ∉ s^i_max do
        canBeNext ← true
        S^{i+1} ← S^i
        for A ∈ s^i_max do
            for C ∉ s^i_max do
                // Rule 7
                if C ≠ B and C ∈ at least one w^B_A then
                    canBeNext ← false
                // Rule 8 (I)
                if C ≠ B and B ∉ all existing w^C_A then
                    S^{i+1}_B ← S^{i+1}_B ∪ A ∪ S^{i+1}_A
                    if B ∈ S^{i+1}_B then
                        canBeNext ← false
                // Rule 8 (II)
                if A ∉ all existing w^B_C then
                    S^{i+1}_A ← S^{i+1}_A ∪ B ∪ S^{i+1}_B
                    if A ∈ S^{i+1}_A then
                        canBeNext ← false
        // continue recursion
        if canBeNext then
            s^{i+1}_max ← s^i_max with B added at the end
            φ ← φ ∪ getOrdering(w, s^{i+1}_max, S^{i+1})
    return φ
```

orderings for the isobaric snapping thresholds (s_{max}, s_{min}) that can realize a desired sequence w when the appropriate input signal p_c is applied simultaneously to all actuators. A separate algorithm generates p_c in function of the found ordering. It represents p_c as a word where every letter corresponds to the threshold of a snapping transition that should be triggered at that point in the sequence. This means that for every element in p_c that represents a p_{max}, the input should increase beyond it while staying below all higher p_{max} at that point in the sequence. Similarly, for every p_{min} in p_c, the pressure should decrease below it to trigger the transition on deflation, but stay above all lower p_{min}. Therefore, a pair of adjacent elements in p_c puts an upper and lower limit on the values that the physical input signal can reach in the time interval between both snapping transitions. Every input signal that stays within these limits at all times performs the desired sequence, irrespective of the exact profile it follows.

Algorithm 2 summarizes the approach to generate p_c given a sequence and a valid ordering for the snapping thresholds. It initializes p_c as a word with one letter corresponding to the p_{max} of the last actuator in s_{max}. This puts all actuators in the extended state by playing them all in the order of s_{max} before the sequence starts. After this initialization, the algorithm loops over every note A of the sequence in chronological order and adds thresholds to p_c depending on the relation between $p_{max,A}$ and the last entry in the partial p_c. If $p_{max,A}$ is lower than the last entry in p_c, then A is in the extended state after applying the partial p_c. By Rule 1, A has to be retracted first before the note can be played. Therefore, the input signal is decreased first. In particular, the algorithm continues p_c with the lowest possible p_{min} that does not disrupt the sequence when the input will be increased later on. Next, $p_{max,A}$ is added to the end of p_c in order to play note A. If instead $p_{max,A}$ is higher than the last entry in p_c, then the algorithm guarantees that A is in the retracted state at that point. Therefore, p_c does not need to decrease at that point and the algorithm adds $p_{max,A}$ directly to the end of p_c.

Algorithm 2 Input signal design algorithm

Input $\underline{w}$: desired sequence of n snap-through events.
Input $\underline{s_{max}}$: ordering of all h thresholds on inflation.
Input $\underline{s_{min}}$: ordering of all h thresholds on deflation.
Output $\underline{p_c}$: signal that realizes w when applied to all actuators with threshold orderings (s_{max}, s_{min}).

```
function getSignal(w, s_max, s_min)
    p_c ← p_max of last element of s_max
    for i ← 0..n do
        A ← actuator at index i of w
        if p_max,A < last element in p_c then
            j ← index of A in s_min
            canBeRetracted ← true
            while canBeRetracted and j>0 do
                B ← actuator at index j − 1 of s_min
                k ← i + 1
                while k-th actuator of w < B in s_max do
                    k ← k + 1
                if k-th actuator of w ⩽ B in s_max then
                    j ← j − 1
                else
                    canBeRetracted ← false
            B ← the actuator at index j of s_min
            p_c ← p_c with p_min,B added at the end
        p_c ← p_c with p_max,A added at the end
    return p_c
```

Whenever the algorithm decreases p_c to the $p_{min,A}$ of an actuator A, all actuators B with $B \geqslant A$ in s_{min} are retracted. By Rule 1, B will then play its note as soon as p_c exceeds $p_{max,B}$. Similarly, playing a note C with $C > B$ in s_{max} requires that $p_c > p_{max,C} > p_{max,B}$. This means that if B is retracted together with A, it is always played between this A and the next C:

Rule 9. Actuators that can not be retracted

If after a point in sequence w, a $C > B$ in s_{max} appears in w and B does not appear before it,

then B can not be retracted at this point.

