MakeSense: Automated Sensor Design for Proprioceptive Soft Robots

Javier Tapia[1,2], Espen Knoop[1], Mojmir Mutný[1], Miguel A. Otaduy[2], Moritz Bächer[1]

[1]Disney Research, [2]Universidad Rey Juan Carlos

***Abstract*—Soft robots have applications in safe human-robot interactions, manipulation of fragile objects, and locomotion in challenging and unstructured environments. In this paper, we present a computational method for augmenting soft robots with proprioceptive sensing capabilities. Our method automatically computes a *minimal* stretch-receptive sensor network to user-provided soft robotic designs, which is optimized to perform well under a set of user-specified deformation-force pairs. The sensorized robots are able to reconstruct their full deformation state, under interaction forces. We cast our sensor design as a sub-selection problem, selecting a minimal set of sensors from a large set of fabricable ones which minimizes the error when sensing specified deformation-force pairs. Unique to our approach is the use of an analytical gradient of our reconstruction performance measure with respect to selection variables. We demonstrate our technique on a bending bar and gripper example, illustrating more complex designs with a simulated tentacle.**

I. Introduction

The field of soft robotics has grown rapidly in recent years, with a large body of work focusing on the development of fundamental technologies for sensorizing [1,2,3,4,5,6], actuating [7,8,9,10,11], and fabricating [12,13,14,15] soft robots. These technologies are enabling functional materials and fully-functional robots [16,17,18,19] that show potential in applications including deep-sea exploration [20], search-and-rescue [21,22], surgical robotics [23], grasping [24], and prosthetics and wearable devices [25,26,27]. The use of soft materials makes these robots inherently safe. By exploiting morphological computation [28], simpler control strategies are sufficient to reliably grasp and manipulate deformable objects of varying shape, size, and stiffness, or to locomote in unpredictable terrains, trapped with obstacles.

Thus far, many design tasks are carried out manually by skilled researchers. However, as the field matures further, and the tackled problems increase in complexity, it will become increasingly important to leverage computational tools in order to automate or guide challenging design tasks.

To enable the design of *autonomous* soft robotic systems, sensory information is key. Looking to nature, sensory nerve fibers wound around our muscle fibers like springs, serve as stretch receptors [29]. Their primary role is to detect length changes of muscles, allowing us to *sense* the posture our body is in. Augmenting soft robots with comparable *proprioceptive sensing* capabilities is desirable but challenging.

A large body of previous work has focused on the *fabrication* of soft robots with embedded channels filled with conductive liquid, signaling resistance changes when deformed. Several special-purpose sensors and sensorized soft robots or robotic components exist [30]. However, they are manually designed.

The manual sensorization of complex soft robot designs is a hopeless endeavor. Resistance is an integrated quantity. Hence, we cannot tell where along a sensor channel a robot stretches (positive signal change) or compresses (negative signal change). Deciding *how many* sensors one would need to reliably reconstruct the many degrees of freedom of a deforming soft robot and *where* to best place such sensors are two non-trivial questions. Further complicating matters, soft robots deform due to actuation *and* unknown interactions.

To navigate this unintuitive design space, we propose a computational technique that automates the placement and sizing of a stretch-receptive sensor network, adding proprioceptive capabilities to soft robotic designs of arbitrary shape and size. The resulting soft robots are capable of sensing both, their internal deformation state *and* external interactions.

Provided with a soft robot design and expected interactions, we first simulate the behavior of the soft robot under specified interactions, serving us as input. We rely on hyperelastic material models and a finite element (FE) discretization to simulate the nonlinear behavior of the silicones or urethanes that soft robots are commonly made of. Our ultimate goal is to add a sensory network that minimizes the difference between deformations under unknown interactions and their reconstructions from sensor signals.

To avoid a discrete optimization problem, we first initialize a large set of *fabricable* sensors, casting the sensor design as a sub-selection problem. Starting with a single sensor, we incrementally add the sensor we expect to increase the reconstruction performance the most. To do so, we compute an analytical gradient with respect to binary inclusion variables that indicate which sensors are part of the selected set. To evaluate this gradient, we propose to mathematically *nest* our simulation, sensing, and design problems into one optimization formulation. Accounting for the incompressibility of the conductive liquid when modeling our sensors, we can accurately predict readings in simulations.

