Mathematical and Computational Applications

MDPI

Article

Generative Design of Soft Robot Actuators Using ESP

Martin Philip Venter *,† and Izak Johannes Joubert †

Department Mechanical and Mechatronic Engineering, Stellenbosch University, Stellenbosch 7600, South Africa
* Correspondence: mpventer@sun.ac.za
† These authors contributed equally to this work.

Abstract: Soft robotics is an emerging field that leverages the compliant nature of materials to control shape and behaviour. However, designing soft robots presents a challenge, as they do not have discrete points of articulation and instead articulate through deformation in whole regions of the robot. This results in a vast, unexplored design space with few established design methods. This paper presents a practical generative design process that combines the Encapsulation, Syllabus, and Pandamonium method with a reduced-order model to produce results comparable to the existing state-of-the-art in reduced design time while including the human designer meaningfully in the design process and facilitating the inclusion of other numerical techniques such as Markov chain Monte Carlo methods. Using a combination of reduced-order models, L-systems, MCMC, curve matching, and optimisation, we demonstrate that our method can produce functional 2D articulating soft robot designs in less than 1 s, a significant reduction in design time compared to monolithic methods, which can take several days. Additionally, we qualitatively show how to extend our approach to produce more complex 3D robots, such as an articulating tentacle with multiple grippers.

Keywords: pneumatic soft robot; reduced-order model; Encapsulation, Syllabus, and Pandamonium

Citation: Venter, M.P.; Joubert, I.J. Generative Design of Soft Robot Actuators Using ESP. *Math. Comput. Appl.* **2023**, *28*, 53. https://doi.org/10.3390/mca28020053

Academic Editors: Hans Beushausen and Sebastian Skatulla

Received: 1 February 2023
Revised: 13 March 2023
Accepted: 30 March 2023
Published: 3 April 2023

1. Introduction

Soft robotics is a sub-field of robotics that focuses on the use of compliant materials and significant material deformation in the design and functionality of robots [1]. Their soft nature allows the robot to move and adapt to its environment more naturally, similar to how living organisms move [2]. The literature categorises soft robots by the materials used, articulation mechanisms and the method of energy transfer [3,4]. In their review of contemporary real-world applications of soft actuators, Li et al. highlight the importance of soft actuators to a wide range of fields [5], including notable fields such as medical and surgical devices [6], industrial automation [7] (Figure 1), human–robot interaction [8,9], environmental monitoring and biomimicry [10–12].

A comprehensive review of soft actuator applications by El-Atab et al. emphasises classifying soft actuators based on their source of activation, highlighting the integrated nature of the structure of soft robots and their control [13]. Soft actuators are classified as responding to electrical, magnetic, thermal, light, and pressure input, distinguishing between slow (conventional fluid flow) and rapid (explosive) pressurisation. Under this classification, this research focussed on pressure-driven soft actuators that use air or fluid pressure to inflate a single internal cavity or network of internal cavities to move various robot sections [14–16].

In Lipson's 2014 "state of the field review" [17], he notes several challenges and opportunities for the design, simulation, and fabrication of soft robots that are still problematic today. Likewise, although there have been several developments in simulation and automated design processes for soft robots, the 2020 review by Chen et al. reaffirms that the same challenges remain [18].

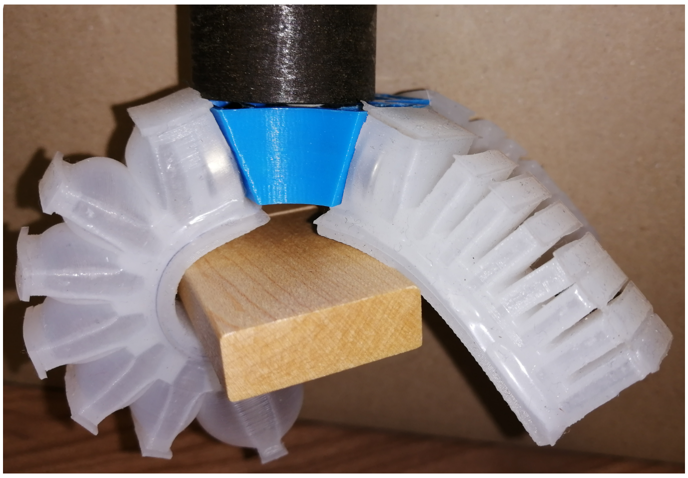


Figure 1. An example of three bending PNA's assembled into a gripper holding a wooden block.

A 2021 review of simulation methods for soft robotic actuators finds that while FEM is a valuable tool for modelling soft fluidic actuators and evaluating the effect of material properties and geometry, several limitations support the need for experimental characterisation to validate FEM results [19]. However, when used with care, we can achieve accurate results, although with a significant computational expense. This review further suggests applying machine learning methods to learn the nonlinear kinematics of soft robots from FEM results. Finally, this paper uses a finite element simulation to generate the training data sets for an alternative data-driven model.

The successful design of soft robots requires a simultaneous design of their topology, control system, and behaviour [20]. The process includes material selection, topology design, control, and fabrication. A deep understanding of soft materials' underlying physics and mechanics is needed to achieve the desired behaviour. Due to the lack of discrete pivot points, the entire material domain contributes to the deformation and response of the robot, resulting in a larger design space with few established design methods. As a result, the design space for soft robots is highly nonlinear and largely unexplored. Current design methods are primarily trial and error, relying on intuition to drive development. However, several research groups have attempted other design methods, such as generative design [18].

Automated design processes, specifically generative design, use algorithmic processes to assist designers and are capable of generating designs for specific design domains by formulating the design process in terms of a set of constraints and objectives. Modern computational resources have increased the capabilities of these automated design approaches, allowing for more powerful generative design generators. Lai et al. [21] provide a compelling overview of the current state of the art. The work by Runge and Raatz [22] is particularly worth mentioning as they provide a robust process for the general design of soft robotic systems using parametric design, simulation, reduced order kinematic modelling and meta-modelling to generate candidate components rapidly. Previous work by this group [23,24] uses a genetic algorithm to design a 15-unit pneumatic network bending actuator using a computational design approach. The algorithm could produce verifiably suitable designs, but the process was computationally expensive.

We propose reducing computational time and improving functionality by breaking down the monolithic optimisation problem into a structured sequence of smaller optimisations and maintaining the benefits of a human-in-the-loop design process. To this end, we are exploring the "Encapsulation, Syllabus, and Pandamonium" (ESP) [25] method as an alternative design approach, using reduced-order models of a typical pneumatic bending actuator trained on data derived from FE modelling. We aim to replicate and extend our previous work to produce a high-articulation multi-gripper.

2. Materials and Methods

2.1. Materials, Manufacturing, and Testing

Typically, soft robots are made from flexible materials, such as silicone, rubber, and various polymers. These materials can be shaped and moulded into various forms, allowing for a wide range of movement and design flexibility. This paper takes the work by Ellis et al. [23,24] as a starting point and extends it to a new application; as such, we use the same materials, Mold Star 15 and Smooth-Sil 950, each produced by mixing binary prepolymer liquid and allowed to cure at 30 °C. This material class requires highly nonlinear material models, and their calibration is a prominent source of uncertainty in modelling the behaviour of soft robots, as noted by both Ellis [23] and Xavier et al. [19].