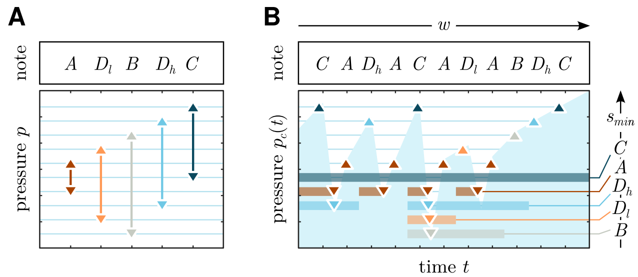

Fig. S23. Generating the input signal to drive a sequence. **A**, Simplified notation of an example system with $s_{max} = AD_lBD_hC$, $s_{min} = BD_lD_hAC$. **B**, Input signal $p_c(t)$ (profile shaded in light blue) to generate the sequence $w = CAD_hACAD_lABD_hC$ for the example system. The resulting snapping transitions on inflation and deflation are marked by ▲ and ▼, respectively. The bold horizontal lines indicate the segments of the sequence where the actuator with the matching p_{min} can be retracted without inevitably disturbing the sequence further on. By Rule 9, they appear whenever no actuator with a higher p_{max} occurs before the next occurrence of the considered actuator in the sequence.

The algorithm evaluates this criterion for every B and decreases p_c to the lowest p_{min} for which none of the actuators that trigger Rule 9 are retracted. This is illustrated for an example system on Fig. S23 where the bold colored lines represent the part of the sequence where Rule 9 is not triggered for a given actuator. It is possible that the rule allows the retraction of an actuator B together with A, but not of another actuator $D > B$ in s_{min}. In that case, the algorithm does not retract B. However, it is guaranteed that in every interval of the sequence over which B does not trigger Rule 9, there is a point where all actuators $D > B$ in s_{min} do not trigger Rule 9 either. At that point, p_c can decrease enough to retract B and its note can be played on the next occurrence of B in w. The proof of this statement consists of two cases.

The first case concerns actuators $D_h > B$ in s_{max} as in the example system in Fig. S23. By Rule 7, no subsequence $w_B^{D_h}$ can then contain actuators with a higher p_{max} than D_h. Moreover, since B does not trigger Rule 9 when A has to be played, the interval of w between A and B only contains actuators with a $p_{max} < p_{max,B} < p_{max,D_h}$. Therefore, the entire interval of the sequence between A and the first D_h after it only contains actuators with a $p_{max} < p_{max,D_h}$. As a result, every $D_h > B$ in s_{max} does not trigger Rule 9 over the entire interval where B does not trigger it either.

The second case concerns actuators $D_l < B$ in s_{max}. Since the full interval over which B does not trigger Rule 9 is always preceded by a $C > B$ in s_{max}, case (II) of Rule 5 applies and D_l inevitably appears between C and B. It is possible that this C does not appear explicitly in the sequence before B, but in that case it is played implicitly during the initialization of p_c. The same C that triggers Rule 9 for B in some part of the sequence triggers that rule also for D_l since $D_l < B$ in s_{max}. Therefore, every interval over which B does not trigger Rule 9 contains both the start and end of the same interval for D_l. If of all actuators D_l has the lowest p_{min} larger than $p_{min,B}$, this means that D_l can be retracted somewhere between C and B. Next, both cases of the proof can be applied to any actuator $D_l' > D_l$ in s_{min} to prove that they can be retracted between C and D_l. This can be repeated recursively until no more actuators with a higher p_{min} are left. At that point, it has been proven that for every note that needs to be played in w, the algorithm is able to retract the actuator in time such that the entire sequence is played correctly.

S7.5. Algorithm performance. Given any sequence w of length n with exactly $h \leqslant n$ distinct elements, the qualitative design algorithm described in S7.3 finds all orderings of the isobaric snapping thresholds (s_{min}, s_{max}) that can realize w. It is possible that multiple solutions exist, but also that no solution exists which means that it is impossible to realize sequence w with the proposed underactuated framework. In this section, we apply the design algorithm to a range of sequences with different n and h to generate statistics on the expected number of solutions as well as the efficiency of the algorithm.