We validate our approach on a fabricated bar undergoing bending, which is the predominating deformation mode in soft robotics. To demonstrate the effectiveness of our sub-selection strategy on industrial use cases, we add proprioceptive capabilities to a pneumatically-actuated gripper, sensing the deformation state under unknown contact forces from manipulation of a cylinder. With a simulated tentacle example, we illustrate applications on bioinspired designs.

II. Related Work

Sensing methods for soft robots is an active area of research, and we refer to a recent review [30] for an overview of sensing

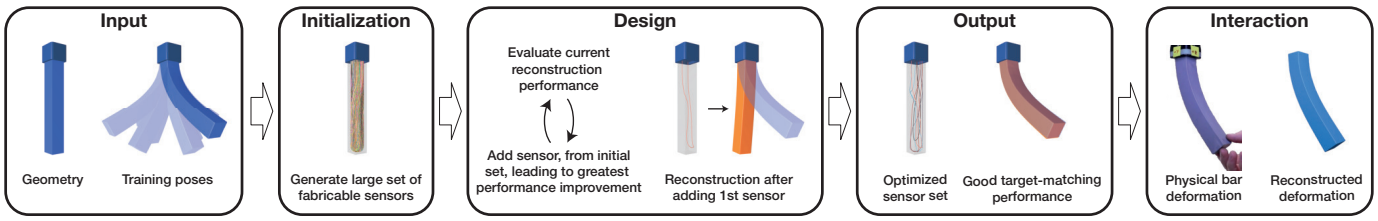

Fig. 1. Automated Sensor Design. A soft robot design together with expected interactions are specified as input (Input). From a large initial set of sensors (Initialization), a minimal subset is selected (Output), minimizing the difference between deformations under expected interactions and their reconstruction from simulated sensor signals (Design). A soft robot with embedded sensors is fabricated as a result, and we compare real interactions with deformations reconstructed from real sensor signals (Interaction).

aspects. Recent reviews of soft fluidic actuators[31] and of soft grippers[19] have highlighted the importance of sensing and proprioception for soft robots.

a) Sensor design: There are few examples of automated systems for designing sensor layouts for proprioceptive soft objects. Culha et al.[4] present one approach, where sensors are routed along a surface by considering the strain field of target deformations. However, they do not attempt to reconstruct the deformed object shape from the sensor readings. Wall et al.[5] propose an approach for sensorizing a soft robotic finger, by building a prototype with a redundant set of sensors and then selecting a subset of these. However, the need for physical prototypes in the design process limits the scalability of this approach. Bächer et al.[32] presented a method for assisting a user with sensorizing soft objects, but this requires the user in the loop and is not fully automated.

b) Soft Robot Sensors: Again referring to a recent review[30] for details, a number of different approaches have been taken for sensorizing soft robots. The use of liquid metal strain sensors for soft robot devices was proposed by Park et al.[33], and there has been much work in this area including integration into actuated soft robots for proprioception[34,35,36,6].

It has been shown that embedded 3D printing can be used to produce such liquid metal sensors[37]. Building on work using embedded 3D printing for fabrication of soft robots[18], it was recently shown that liquid metal sensors and fluidic actuation can be combined in order to print a soft sensorized actuator[38]. Such approaches could enable robots with significantly higher numbers of sensors than would be feasible with manual fabrication methods, for which an automated design approach becomes increasingly important.

Alternatives to liquid metal for strain sensing include ionic liquids[39] and conductive polymers[32,4]. Moreover, capacitive[2,40] and inductive[41] methods for soft strain sensing exist. We note that our approach taken here is not specific to the type of strain sensor used, and could be readily applied to these sensor paradigms.