The principal methods for fabricating soft fluidic actuators include casting and 3D printing [26], supported by a few supplemental techniques, including reinforcement, thin-film manufacturing, and bonding. In cases where the components are small or require tight tolerances, they are produced using soft lithography [27]. In casting, a liquid material is poured into a mould and solidifies, forming the internal cavity. Three-dimensional printing is creating a physical object from a digital model by laying down successive layers of material. In this work, we cast all actuators in 3D printed moulds, which gives a smooth surface finish and tight tolerancing on the final parts [28].

Testing methods for soft robots include visual inspection, mechanical testing, and functional testing. In this work, we use inspection and functional testing and take the mechanical testing results from the literature. Next, we inspect the manufactured parts for defects and geometric non-conformance, with defective actuators excluded from further testing. Finally, the function of the part is measured against a given tip displacement target, measured using photogrammetry or extracted from a computer simulation.

This work is limited to the design of a single pneumatic bending actuator comprised of 15 single chamber units. Each unit comprised an air chamber with height h, width W and thickness t_1 cast in the more compliant Mold Star 15, and a strain limiting layer of thickness t_2 cast in stiffer Smooth-Sil 950. The asymmetric stiffness of each unit causes a biased expansion when subjected to internal pressure. Fifteen of these single chamber units are cast in a single reconfigurable mould proposed by Ellis [29] together, either with the strain limiting layer either in the 'up' or 'down' position, Figure 2.

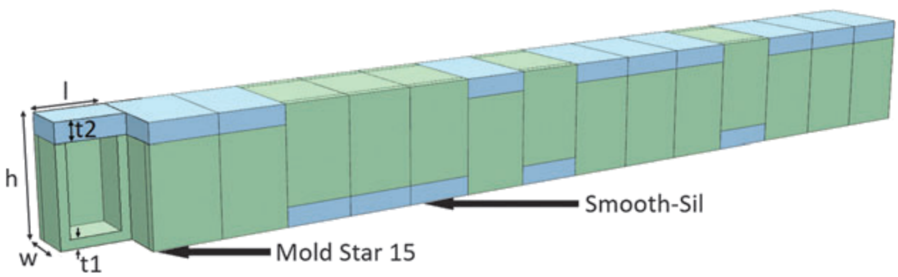


Figure 2. Example of a 15-unit pneumatic network bending actuator, showing the geometric parameters of a single representative unit. Regions cast in Mold Star 15 are shown as green, and Smooth-Sil 950 as blue. Units with the Smooth-Sil 950 strain limiting layer on top are in the 'up' position, and those with the strain limiting layer at the bottom are in the 'down' position.

2.2. Encapsulation, Syllabus, Pandamonium

In a sequence of papers, Lessin et al. [25,30,31] propose breaking down the traditional monolithic optimisation problem posed by traditional generative design into a structured sequence of smaller optimisations in a heuristic method they call "Encapsulation, Syllabus, and Pandamonium". The method consists of three main steps:

1. Encapsulation: dividing the design problem into smaller sub-problems or modules, each with a specific function or target behaviour. Each sub-problem or module is then optimised separately. Once a solution is found for a given encapsulation, that unit can be stored as a complete component and reused in subsequent encapsulations.

2. Syllabus: organising the sub-problems or modules into a structured sequence or curriculum. Modules are sequenced logically to build from simple to complex behaviours or targets.
3. Pandemonium: combining the optimised sub-problems or modules into a final design. The modules compete in a virtual environment, given the overall design objectives and constraints to propose a final design.

The overall goal of this method is to reduce computational time and improve functionality by breaking down the monolithic optimisation problem into smaller, more manageable sub-problems that can be optimised separately. It also allows for a better understanding of the design problem and space, which can help guide the design process. The method relies on computational power to explore the design domain for the designer, who steers the development by structuring the elements of the problem.

For example, consider the problem of designing a soft robot that moves between points on a plane when subjected to an oscillating input pressure. Cheney et al. successfully solve the monolithic problem [32]. However, the solution is challenging to realise practically, and the time taken to find a solution is immense. Therefore, we can instead reframe the problem using the ESP paradigm. We can consider the overall objective of producing a locomoting robot as the highest level of encapsulation and define the robot's behaviour at this level. Figure 3 shows the high-level encapsulation for a locomoting robot and the lower-level encapsulation with arrows showing how solved encapsulations are used as components in higher-level encapsulations. To achieve the overall behaviour, we will follow a syllabus where first, we learn how to produce elementary movements. In this case, a small material region can expand, elongate or shear when subjected to internal pressure. We do this in such a way that we produce a meta-model for each case relating the input factors, such as pressure and scale, to output features, such as deformation.

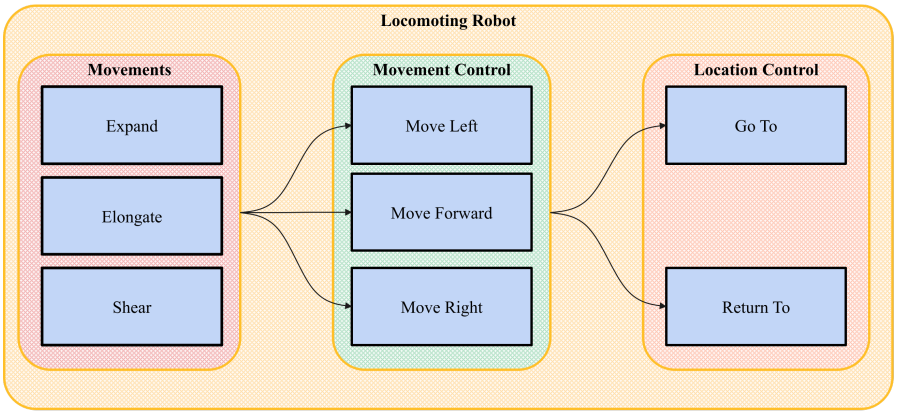


Figure 3. Framing the development of a locomoting robot using ESP. The high-level encapsulation defines the locomoting robot's properties, while lower-level encapsulations such as "expand" are grouped within a given syllabus. Arrows indicate where lower-level encapsulations are used as elements in higher-level encapsulations.

Once we have satisfactory encapsulations for each movement, we can use them as base components in the following "Movement Control" syllabus item. Here we define a domain comprised of "Movement" encapsulation and target movement to the left, right or forward. These encapsulations can then, in turn, become the base components for the last syllabus item of "location control", where the robot learns how to "go to" or "return to" a specified location.

The learning mechanism at each encapsulation level creates multiple candidate solutions and allows these to compete to form a list of high-performing encapsulations. In addition, it is possible to allow several encapsulation variants to maintain variability throughout the development process.

2.3. Reduced-Order Model

A reduced-order model can significantly alleviate the computational requirements of a simulation by removing or combining redundant or unnecessary aspects of the design [33]. For example, nonlinear FEA is a common technique for representing or simulating the behaviour of soft robot elements. It can capture both the time-transient nature of robot motion and the nonlinear material behaviour and large displacements typical in this field. This field does use several alternative order reduction strategies, including linear and nonlinear beam theory [34], eigenvalue analysis and orthogonal decomposition [35], machine learning [36], and inverse kinematics [37].

Ellis et al. [24] simulate each candidate bending actuator using a complete, 3D FE model in *MSC.Marc*. The computational time to simulate each candidate is significant (≈20 min per evaluation), compounded by the need for many such simulations in a generative design environment. Ellis et al. found that by replacing the internal cavity with a tuned low-stiffness material, one can represent the 3D actuator in 2D, significantly reducing simulation time (≈45 s per evaluation). However, the material models must be calibrated for the reduced dimension. The 2D reduced model worked well for the case with only a single geometry but will require additional calibration to include various geometries.