In order to limit the number of sequences that has to be sampled to generate accurate statistics for a given n and h, only non-equivalent sequences are sampled. Two sequences are equivalent if it is possible to consistently relabel the elements of one sequence to obtain the other sequence. For example, $w = ABCAAB$ is equivalent to $w' = CABCCA$ because w' results from w by relabeling A to C and B to A and C to B. Since the precise element labels are irrelevant for the design algorithm, the solutions for w' can be obtained by applying the same relabeling to the solutions for w. Every sequence has $h! - 1$ equivalent sequences, all of which have the same number of solutions and require the same number of steps in the algorithm. Therefore, it is possible to reduce the number of sequences that needs to be sampled by a factor $h!$ without skewing the solution statistics by only considering non-equivalent sequences.

To prevent sampling equivalent sequences, for a given n and h we generate sequences as lists of integer in the range $[0, h-1]$ where an integer h_i can only appear in the sequence if all lower integers $[0, h_i - 1]$ already appear in the sequence preceding it. The number of such sequences is given by the Stirling numbers of the second kind:

$$S(n,h) = \frac{1}{h!}\sum_{k=0}^{h}(-1)^k\binom{h}{k}(h-k)^n, \qquad [10]$$

which can also be calculated through the recursive relation:

$$S(n,h) = h\ S(n{-}1,h) + S(n{-}1,h{-}1), \qquad [11]$$

with $h \leqslant n$ and $S(n,h) = 1$ if $h = 1$. In the context of non-equivalent sequences, the two terms in equation 11 correspond to the two ways in which a sequence of length n with exactly h unique elements can be generated from a sequence with a lower n or h. The first option is to start from one of the $S(n{-}1,h)$ sequences that are one position shorter and append either of the h elements that appear in those sequences to it. The second option is to start from one of the $S(n{-}1,h{-}1)$ sequences that also feature one less unique symbol and add the new symbol $h{-}1$ to it. With this method, it is possible to map every integer $i \in [0, S(n,h){-}1]$ to a distinct sequence as illustrated by Algorithm 3. This recursive algorithm is used in two ways to generate the sequence samples to which the design algorithm is applied for a given n and h. For small n and h, it is possible to iterate exhaustively over all non-equivalent sequences within a reasonable time. Algorithm 3 is then applied for every integer in the interval $[0, S(n,h){-}1]$. However, $S(n,h)$ quickly grows with n and h and the exhaustive sampling scheme then becomes prohibitively slow. Therefore, the Monte Carlo method is used as soon as $S(n,h)$ exceeds 10^4, which occurs at the dashed white contour line in Fig. S24. In that case,

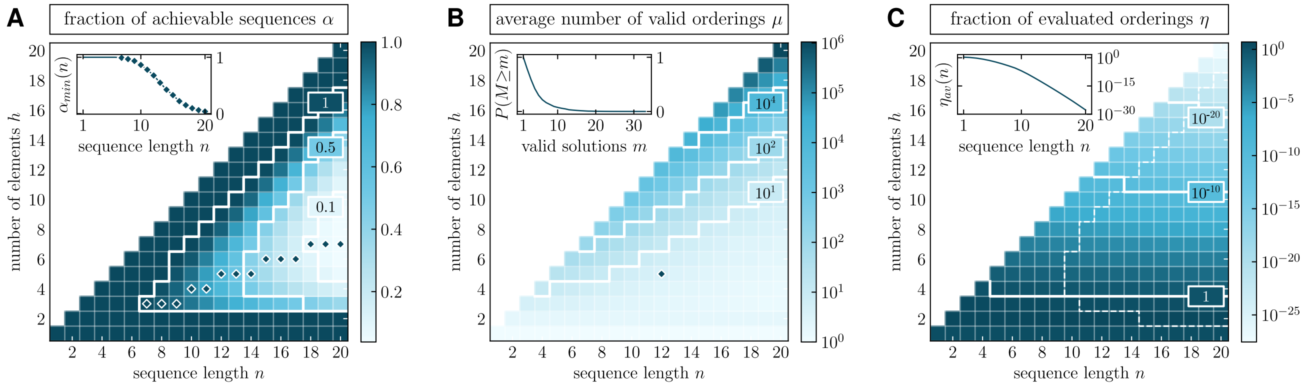

Fig. S24. Qualitative sequence design statistics. **A**, Some sequences can not be achieved by any (s_{min}, s_{max}) with the proposed underactuated architecture. The color of each cell represents the fraction α of all sequences with length n involving exactly h unique elements that can be achieved with our architecture. Selected contours of α are shown as solid white lines. Inside of the contour for α =1, sequences exist that are not achievable. The diamond markers indicate h that yields the lowest α for every value of n in this area. The corresponding α-values are plotted in the inset. **B**, Most sequences that can be achieved by a particular ordering (s_{min}, s_{max}) have more than one equivalent solution. The color of each cell represents the average number of equivalent solutions μ for all sequences of length n with h unique elements that have at least one solution. The inset shows the probability that a sequence with n =12 and h =5 (diamond marker) has more than m solutions in function of m. **C**, The qualitative sequence design algorithm is called recursively for every sequence until it either finds a solution or finds that there is no solution. The color of each cell represents the average number of function calls as a fraction η of the number of permutations of (s_{min}, s_{max}). Above the contour line for η =1, the design algorithm is more efficient than an exhaustive search. The inset shows the average of η over all h for a given n. The data points outside of the dashed line are obtained by exhaustively iterating over all possible sequences and those inside the line are generated from randomly sampled sequences.