Aside from strain sensors, different sensing modalities including pressure sensing[42], electric field tomography[43] and acoustic sensing[44] have been researched. However, these methods would seem to provide insufficient sensory information for performing full pose reconstructions. Vision-based approaches[45,46] can provide detailed pose information, but rely on an unoccluded view of the deformed object to be tracked and therefore do not provide a solution that generalizes to arbitrary shapes. Looking at proprioception of actuators, Helps and Rossiter[47] propose to use a conductive working fluid to enable proprioceptive fluidic actuators, but again the method as presented does not provide sufficient information to enable full robot state reconstruction.

c) Fabrication-Oriented Design: In the computer graphics community, there has recently been substantial interest in the development of either fully automated or semi-automated design tools, leveraging simulation and optimization in order to facilitate different design tasks. Closely related to our work are techniques that solve for a set of optimal design parameters, constraining physical systems to equilibria during optimizations. Applications include the design of mechanical[48], deformable[49], and inflatable[50] characters, compliant[51] and cable-driven[52] mechanisms, or flexible rod meshes[53]. While we focus on sensing instead of cable-driven actuation, we share with some works the goal of selecting a sparse subset from a larger set[49,51]. However, in contrast to previous work, we formulate the sub-selection problem by nesting a simulation, a sensor reconstruction, and a sensor design problem, allowing us to compute an analytical gradient with respect to selection variables. Intuitively, our design problem measures the performance of the reconstruction problem under changes to selection variables. Our design methodology is therefore generally applicable, and has applications beyond sensing.

III. Overview of Sensor Design

As we illustrate in Fig. 1 on a proprioceptive bar example, we start our processing with a soft robot design and a set of simulated interactions (Input; Sec. IV). We then use the strain field from simulated interactions to initialize a large set of manufacturable sensors that are sensitive to input interactions (Initialization; Sec. VIII). To sense *unknown* interactions, we seek the interaction forces that correspond to a deformation that minimizes the difference between measured and simulated sensor readings (Sensing; Sec. V). To identify the best set of sensors from the initial set, we then incrementally select sensors that maximize the reconstruction performance of our sensing (Design; Sec. VI). After optimization, we fabricate our proprioceptive soft robots (Output; Sec. IX). We discuss our sensor model in Sec. VII and demonstrate our method in Sec. X.

Our computational approach to sensor sub-selection nests three optimization problems (Fig. 2). An outer *design* problem

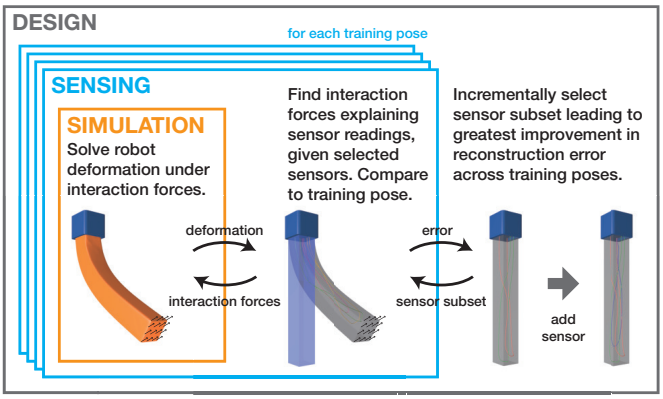

Fig. 2. Nesting of Optimization Problems. Optimality of each simulation subproblem is formulated as a constraint for its corresponding sensing subproblem, and optimality of the sensing subproblems for all input interactions are expressed as constraints for the design problem.

aims to choose sensors that minimize the error when reconstructing the robot's deformation. In turn, to reconstruct deformations, a middle *sensing* problem aims to compute interaction forces that lead to deformations that minimize the error with respect to sensor signals. Finally, an inner *simulation* problem computes deformations resulting from interaction forces.

To formalize this nesting, we make use of first-order optimality constraints: For each expected interaction, the inner *simulation* subproblem consists of computing the robot deformation that minimizes energy subject to interaction forces. Then, its corresponding *sensing* subproblem consists of computing the interaction forces that minimize sensing error, with optimality of the simulation subproblem expressed as a constraint. Finally, the outer *design* problem consists of the computation of binary sensor selection variables that minimize the overall reconstruction error for the set of input interactions, with optimality of all sensing subproblems expressed as constraints. In practice, we use the gradient of the design objective as a heuristic to incrementally select sensors.