In this paper, we can initially reduce the problem by acknowledging that the motion profile we are working with is 2D and that the actuator is a sequence of replicating units. The reduced order model we require only needs to map the parametric geometry (t_1, t_2, h, W, l, see Figure 2), of a single unit to a change in length (ΔL) and bending angle (θ) for various internal pressure loads (P), Figure 4. We can estimate the behaviour of a 15-unit actuator by summing the behaviour of its component units. Further, we are only interested in the quasi-static behaviour and ignore any time-transient behaviour.

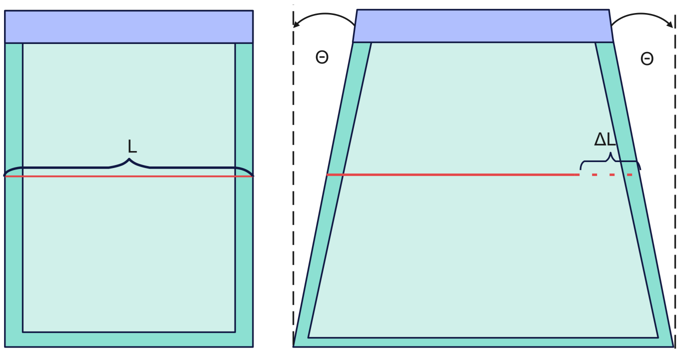


Figure 4. Geometric representation of the reduced-order model. Left is the cross-section of a single unit in a profile. The compliment Mold Star 15 cavity (green) and Smooth-Sil 950 strain-limiting layer (blue) are shown. The reduced model replaced the entire unit with a single line (red) of length L. The right shows the change in geometry experienced by each unit when pressure is applied. The cavity expands significantly, while the strain-limiting layer does not. This is represented by an increase in length ΔL and a change in angle for the two ends of the line Θ.

To determine the relationship between a unit of given parameters and its response, we simulate a sequence of three identical units and measure the response only of the middle unit, Figure 5. The first unit in the sequence has a fully fixed boundary condition, not representative of the general unit, and the last unit is prone to unrepresentative bulging.

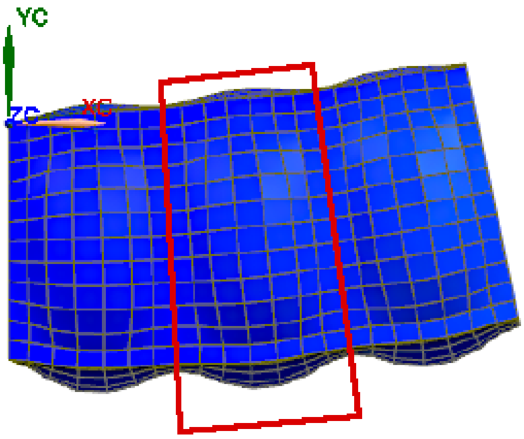


Figure 5. Three unit FE model of a soft bending actuator. The red wedge indicates the centre unit used as input for the reduced-order model.

We construct the FE model in *MSC.Marc* based on the parametric CAD shown in Figure 2; instead of all 15 units, we only model three, and we make use of a longitudinal symmetry plane suggested by Xavier et al. to reduce the computational cost of each simulation [19]. Next, we mesh the geometry in *MSC.Apex* using second-order hexahedral elements with a nominal dimension of 0.5 mm, resulting in a mesh with at least two elements through the thickness of any wall. Finally, for simulation purposes, we represent each material using an Ogden constitutive model [38] with material parameters derived from testing performed per ISO 37:2011 [39] and ISO 7743 [40], as shown in Table 1.

Table 1. Ogden parameters for Mold Star 15 and Smooth-Sil 950 derived from materials testing in accordance with ISO 37:2011 and ISO 7743.

Material	μ_1 **(N mm^{-2})**	α_1	μ_2 **(N mm^{-2})**	α_2	μ_3 **(N mm^{-2})**	α_3
Mold Star 15	-6.503×10^{-6}	-21.32	0.2169	1.180	1.372×10^{-3}	4.884
Smooth-Sil 950	-0.3062	-3.059	0.0283	4.597	6.596×10^{-9}	17.69

The left edge of the three-unit mesh is fully constrained, representing a fully clamped condition, while the right edge remains free. The simulation uses implicit integration in a full Newton–Raphson scheme, with large displacements and follower forces active.

We train a response surface constructed with radial basis functions [41] using a Latin hypercube design of experiments (DOE), resulting in a reconstruction error $R^2 = 0.98$ with 100 training points, as shown in Table 2. Note that the unit length is constant, allowing us to maintain the overall length of the actuator with 15 units for comparison purposes.

Table 2. Parameter ranges used in the DOE for a reduced-order model of a single bending actuator unit. Nominal values were taken from Ellis et al. [24] as $t_1 = 1$ mm, $t_2 = 2$ mm, $h = 17.5$ mm, $W = 15$ mm, and $l = 10$ mm.

Parameter	**Min**	**Max**
Cavity wall thickness t_1	0.5 mm	3.0 mm
Strain limiter thickness t_2	0.5 mm	3.0 mm
Cell height h	15.0 mm	20.0 mm
Cell width W	12.5 mm	17.5 mm
Cell length l	10 mm	10 mm
Pressure P	0.1 bar	1.1 bar

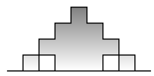
2.4. Lindenmeyer Systems

Lindenmayer systems, also known as L-systems, are a type of formal grammar used to describe the growth of plants and other organisms and the shapes of specific natural structures, such as crystals and snowflakes [42]. It was first described by the Hungarian biologist Aristid Lindenmayer in the 1960s and has been used in various fields, such as biology [43], computer graphics [44], and architectural and graphic design [45]. This paper uses L-systems to encode repeating building block sequences of actuator units. The L-system can be considered a mathematical model of the growth process of the organism or system, with the symbols in the string representing different parts of the organism or system, and the production rules specifying how those parts change over time. An L-system consists of an alphabet of symbols, variables or immutable constants, an initial axiom, and production rules. Starting with the axiom as iteration 0, the production rules specify how to replace each symbol in the produced string during the successive iterations of the L-system. For example, Table 3 shows how an L-system can produce a Koch curve [46]. Table 4 shows the resulting string and visual representation.

Table 3. Construction rules to produce a Koch curve using L-systems. Starting with the axiom "F", at each iteration, "F" is replaced by "F+F–F–F+F". As constants "+" and "–" are not replaced in successive iterations. Including an interpretation layer, we can sketch the Koch curve created. "F" represents a line, "+" represents a right turn in the lines' direction, and "–" represents a left turn in the lines' direction.

Variables:	F
Constants:	+, −
Axiom:	F
Production Rules:	F → F+F–F–F+F

Table 4. Result of three iterations of the presented Koch curve using L-systems.

Iteration	String	Representation
0	F	—
1	F+F–F–F+F	
2	F+F–F–F+F+F+F–F–F+F–F+F–F–F+F–F+F–F–F+F+F+F–F–F+F	

2.5. Markov Chain Monte Carlo Methods

This research requires exploring permutations of a sequence of bending actuator units with different geometries. The eventual response of the final unit in the sequence depends on the response of all previous units. Markov chain Monte Carlo (MCMC) methods are a class of computational algorithms for estimating the properties of complex probability distributions. These algorithms are widely used in Bayesian statistics and machine learning, where they can perform parameter estimation, model selection, and model averaging tasks. These methods can also be used to explore a combinatorial design space in an organised way [47]. Gibbs sampling [48] is a special case of the Metropolis–Hastings algorithm [49] that is used when the target distribution can be written as a product of simpler distributions. It works by sampling from each simpler distribution, one at a time.