Algorithm 3 Generating a unique sequence from an integer

Input $\underline{n}$: length of the sequence w.
Input $\underline{h}$: number of unique elements in the sequence w.
Input $\underline{i}$: integer in the range $[0, S(n,h) - 1]$.
Output $\underline{w}$: sequence represented as a list of integers.

```
function GENERATESEQUENCE(n, h, i)
    if n ⩽ 1 or h ⩽ 1 then
        w ← list of n zeros
    else if i ⩾ h S(n−1, h) then
        i ← i − h S(n−1, h)
        w ← GENERATESEQUENCE(n−1, h−1, i)
        w ← w with h−1 added at the end
    else
        h_i ← ⌊i/S(n−1, h)⌋
        i ← i − h_i S(n−1, h)
        w ← GENERATESEQUENCE(n−1, h, i)
        w ← w with h_i added at the end
    return w
```

10^4 distinct random integers in the interval $[0, S(n,h)-1]$ are generated and Algorithm 3 is only applied with those integers as i to provide the samples for the design algorithm. This method guarantees that every possible sequence has the same probability of being sampled so that the Monte Carlo sampling does not introduce a bias in the solution statistics.

Both in the case of exhaustive and Monte Carlo sampling, the generated sequences with a certain n and h are processed in the same way. The qualitative design algorithm is applied to every sequence sample to find all possible orderings that realize that sequence. Next, a number of metrics is calculated. The first metric α is the ratio between the total amount of sequences for which a solution was found and the total amount of sampled sequences at the given combination of n and h. For all points that lie outside of the solid white line in Fig. S24A, all sampled sequences have a solution so α is exactly equal to 1. In particular $\alpha = 1$ for any n and $h \leqslant 2$ or $n - 2 \leqslant h \leqslant n$. For $h \leqslant 2$, this follows from that fact neither case in Rule 5 is triggered if $s_{min} = s_{max}$. Case (I), on the one hand, does not apply because it involves three distinct symbols while the sequence has at most two. Case (II), on the other hand, does not apply because s_{max} and s_{min} have the same order.

For $h \geqslant n - 2$, all symbols in a sequence occur once apart from one or two symbols X and Y. Therefore, a sequence can be rewritten in terms of X and Y and blocks of symbols separating them, where no two blocks have symbols in common. For example, $w = A_1A_2XB_1B_2YC_1C_2YD_1D_2XE_1E_2$ contains blocks A_1A_2, B_1B_2, C_1C_2, D_1D_2 and E_1E_2. Any two symbols A_1 and A_2 within the same block have the same constraints on their positions in s_{max} and s_{min} relative to any symbol B or C that does not appear in that same block. This follows from Rule 5 where A_1 and A_2 are equivalent for case (I) because all B that appear in $w^C_{A_1}$ do so after the block containing A_1 and A_2, so they also appear in $w^C_{A_2}$. Similarly, A_1 and A_2 are equivalent for case (II) because if a w^B_C contains A_1, it contains the full block that A_1 belongs to including A_2. Because all symbols A_1 and A_2 within the same block in the sequence share the same constraints with respect to symbols outside of their block, they can also appear as a block in s_{max} and s_{min}. Moreover, if A_2 appears after A_1 in the block in the sequence, then any existing $w^C_{A_1}$ contains A_2. Therefore, if A_1 and A_2 appear in the same order in s_{max} and s_{min} as in their block in the sequence, the constraints on their mutual order are satisfied by Rule 5 (I). This means that blocks of actuators that appear once in the sequence can always appear as the same blocks in s_{max} and s_{min}. These blocks then act as a single unit and can be substituted by a single symbol in the sequence without affecting the constraints. Any sequence with $h \geqslant n - 2$ can therefore be rewritten as a sequence with at most nine elements and at most seven unique symbols. For example, the sequence at the beginning of this paragraph reduces to $w = AXBYCYDXE$. An exhaustive search shows that any such sequence with $(n, h) = (9, 7)$ can be realized by an ordering $(s_{max},\, s_{min})$, so all sequences with $h \geqslant n-2$ can be realized as well.