We devote the next three sections to describe each optimization problem in detail, as well as the mathematical nesting of the three problems.

IV. Simulating Robot Deformations

The behavior of soft robots made of a single or several soft silicones or urethanes is well-captured by standard hyperelastic material models[54].

Discretizing the volume into finite elements e, we integrate the material-dependent strain energy density Ψ over the elemental rest volumes Ω_e, defining our elastic potential

$$E(\mathbf{x}) = \sum_e \int_{\Omega_e} \Psi(\mathbf{F}(\mathbf{X}, \mathbf{x}, \bar{\mathbf{X}}))\, \mathrm{d}\bar{\mathbf{X}}, \tag{1}$$

where the deformation gradient $\mathbf{F}$ is a function of the position $\bar{\mathbf{X}} \in \mathbb{R}^3$ within undeformed elements, and the undeformed $\mathbf{X} \in \mathbb{R}^{3n}$ and deformed $\mathbf{x} \in \mathbb{R}^{3n}$ configurations, restricted to the nodes influencing element e (see Appendix for more details).

Holding a sufficient set of degrees of freedom fixed, we simulate our soft robots by minimizing the total potential energy

$$f_{\text{sim}}(\mathbf{x}) = E(\mathbf{x}) - \mathbf{f}^T(\mathbf{x} - \mathbf{X}) - \sum_i p_i \left(V_i(\mathbf{x}) - V_i(\mathbf{X})\right), \tag{2}$$

obtained by subtracting the work performed by interaction forces $\mathbf{f}$ and the pressurized chambers with volumes V_i and corresponding gauge pressure p_i.

At the static equilibrium, elastic and external forces are in balance, and the gradient

$$\mathbf{g}_{\text{sim}}(\mathbf{x}) = \frac{\partial E(\mathbf{x})}{\partial \mathbf{x}} - \mathbf{f}^T - \sum_i p_i \frac{\partial V_i(\mathbf{x})}{\partial \mathbf{x}} \tag{3}$$

is zero. Minimizing with a standard Newton, we use the Hessian or tangent stiffness

$$\mathbf{H}_{\text{sim}}(\mathbf{x}) = \frac{\partial^2 E(\mathbf{x})}{\partial \mathbf{x}^2} - \sum_i p_i \frac{\partial^2 V_i(\mathbf{x})}{\partial \mathbf{x}^2} \tag{4}$$

to compute search directions. Note that fibre-reinforced robots could also be supported, by extending the simulation objective or adding inextensibility constraints (see, e.g.,[55]).

V. Estimating States from Sensor Readings

Our ultimate goal is the automated placement and sizing of a network of sensors to reliably and accurately reconstruct the equilibrium state of a solid. However, before we discuss how we automate the *design* of sensor networks, we introduce how we *reconstruct* the state from sensor readings.

If we embed stretch receptive sensors in our soft robots, then actuate and interact with them, the sensors deform and their signals change. Integrating the resistivity along each sensor channel, we can *compute* sensor signals s_j for a simulated deformation $\mathbf{x}$ as we describe in more detail in Sec. VII. We aim at reconstructing the deformed configuration $\bar{\mathbf{x}}$ and interaction forces $\bar{\mathbf{f}}$ from the corresponding sensor readings $\bar{s}_j$. We formulate a sensing objective as the difference between simulated s_j and measured readings $\bar{s}_j$

$$f_{\text{sense}}(\mathbf{f}) = \frac{1}{2} \sum_j \left(s_j(\mathbf{x}(\mathbf{f})) - \bar{s}_j\right)^2 + w_{\text{sense}} \frac{1}{2} \|\mathbf{f}\|^2. \tag{5}$$

To bound the forces, we add the L2-regularizer to our sensing objective, weighted with a small scale factor w_{sense}.