In this work, there is only one probabilistic component, whether a particular unit in a sequence is up or down. Using a probability of 50% that a unit will be in the "up" orientation, we can readily explore our design space. We use Gibbs sampling to explore the design space without bias, allowing us to identify high-performing and unique multi-unit combinations for reuse.

2.6. Curve Matching

One key element of this work is the capability to measure the relative performance of two designs in their ability to match a given target curve. The user can use an ordered sequence of points from an analytical equation to represent the target curve. The output curve is constructed from the sequence of unit centroids for a given actuator geometry at a given pressure. The closer the size and shape of the target and output curve, the higher the actuator performance.

Several alternatives are available for measuring the difference between curves, including discrete Fréchet distance [50], dynamic time warping (DTW) [51], and partial curve matching (PCM) [52]. In robotics, PCM is used for grasping, path planning, tracking, and recognising non-rigid objects. PCM is simple to implement and robust. However, it can become computationally expensive as the curves' complexity increases. The basic idea behind PCM is to divide the given curve and the reference curves into smaller segments and compare the segments to find the best match. The segments can be divided into equal parts, or a technique such as DTW can be used to align the segments based on their shape.

In our case, we map each target curve segment to the output curve's corresponding segment. This requires the target and output curves to have the same number of segments. We use simple linear interpolation to divide the target curve into the same number of segments as the output curve if they initially differ. Each pair of segments is then connected to form a quadrilateral. The area of each quadrilateral is then calculated and summed to quantify the total deviation between the two curves. This method provides flexibility because parts of the target or output curves can be used rather than the whole curve when beneficial. Figure 6 shows an example of comparing two curves.

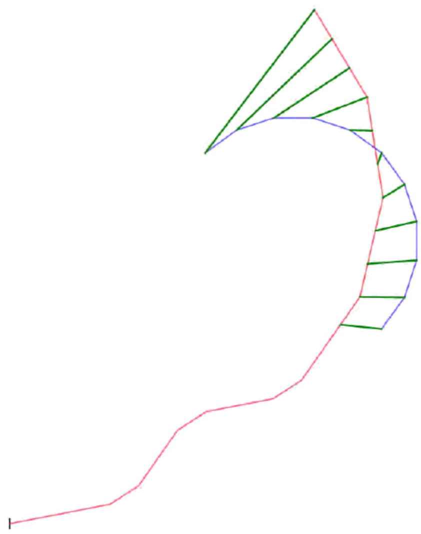


Figure 6. PCM example showing a target curve (blue) and a candidate curve (red). Both the target and candidate curves are discretised and connected into simple quadrilaterals (green). The area of each quadrilateral is calculated and accumulated over all quadrilaterals to produce a performance measure for later optimisation. In this case, the candidate curve is not required to have a complete match with the target curve, so only part of the candidate curve is discretised.

2.7. Optimisation

This paper uses numerical optimisation to minimise the distance between two points or the area between two curves. We use the Euclidian distance ($d_E(x, y)$), where x and y are the initial and target tip coordinates in the first and PCM (PCM_{area}) in the second. Equation (1) describes the optimisation problem for minimising the difference. The parameters and ranges are the same as in Section 2.3, and an additional discrete orientation variable O with states $\uparrow$ or $\downarrow$.

$$\begin{aligned} \text{minimize} \quad & d_E(x,y) \ \text{ or } \ PCM_{area} \\ \text{subject to} \quad & 0.5\,\text{mm} \leq t_1 \leq 3.0\,\text{mm} \\ & 0.5\,\text{mm} \leq t_2 \leq 3.0\,\text{mm} \\ & 15.0\,\text{mm} \leq h \leq 20.0\,\text{mm} \\ & 12.5\,\text{mm} \leq W \leq 17.5\,\text{mm} \\ & 0.1\,\text{bar} \leq P \leq 1.1\,\text{bar} \\ & l = 10.0\,\text{mm} \\ & O \in [\uparrow or \downarrow] \end{aligned} \tag{1}$$

Although gradient information is available from some reduced-order models, we use a non-gradient-based genetic algorithm (GA), [53]. Firstly, it allows for the simple inclusion of discrete variables such as orientation. Secondly, the survival of the fittest heuristic mechanism used by GA's closely matches the intent of the pandamonium phase of the ESP methods described in Section 2.2. Lastly, GA's are better suited to produce a collection of high-performing options rather than a final optimal solution. This paper uses a basic GA with parameters listed in Table 5. That said, we invested little effort in tuning the hyperparameters.

Table 5. Hyperparameters for simple GA used in this paper.

Population size:	250 individuals
Maximum iterations:	1000
Patience:	5 iterations
Elites:	5%
Crossover:	50%
Mutation rate:	5%
Mutation strength:	5%

3. Results and Discussion

Here we present two sets of results. The first shows how we replicate the work by Ellis et al. [24] using ESP, and then we show how we can extend the method to a more complex example.

3.1. Two-Dimensional Bending Actuators with ESP

Ellis et al. assemble 15 identical bending units in either the up (↑) or down (↓) position to target one of four cases using a GA. For reference, the length of the assembled bending actuator is shown in Figure 2 as the horizontal direction (x), with increasing magnitude moving from left to right with a fully clamped condition representing $x = 0$ on the left end of the actuator. The vertical direction (y) is perpendicular to the horizontal with $y = 0$ also at the centre of the clamped left edge of the actuator. Ellis et al. [24] investigates four target cases, Table 6, when a bending actuator is subjected to a given internal pressure.

Table 6. Target cases investigated by Ellis et al. [24] replicated in this paper.

Maximise x-position of the free tip	(max horizontal)
Minimise x-position of the free tip	(min horizontal)
Maximise y-position of the free tip	(max vertical)
Minimise the distance between fixed and free ends	(min radius)

Figure 7 shows our approach using the ESP framework. Here we start with two encapsulations that represent pre-built modules used in later training. These two encapsulations are a parametric FE model of a three-unit bending actuator (three-unit FE analysis) and a dataset containing the ranges for each parameter in a single unit (Unit Parameters). These encapsulations are somewhat independent and can be swapped for encapsulations with similar properties easily. However, they represent a necessary starting position and tools that are not learnt within the ESP syllabus in this case. The syllabus consists of three modules, "represent bending unit", "represent bending actuator" and "target measure". The "represent bending unit" module is tasked with learning the bending response of a single bending unit within the parameter range contained within defined parameter ranges, using the method described in Section 2.3, resulting in a model of a single bending unit (bending unit). The "represent bending actuator" module learns how to assemble the bending units into a predicted performance for a full actuator (bending actuator). The "target measure" module consists of learning the best configurations for each of the four target measures. For this, we need an encapsulation containing the definitions for the desired performance measures (performance measures).

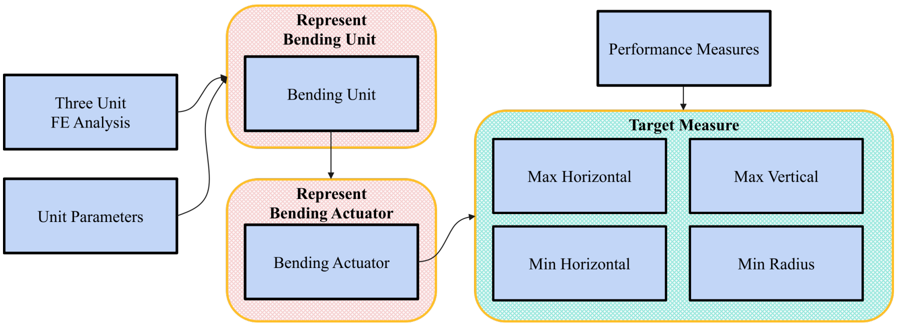


Figure 7. ESP framework for generating candidate designs for various pneumatic bending actuators to replicate the work of Ellis et al. [24].