Fig. S24A shows that $\alpha < 1$ if (n, h) is $(7, 3)$, $(8, 3)$ or $(8, 4)$ or if $n > 8$ and $2 < h < n-2$. Otherwise, α decreases monotonically with increasing n if h stays constant. For constant n, α reaches a minimum around $h \approx 0.37n$ as indicated by the diamond markers on Fig.S24A. The inset shows that this minimum in α drops below 0.5 for $n \geqslant 14$ and then converges to zero for large n. As a result, the proposed underactuated architecture is only a practical solution for short sequences, sequences with a small number of distinct symbols or sequences where most symbols occur only once.

A second metric of interest is the average number of solutions for all sequences that have at least one solution, denoted as μ. As shown on Fig. S24B, on the one hand this metric decreases with n because on average the number of constraints generated by Rule 5 increases with n. On the other hand, μ increases dramatically with h because s_{max} and s_{min} then have more positions so any constraint on the position of two elements in those orderings can be satisfied in many ways. While the average μ can be high, in general the distribution of the number of solutions is skewed towards a small number of solutions. For example the inset on Fig. S24B shows the probability that a sequence has more than m solutions for $(n, h) = (12, 5)$. In this case, some sequences feature over 25 solutions, but they only represent 0.1 % of all sequences with a valid solution. Instead, more than half of all achievable sequences have between one and four solutions. This shows that on average, valid solutions for $(s_{min},\, s_{max})$ are rare, limiting the utility of brute force and incomplete constraint satisfaction problem solvers.

A final metric measures the efficiency with which the algorithm navigates the space of all possible orderings $(s_{min},\, s_{max})$. The efficiency is defined as the ratio between the number of recursive function calls required to find all solutions for a given sequence and the total number of possible $(s_{min},\, s_{max})$ combinations $(h!)^2$. This ratio is averaged over all sampled sequences for a combination of (n, h) to produce the metric η. Fig. S24C shows that $\eta > 1$ for $h < 4$. This means that the design algorithm then checks more than $(h!)^2$ combinations which is due to the fact that it also checks partially completed orderings (s^i_{max}, s^i_{min}). The design algorithm is then less efficient than an exhaustive search through all possible orderings, but since h is small, both methods find all valid orderings quickly. If $h \geqslant 4$, however, η drops steeply such that the design algorithm quickly outperforms an exhaustive search by orders of magnitude. For example, the design algorithm checks less than one in every thousand possible orderings for $h = 7$. η also decreases with increasing n, although this dependence is less pronounced than with h. These two effects cause a decrease in the average value for η over all sequences with a fixed length n approximately following the relation $\log_{10}(\eta_{av}) = -0.077n^{1.97}$ as shown on the inset in Fig. S24C. In conclusion, the design algorithm is highly efficient and therefore allows to easily design underactuated systems for highly complex sequences.