To reconstruct $(\bar{\mathbf{x}}, \bar{\mathbf{f}})$, we therefore ask the sensing objective to be minimized, with a first-order optimality constraint on our simulation objective:

$$\min_{\mathbf{f}} f_{\text{sense}}(\mathbf{f}) \quad \text{s.t.} \quad \mathbf{g}_{\text{sim}}(\mathbf{f}, \mathbf{x}(\mathbf{f})) = 0. \tag{6}$$

The first-order optimality constraint is important because we are looking for interaction forces $\mathbf{f}$ that lead to an *equilibrium* $\mathbf{x}$ whose sensor readings can explain the measured readings in a least-squares sense.

To minimize the objective, we implicitly enforce the first-order optimality constraint, simulating to static equilibrium whenever we evaluate our sensing objective f_{sense} or its

gradient. We can then compute the analytical sensing gradient with the help of the implicit function theorem

$$\mathbf{g}_{\text{sense}}(\mathbf{f}) = \frac{\partial f_{\text{sense}}(\mathbf{x})}{\partial \mathbf{x}} \mathbf{H}_{\text{sim}}^{-1}(\mathbf{x}) + w_{\text{sense}}\, \mathbf{f}^T, \tag{7}$$

pointing the interested reader to the appendix for a complete derivation.

We use a first-order, momentum-based scheme[56,57] for minimization, referring the unfamiliar reader to Goh's excellent introductory text[58].

VI. Designing Proprioceptive Soft Robots

As input to our design optimization, the user specifies a 3D model together with a set of example deformation-force pairs $(\bar{\mathbf{x}}, \bar{\mathbf{f}})$. To create these pairs, the user applies a set of expected interaction forces using our simulator.

For a small number of sensors, the reconstruction problem (6) is ill-posed, i.e., many combinations of external forces produce deformations where simulated sensor readings match measured ones. While the accuracy of reconstruction can be improved by increasing the number of sensors to the hundreds, fabrication becomes infeasible. Instead, we wish to find a small, fabricable number of sensors that produces good reconstructions. Choosing such sensors is complex, due to the integrated nature of sensor readings; therefore, we resort to computational methods. We first use the user-specified deformation-force pairs to initialize a large set of fabricable sensors (see Sec. VIII), and then select a small number of sensors that produces the desired reconstruction accuracy.

We aim at selecting the *minimal* subset of sensors that leads to a good reconstruction performance for *all* example deformations. To this end, we reformulate our sensing objective (5), multiplying each sensor's error with a binary selection variable $w_j \in \{0, 1\}$

$$f_{\text{sense}}(\mathbf{f}) = \frac{1}{2} \sum_j w_j \left(s_j(\mathbf{x}(\mathbf{f})) - \bar{s}_j \right)^2 + w_{\text{sense}} \frac{1}{2} \|\mathbf{f}\|^2. \tag{8}$$

We formulate our design objective by summing the reconstruction error of each example deformation k

$$f_{\text{design}}(\mathbf{w}) = \frac{1}{2} \sum_k \left\| \mathbf{x}_k(\mathbf{f}_k(\mathbf{w})) - \bar{\mathbf{x}}_k \right\|^2. \tag{9}$$

Interestingly, if we interpret the binary variables as continuous weights, we can formulate a continuous design optimization

$$\min_{\mathbf{w}} f_{\text{design}}(\mathbf{w}) \quad \text{s.t.} \quad \mathbf{g}_{\text{sense}}(\mathbf{w}, \mathbf{x}_k(\mathbf{f}_k(\mathbf{w}))) = 0 \quad \forall k. \tag{10}$$

The first-order optimality constraint on the sensing gradient (7) ensures that the deformation-force pairs $(\mathbf{x}_k, \mathbf{f}_k)$ produce optimal sensor readings for a given a set of weights.