This paper replicates the unit orientation results of Ellis et al., as shown in Table 7. We further show that if we reverse each unit in the four results, there is a second viable configuration for each case, something not shown in Ellis's work. Finally, Table 8 compares the tip displacement measured and simulated by Ellis et al. with those found using ESP. In each case, the results of the ESP method proposed here show close conformance to those previously presented, as visually confirmed in Figure 8. It is important to note that by making use of the pre-trained reduced order models resulting from the "Three Unit FEA" and "Unit Parameters" encapsulations, the optimisation time for each of the four target measures in the "Target Measure" module takes less than 1 s compared to a single function evaluation using Ellis et al.'s reduced-order model taking around 45 s on the same hardware. A single function evaluation of the "Three Unit FE Analysis" encapsulation used in training takes around 40 s.

Table 7. Resulting unit orientations for each unit in sequence from left to right targeting each of the four cases described in Table 6. Note that in all four cases, if each is reversed, they still produce the same result, though the results will be visually reflected about the x-axis.

Case	Configuration
max horizontal	[↓↑↓↑↓↑↑↓↑↓↑↓↑↓↑]
min horizontal	[↑↑↑↑↑↑↑↑↑↑↓↓↑↓↓]
max vertical	[↑↑↑↑↑↓↑↓↑↓↑↓↑↓↑]
min radius	[↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓]

Table 8. Comparison of the results measured and simulated by Ellis et al. [24] and generated using the ESP as proposed.

	Ellis et al. [24] Measured	Ellis et al. [24] Simulated	ESP Simulated
max horizontal	225.2 mm	232.1 mm	222.0 mm
min horizontal	−101.2 mm	−99.9 mm	−97.8 mm
max vertical	204.6 mm	211.2 mm	201.4 mm
min radius	10.4 mm	16.8 mm	16.5 mm

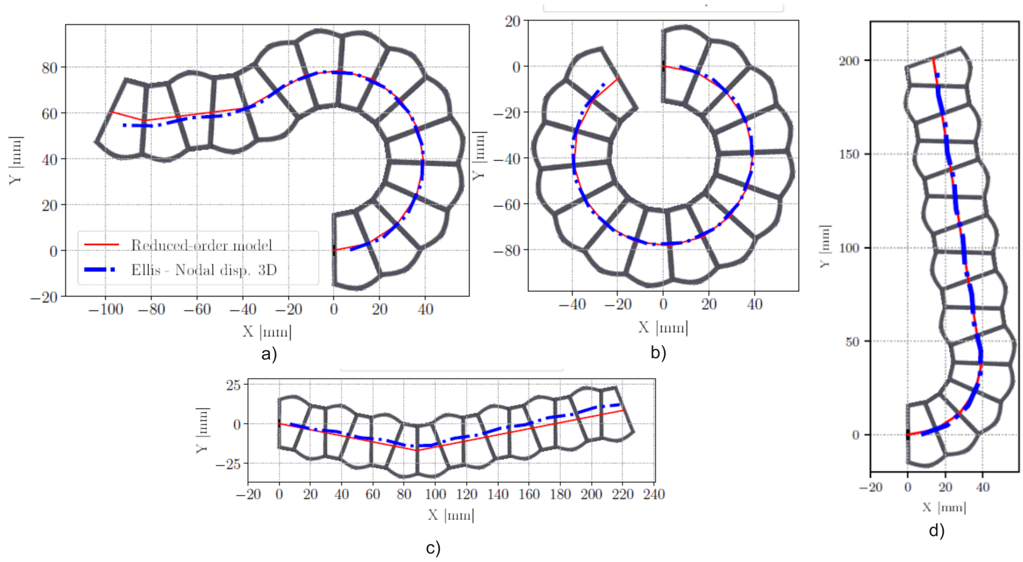


Figure 8. Visual representations of the resulting deformed bending actuators for each case described in Table 6. (**a**) Min horizontal, (**b**) min radius, (**c**) max horizontal, (**d**) max vertical. It should be noted that the result for the simple linear extension case (**d**) has an unexpected shape with a "kink" in the middle. This results from the fixed unit's mounting angle, which is corrected by the mid-span "kink".

3.2. Multi-Gripper Tentacle with ESP

In Section 3.1, we frame an established design problem within the ESP framework and found that in conjunction with the reduced-order model proposed in Section 2.3, we achieve comparable results in less time compared to using GA and FE simulations on their own. We now aim to use the ESP framework to create a tentacle-like soft robot that can articulate and grip multiple objects.

We have already constructed two useful encapsulations, (bending unit) and (bending actuator) in Section 3.1 that generate the response of various bending units and predict the behaviour of various bending unit assemblies. Since these are already trained and self-contained, we can simply use them in this project. To advance from these building blocks to a full tentacle, we have divided the design task into learning modules, as shown in Figure 9.

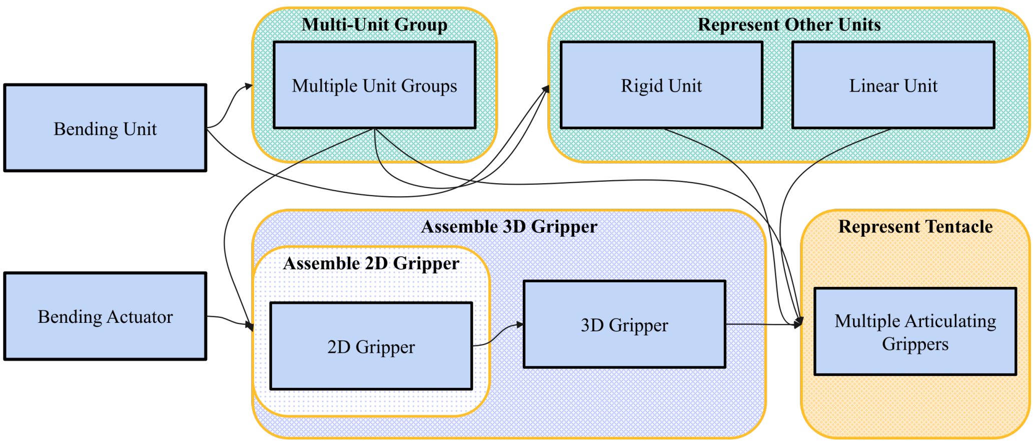


Figure 9. ESP framework for generating candidate designs for a multi-gripper tentacle.

We know that some combinations of single units produce a cumulative effect different from that of a single unit. For example, a pressurised single unit forms a wedge shape. In contrast, a sequence of wedged units with the same orientation causes the assembly to curl, and a sequence of units with alternating orientation causes linear displacement. The "Multi-Unit Group" module explores and identifies unique combinations of units that produce alternate behaviours, such as a shallower bending angle, or no bending at all. These (multiple-unit groups) are not fully functional actuators but rather a larger pool of primitive components.