S8. Demonstrator design

For the melody of "Ode to Joy", the sequence design algorithm yields the solution $s_{max} = G_4A_4D_4B_4C_5D_5$, $s_{min} = D_4G_4A_4B_4C_5D_5$. In theory, any set of p_{max} and p_{min} that respects these orderings can play the sequence. In the practical demonstrator, however, there are additional constraints on the allowable values for p_{max} and p_{min}. A first constraint is that the difference in the desired p_{max} or p_{min} between any two actuators should be high enough. This prevents that uncontrollable variations in the material stiffness increase the snapping threshold of one actuator above another one, which would result in a different s_{max} or s_{min}. Moreover, pressure fluctuations due to a limited precision of the pressure controller can then trigger undesired snapping transitions. Because of this constraint, we limit the minimal difference between the snapping threshold of different actuators made from PDMS (G =701 kPa) to 1.8 kPa. A second constraint is that the difference between the geometric parameters of any two actuators should be large enough. Otherwise, small imperfections in the dimensions or the alignment of the molds can affect the resulting s_{max} and s_{min}. Therefore, we only consider geometries where p_{max} does not change by more than 1.8 kPa if the cone angle changes by 1.3° or the cone thickness by 26 µm. On Fig. S25, these geometries lie below the lines $dp_{max}/d\theta_s = .002\ G$ and $dp_{max}/dt_s = 1.0\ G/r_o$. The final constraint is that in the keyboard demonstrator the displacement of the conical shell apex should be low enough such that the actuator does not contact the key throughout the retracted state. Moreover, the displacement should be high enough such that the key registers contact throughout the extended state.

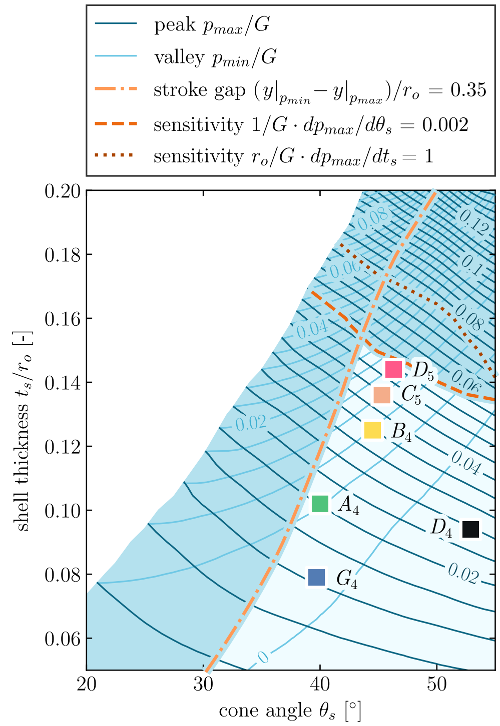

Fig. S25. Tuning actuators for the piano demonstrator. Contours of the isobaric snapping thresholds in the $(\theta_s,\, t_s/r_o)$-plane with markers indicating the thresholds and geometric parameters for the actuators involved in the piano demonstrator. The parameters of these actuators are chosen such that the resulting snapping thresholds respect $s_{max} = G_4A_4D_4B_4C_5D_5$ and $s_{min} = D_4G_4A_4B_4C_5D_5$ while staying within the lightly shaded region. Within that region, the difference in the apex displacement between the two states of the actuator $(\, y|_{p_{min}} - y|_{p_{max}})/r_o >$ 0.35 and the sensitivity of the snapping threshold on inflation to variations in the manufactured geometry $1/G \cdot dp_{max}/d\theta_s < 0.002$ and $r_o/G \cdot dp_{max}/dt_s < 1$.

note	color	θ_s	t_s/r_o
D_4	black	52.9°	0.094
G_4	blue	39.7°	0.079
A_4	green	40.0°	0.102
B_4	yellow	44.5°	0.125
C_5	orange	45.3°	0.136
D_5	red	46.3°	0.144

Table S5. Actuator parameters for the piano demonstrator

Both the highest displacement in the retracted configuration and the lowest displacement in the extended configuration appear at the respective snapping thresholds. Therefore, we only consider geometries to the right of the line at which $y|_{p_{min}} - y|_{p_{max}} = 3.5\,\text{mm}$ on Fig. S25.

With those three constraints, the actuators have to be placed in the lightly shaded area of Fig. S25. This results in the geometric parameters listed in table S5. Finally, conical shell actuators with those parameters are manufactured out of PDMS following the procedure in S2. PDMS is selected because of its high shear modulus which creates a sufficiently large difference between the snapping thresholds of the different actuators with respect to the precision of the pressure supply. Moreover, a high modulus increases the actuator force and limits the sensitivity of p_{min} to the contact force between each actuator and its piano key such that s_{min} is not distorted by the interactions with the keys. For the same reason, the piano keyboard in the demonstrator is a custom construction where the springs that keep the keys in the upper position are less stiff than on commercially available keyboards.