In the "multi-unit group" module, we use MCMC methods described in Section 2.5 to explore and identify unique combinations of single units. Our objective is to find unique accumulations of tip displacement (Δl) and bending angle (θ). We normalised the response of the multi-unit groups by the number of units in the assembly to favour smaller groups. This process combines elements of classification and selection, as we are generating a list of assemblies that perform a given displacement while acknowledging that multiple variations produce similar cumulative results. The resulting multiple-unit groups are then encapsulated as primitive components with their own behaviour for later modules. Figure 10 shows a few examples of multiple-unit groups. It is important to note the stochastic nature of the process does not guarantee the same number of groups every time the algorithm runs or that the same groups will be found. This is in no way detrimental to the design process as the groups are potential options for later encapsulations.

Figure 10. Examples of multiple-unit groups found using MCMC methods. The red centre lines show the response of an initially straight reduced-order model to applied pressure. The wedge shape of the inflated bending unit is included for clarity.

The "Assemble 3D Gripper" combines the 2D Grippers learnt in the nested "Assemble 2D Gripper" module to produce a 3D Gripper capable of lifting volumetric objects. Here we use a nested module rather than two sequential modules to improve the reusability of the "Assemble 3D Gripper" module in later syllabi by including the ability to learn its own components rather than being limited to simply assembling previously generated elements.

In the "Assemble 2D Gripper" module, we learn how to assemble units and multi-unit groups to wrap around various 2D shapes. We start with the idea that construction follows a somewhat biological growth model, which can be easily scaled depending on the application. This means that a gripper well suited to gripping circles should look much the same regardless of the circles' size. We begin with a simplified 2D case, encapsulating the gripper designs suitable for whichever object shape. Then, we use these 2D Grippers as elements in a higher-level encapsulation to generate 3D Gripper encapsulations.

With efficient access to encapsulated single and multi-unit groups, we can start targeting various profiles to be gripped. We use L-Systems, discussed in Section 2.4, to learn an encoding for recursively generating scale-invariant assemblies that meet our objective of gripping an object in a particular state. For simplicity, we have a fixed starting point and assume the actuator lies in a straight line when noninflated. Table 9 shows the L-system constructed to produce candidate 2D grippers. Here we again use a GA to learn the optimal production rules for a given target shape. The GA can replace any three-letter encoding with between four and ten randomly selected encodings and orientation symbols from the alphabet (shown as wildcard *).

Table 9. L-systems construction for generating 2D gripper encodings. The L-system is constructed with an alphabet, including letter encodings for each of the single units or multi-unit group ([aaa, . . . , zzz]), then two characters to indicate the orientation of the succeeding unit (↑ and ↓). A starting point (XXX) is added as a constant to identify the global location of the assembly and the fixed boundary condition. The axiom starts with the starting point constant, followed by a random three-character code from the alphabet.

Alphabet:	aaa, . . . , zzz, ↑, ↓
Constants:	XXX
Axiom:	XXX***
Production Rules:	*** → *********
	. . .

As target shapes, we generated circles, triangles, and squares at various distances from the starting point and created an offset shape half the height of the first unit away. We then made use of this offset shape as our target curve. If we can approximate the target shape with our bending actuator, we consider this to have "gripped" the shape. Figure 11 shows the results of this gripper design process for three square targets at three distances and scales, using the same learnt L-system. Take note of the geometric similarity of the generated actuator at each scale.

As you can see, the curve matching is imperfect, but the square is sufficiently surrounded to be considered "gripped". This is an excellent example of how the designer is involved with decision-making through the ESP process. If the designer is satisfied with the results of a given encapsulation, they are free to use it in later modules. The 2D "gripper" is extended to the 3D case with a simple encapsulation, including a rigid centre and the option to attach n 2D grippers radially about the periphery of the rigid centre. Simply testing a few variants leads to an encapsulation of three gripper units spaced 120° apart, as shown in Figure 12.

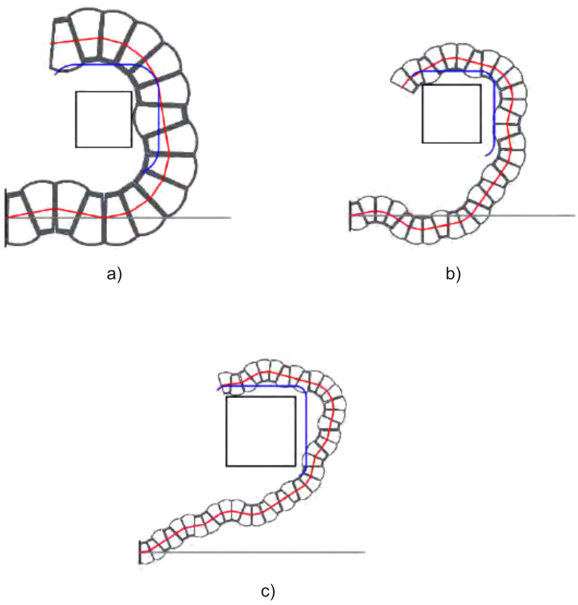


Figure 11. Three examples of 2D grippers for square targets on various scales using the same L-system encoding. Subfigure (**a**) shows a 16-unit actuator, while subfigures (**b**,**c**) show 31- and 39-unit configurations.

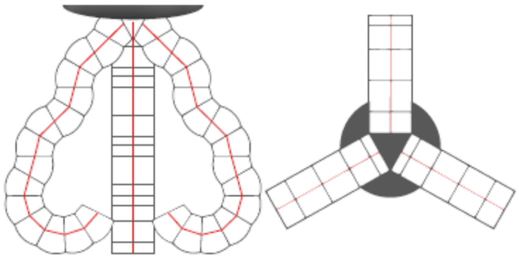


Figure 12. A 3D gripper resulting from the radial assembly of three 2D gripper actuators.

Before we attempted to generate the full tentacle, we saw a need to include a few additional actuator options and created a "represent other units" module to learn a rigid unit and linear unit. Finally, in the last module, "represent tentacle", we combine all earlier encapsulations to solve the overall design task of generating an actuating tentacle with multiple grippers, as shown in Figure 13. Producing this design using the knowledge encapsulation produced in the first case study means that this result was produced in less than 30 s. An advantage of the divide-and-conquer approach of ESP is that various methods can be used to accomplish each encapsulation task.

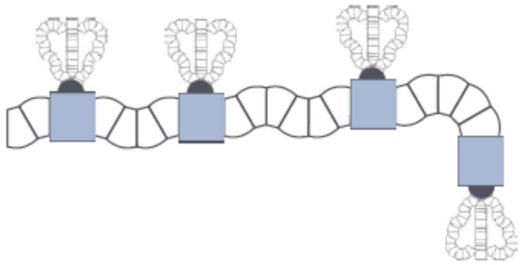


Figure 13. A four-gripper articulating tentacle resulting from applying ESP.

4. Conclusions and Future Work

The research presented uses ESP as a framework suited to exploring and designing soft robots. The generic framework allows designers significant flexibility in achieving and connecting various encapsulations. This paper highlights this by using FE modelling, reduced-order kinematic models, L-systems, and MCMC and PCM methods to design a soft robot that can articulate and grip multiple objects. It is, however, important to keep in mind that any methods shown here are chosen by convenience, and other methods can be used to generate encapsulations on the condition that the encapsulation input and output are clearly defined.

The ESP framework divides the design task into smaller, manageable modules, and various methods are used to accomplish each encapsulation task. The multi-unit group module uses MCMC to explore and identify unique combinations of single units. L-systems are used to recursively generate scale-invariant assemblies that meet the objective of gripping an object. The reduced-order model within the ESP framework proposed in the paper can replicate the results of previous work with improved computational efficiency. The research also extends the method to a more complex problem of creating a tentacle-like soft robot with multiple grippers.

Moving from the 15-unit 3D FE model previously constructed to the proposed kinematic reduced-order model reduces the function evaluation time in the optimiser from $\approx$20 min to less than 1 s with no observable difference in the quality of the result. This first set of results quantitatively shows that the proposed method produced similar results to an established benchmark.

The model is extended to show how it could generate a design given a more complex set of requirements resulting in a practical design for a multi-gripper tentacle. It is important to remember that the purpose of the proposed design tool is not to provide a single optimum but rather a reasonable result for the designer to evaluate. This second set of results qualitatively highlights the benefits of the method.

The method could be further expanded into a 3D space without changing the conceptual framework, but we would need to change the reduced-order model. Currently, the reduced order model is defined and trained for two degrees of freedom, and at least one more would need to be included to account for movement in an additional dimension. For example, an additional angle change can be defined for the reduced-order model, and a series of FE simulations with movement in 3D can be generated for training. The optimisation routine and partial curve matching will remain the same.

We propose measuring contact force and system power in the simulation environment and reduced-order models in future work. In addition to these incremental improvements already discussed, the modular nature of ESP and the direct integration of control and behaviour elements shown by Lessin et al. [25] means that the method can easily integrate the mechanical and control behaviours of flexible sensors [54,55] and environmental simulators, such as [20,56]. All that is required to integrate sensor and control elements into the design process is to include encapsulations for each component and construct the control environment to utilise additional inputs. Access to this information during the design optimisation routine will allow for designing behaviours with feedback responses from a virtual environment.

Overall, this paper contributes a practical generative design process that significantly reduces the design time for soft robots while producing comparable results to existing state-of-the-art methods. Our approach meaningfully involves the human designer in the design process. It enables the inclusion of other numerical techniques, ultimately opening up new avenues for exploring the vast design space of soft robots.

Author Contributions: Conceptualization, M.P.V.; methodology, M.P.V. and I.J.J.; software, M.P.V. and I.J.J.; validation, M.P.V. and I.J.J.; formal analysis, M.P.V.; investigation, I.J.J.; resources, M.P.V.; data curation, M.P.V. and I.J.J.; writing—original draft preparation, M.P.V.; writing—review and editing, M.P.V.; visualization, M.P.V. and I.J.J.; supervision, M.P.V.; project administration, M.P.V.; funding acquisition, M.P.V. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Research Foundation of South Africa Grant Number 129381.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Rus, D.; Tolley, M. Design, fabrication & control of soft robots. *Nature* **2015**, *521*, 467–475. [PubMed]
2. Laschi, C.; Mazzolai, B.; Cianchetti, M. Soft robotics: Technologies and systems pushing the boundaries of robot abilities. *Sci. Robot.* **2016**, *1*, eaah3690. [CrossRef] [PubMed]
3. Armanini, C.; Boyer, F.; Mathew, A.; Duriez, C.; Renda, F. Soft robots modeling: A structured overview. *IEEE Trans. Robot.* **2023**, 1–21. [CrossRef]
4. Pagoli, A.; Chapelle, F.; Corrales-Ramon, J.; Mezouar, Y.; Lapusta, Y. Review of soft fluidic actuators: Classification and materials modeling analysis. *Smart Mater. Struct.* **2022**, *31*, 013001. [CrossRef]
5. Li, M.; Pal, A.; Aghakhani, A.; Pena-Francesch, A.; Sitti, M. Soft actuators for real-world applications. *Nat. Rev. Mater.* **2022**, *7*, 235–249. [CrossRef]
6. Gifari, M.; Naghibi, H.; Stramigioli, S.; Abayazid, M. A review on recent advances in soft surgical robots for endoscopic applications. *Int. J. Med. Robot. Comput. Assist. Surg.* **2019**, *15*, e2010. [CrossRef] [PubMed]
7. Zhang, C.; Zhu, P.; Lin, Y.; Jiao, Z.; Zou, J. Modular soft robotics: Modular units, connection mechanisms, and applications. *Adv. Intell. Syst.* **2020**, *2*, 1900166. [CrossRef]
8. Arnold, T.; Scheutz, M. The tactile ethics of soft robotics: Designing wisely for human-robot interaction. *Soft Robot.* **2017**, *4*, 81–87. [CrossRef] [PubMed]
9. Das, A.; Nabi, M. A review on soft robotics: Modeling, control and applications in human-robot interaction. In Proceedings of the 2019 International Conference on Computing, Communication, and Intelligent Systems, Greater Noida, India, 18–19 October 2019; pp. 306–311.
10. Mazzolai, B.; Kraus, T.; Pirrone, N.; Kooistra, L.; Simone, A.; Cottin, A.; Margheri, L. Towards new frontiers for distributed environmental monitoring based on an ecosystem of plant seed-like soft robots. In Proceedings of the 2021 Conference on Information Technology for Social Good, Rome, Italy, 9–11 September 2021; pp. 221–224.

11. Youssef, S.; Soliman, M.; Saleh, M.; Mousa, M.; Elsamanty, M.; Radwan, A. Underwater soft robotics: A review of bioinspiration in design, actuation, modeling, and control. *Micromachines* **2022**, *13*, 110. [CrossRef] [PubMed]
12. Stuttaford-Fowler, A.; Samani, H.; Yang, C. Biomimicry in soft robotics actuation and locomotion. In Proceedings of the 2022 International Conference on System Science and Engineering, Taichung, Taiwan, 26–29 May 2022; pp. 17–21.
13. El-Atab, N.; Mishra, R.; Al-Modaf, F.; Joharji, L.; Alsharif, A.; Alamoudi, H.; Diaz, M.; Qaiser, N.; Hussain, M. Soft actuators for soft robotic applications: A review. *Adv. Intell. Syst.* **2020**, *2*, 2000128. [CrossRef]
14. Ilievski, F.; Mazzeo, A.; Shepherd, R.; Chen, X.; Whitesides, G. Soft robotics for chemists. *Angew. Chem.* **2011**, *123*, 1930–1935. [CrossRef]
15. Shepherd, R.; Ilievski, F.; Choi, W.; Morin, S.; Stokes, A.; Mazzeo, A.; Chen, X.; Wang, M.; Whitesides, G. Multigait soft robot. *Proc. Natl. Acad. Sci. USA* **2011**, *108*, 20400–20403. [CrossRef] [PubMed]
16. Mosadegh, B.; Polygerinos, P.; Keplinger, C.; Wennstedt, S.; Shepherd, R.; Gupta, U.; Shim, J.; Bertoldi, K.; Walsh, C.; Whitesides, G. Pneumatic networks for soft robotics that actuate rapidly. *Adv. Funct. Mater.* **2014**, *24*, 2163–2170. [CrossRef]
17. Lipson, H. Challenges and opportunities for design, simulation, and fabrication of soft robots. *Soft Robot.* **2014**, *1*, 21–27. [CrossRef]
18. Chen, F.; Wang, M. Design optimization of soft robots: A review of the state of the art. *IEEE Robot. Autom. Mag.* **2020**, *27*, 27–43. [CrossRef]
19. Xavier, M.; Fleming, A.; Yong, Y. Finite element modeling of soft fluidic actuators: Overview and recent developments. *Adv. Intell. Syst.* **2021**, *3*, 2000187. [CrossRef]
20. Bhatia, J.; Jackson, H.; Tian, Y.; Xu, J.; Matusik, W. Evolution gym: A large-scale benchmark for evolving soft robots. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2021; Volume 34, pp. 2201–2214.
21. Lai, G.; Leymarie, F.; Latham, W.; Arita, T.; Suzuki, R. Virtual creature morphology—A review. *Comput. Graph. Forum* **2021**, *40*, 659–681. [CrossRef]
22. Runge, G.; Raatz, A. A framework for the automated design and modelling of soft robotic systems. *CIRP Ann.* **2017**, *66*, 9–12. [CrossRef]
23. Ellis, D.; Venter, M.; Venter, G. Computational design for inflated shape of a modular soft robotic actuator. In Proceedings of the 2019 IEEE International Conference on Soft Robotics, Seoul, Republic of Korea, 14–18 April 2019.
24. Ellis, D.; Venter, M.; Venter, G. Generative design procedure for embedding specified planar behavior in modular soft pneumatic actuators. *Soft Robot.* **2022**, *9*, 552–561. [CrossRef]
25. Lessin, D.; Fussell, D.; Miikkulainen, R. Open-ended behavioral complexity for evolved virtual creatures. In Proceedings of the 2013 Genetic and Evolutionary Computation Conference, Amsterdam, The Netherlands, 6–10 July 2013; pp. 335–342.
26. Schmitt, F.; Piccin, O.; Barbé, L.; Bayle, B. Soft robots manufacturing: A review. *Front. Robot. AI* **2018**, *5*, 84. [CrossRef] [PubMed]
27. Rogers, J.; Nuzzo, R. Recent progress in soft lithography. *Mater. Today* **2005**, *8*, 50–56. [CrossRef]
28. Smooth-On. ISO 7743: Mold Making and Casting Materials. Available online: https://www.smooth-on.com/ (accessed on 1 April 2017).
29. Ellis, D. Generative Design Procedure for Embedding Complex Behaviour in Pneumatic Soft Robots. Ph.D. Thesis, Stellenbosch University, Stellenbosch, South Africa, 2020.
30. Lessin, D.; Fussell, D.; Miikkulainen, R. Trading control intelligence for physical intelligence: Muscle drives in evolved virtual creatures. In Proceedings of the 2014 Genetic and Evolutionary Computation Conference, Vancouver, BC, Canada, 12–16 July 2014; pp. 705–712.
31. Lessin, D.; Fussell, D.; Miikkulainen, R.; Risi, S. Increasing behavioral complexity for evolved virtual creatures with the ESP methods. *arXiv* **2015**, arXiv:1510.07957.
32. Cheney, N.; Maccurdy, R.; Clune, J.; Lipson, H. Unshackling evolution: Evolving soft robots with multiple materials and a powerful generative encoding. *ACM SIGEVOlution* **2014**, *7*, 11–23. [CrossRef]
33. Chenevier, J.; González, D.; Aguado, J.; Chinesta, F.; Cueto, E. Reduced-order modeling of soft robots. *PLoS ONE* **2018**, *13*, e0192052. [CrossRef]
34. Du, T.; Hughes, J.; Wah, S.; Matusik, W.; Rus, D. Underwater soft robot modeling and control with differentiable simulation. *IEEE Robot. Autom. Lett.* **2021**, *6*, 4994–5001. [CrossRef]
35. Goury, O.; Duriez, C. Fast, generic, and reliable control and simulation of soft robots using model order reduction. *IEEE Trans. Robot.* **2018**, *34*, 1565–1576. [CrossRef]
36. Thuruthel, T.; Falotico, E.; Renda, F.; Laschi, C. Model-Based Reinforcement Learning for Closed-Loop Dynamic Control of Soft Robotic Manipulators. *IEEE Trans. Robot.* **2019**, *35*, 127–134. [CrossRef]
37. Fang, G.; Matte, C.; Scharff, R.; Kwok, T.; Wang, C. Kinematics of soft robots by geometric computing. *IEEE Trans. Robot.* **2020**, *36*, 1272–1286. [CrossRef]
38. Ogden, R. *Non-Linear Elastic Deformations*; Courier Corporation: North Chelmsford, MA, USA, 1997.
39. *ISO 37:2017*; Rubber, Vulcanized or Thermoplastic—Determination of Tensile Stress-Strain Properties. International Organization for Standardization: Geneva, Switzerland, 2017.
40. *ISO 7743:2017*; Rubber, Vulcanized or Thermoplastic—Determination of Compression Stress-Strain Properties. International Organization for Standardization: Geneva, Switzerland, 2017.
41. Fang, H.; Horstemeyer, M. Global response approximation with radial basis functions. *Eng. Optim.* **2006**, *38*, 407–424. [CrossRef]

42. Lindenmayer, A. Mathematical models for cellular interactions in development I. Filaments with one-sided inputs. *J. Theor. Biol.* **1968**, *18*, 280–299. [CrossRef]
43. Lucas, M.; Laplaze, L.; Bennett, M. Plant systems biology: Network matters. *Plant Cell Environ.* **2011**, *34*, 535–553. [CrossRef] [PubMed]
44. Ebert, D. Advanced Modeling Techniques for Computer Graphics. *ACM Comput. Surv.* **1996**, *28*, 154–156. [CrossRef]
45. Wonka, P.; Wimmer, M.; Sillion, F.; Ribarsky, W. Instant architecture. *ACM Trans. Graph.* **2003**, *22*, 677–699. [CrossRef]
46. Lakhtakiatt, A.; Varadantt, V.; Messier, R.; Varadantt, V. Generalisations and randomisation of the plane Koch curve. *J. Phys. Math. Gen.* **1987**, *20*, 3537–3541. [CrossRef]
47. Xiong, Y.; Duong, P.; Wang, D.; Park, S.; Ge, Q.; Raghavan, N.; Rosen, D. Data-driven design space exploration and exploitation for design for additive manufacturing. *J. Mech. Des. Trans. ASME* **2019**, *141*, 101101. [CrossRef]
48. Gelfand, A. Gibbs sampling. *J. Am. Stat. Assoc.* **2000**, *95*, 1300–1304. [CrossRef]
49. Chib, S.; Greenberg, E. Understanding the Metropolis–Hastings algorithm. *Am. Stat.* **1995**, *49*, 327–335.
50. Dowson, D.; Landau, B. The Fréchet distance between multivariate normal distributions. *J. Multivar. Anal.* **1982**, *12*, 450–455. [CrossRef]
51. Bemdt, D.; Clifford, J. *Using Dynamic Time Warping to Find Patterns in Time Series*; New York University: New York, NY, USA, 1994.
52. Jekel, C.; Venter, G.; Venter, M.; Stander, N.; Haftka, R. Similarity measures for identifying material parameters from hysteresis loops using inverse analysis. *Int. J. Mater. Form.* **2019**, *12*, 355–378. [CrossRef]
53. Holland, J. *Adaptation in Natural and Artificial Systems*; MIT Press: Cambridge, MA, USA, 1975.
54. Mishra, R.; Al-Modaf, F.; Babatain, W.; Hussain, A.; El-Atab, N. Structural engineering approach for designing foil-based flexible capacitive pressure sensors. *IEEE Sens. J.* **2022**, *22*, 11543–11551. [CrossRef]
55. Thuruthel, T.; Shih, B.; Laschi, C.; Tolley, M. Soft robot perception using embedded soft sensors and recurrent neural networks. *Sci. Robot.* **2019**, *4*, eaav1488. [CrossRef] [PubMed]
56. Schegg, P.; Ménager, E.; Khairallah, E.; Marchal, D.; Dequidt, J.; Preux, P.; Duriez, C. SofaGym: An open platform for reinforcement learning based on soft robot simulations. *Soft Robot.* **2022**. [CrossRef] [PubMed]