The design gradient can be expressed analytically as

$$\mathbf{g}_{\text{design}}(\mathbf{w}) = \sum_k \left(\mathbf{x}_k - \bar{\mathbf{x}}_k \right)^T \mathbf{H}_{\text{sim}}^{-1}(\mathbf{x}_k) \frac{\mathrm{d}\mathbf{f}_k(\mathbf{w})}{\mathrm{d}\mathbf{w}}, \tag{11}$$

where the Jacobian of interaction forces with respect to the weights is obtained by applying the implicit function theorem to the optimality constraint on the sensing gradient (7)

$$\frac{\mathrm{d}\mathbf{f}_k(\mathbf{w})}{\mathrm{d}\mathbf{w}} = -\mathbf{H}_{\text{sense}}^{-1}(\mathbf{w}, \mathbf{x}_k) \frac{\partial \mathbf{g}_{\text{sense}}(\mathbf{w}, \mathbf{x}_k)}{\partial \mathbf{w}}, \tag{12}$$

referring the reader to the appendix for further details on the derivation.

We first experimented with solving the design optimization (10) using continuous optimization methods. We added a sparsity regularizer[49] to our design objective (9), penalizing the inclusion of sensors in the selected set, while effectively approximating the solution to the discrete optimization problem. However, for sensors where the error between simulated s_j and target readings $\bar{s}_j$ is zero, the corresponding weight w_j can take on arbitrary values without changing our sensing objective. The lack of meaning of non-zero weights causes numerical problems during optimization. If the sensor error is zero, the sparsity regularizer pushes *all* weights uniformly to zero. However, we observe that zero or very small weights cause a sudden increase in our design and sensing objectives, suggesting that there is a discontinuity.

To circumvent this discontinuity, we decided to use a heuristic approach instead of solving problem (10), where we utilize the continuous design gradient (11) to guide an *incremental* selection of sensors: In each iteration, we set the weights of all selected sensors (from previous iterations) to 1, and weights of all other sensors to 0. We then solve the sensing problem (6) for all example deformations to first-order optimality. When completed, we evaluate the design gradient (11), and select the yet unselected sensor with largest (negative) gradient as we expect this sensor to reduce our design objective (9) the most.

Initialization of the selection requires special treatment as the Hessian of the sum of differences between simulated and measured readings is zero if no sensors are selected. We start by randomly choosing a subset of 10 sensors. We then solve 10 instances of the sensing problem (6), for each sensor independently with its weight set to 1 (and all other weights to zero). We initialize the selection with the sensor that performs best.

Intuitively, we can expect our design problem to have several solutions if only one sensor is selected. However, the more sensors we select, the more we constrain the deformation space, minimizing the differences between simulated and specified poses. While we observe the error to increase occasionally between subsequent steps at the very beginning of our selection process, the error monotonically decreases after a sufficiently large subset is selected.

VII. Modeling Stretch Receptive Sensors

To sensorize our soft robots, we use silicone tubes of radius r, filled with Eutectic Gallium-Indium (EGaIn)[34]. At rest, we can compute the resistance of a sensor with Pouillet's law $R_0 = \rho \frac{l}{A}$ where l is the rest length of the sensor, $A = \pi r^2$ its cross-sectional area, and ρ the tabulated resistivity of the alloy.

If the surrounding silicone or urethane deforms, the sensor changes its length to $l + \Delta l$ and its radius to $r + \Delta r$. Due to these changes, the resistance along the filled tube will change, resulting in a sensor reading

$$s = R - R_0 = \frac{\rho}{\pi} \left(\frac{l + \Delta l}{(r + \Delta r)^2} - \frac{l}{r^2} \right). \tag{13}$$

To estimate sensor readings from simulated deformations, we miss a constitutive law that relates the change of length to the change of the cross-sectional radius. A common assumption is that the axial and transversal strain are related by Poisson's ratio[33]. However, Poisson's ratio models a linearized volume preservation, leading to significant error in predictions of sensor readings.

Motivated by the observation that the volume of the alloy does not change, we enforce volume preservation directly[59], setting $V = (l + \Delta l)\pi(r + \Delta r)^2$ equal to the volume at rest $V_0 = l\pi r^2$. The resulting sensor model

$$s = R_0 \left(\left(1 + \frac{\Delta l}{l} \right)^2 - 1 \right) \tag{14}$$

predicts measured sensor readings accurately.

In practice, we compute both the rest and deformed length of each sensor by sampling points along the sensor, summing up the lengths of the resulting segments. To determine the undeformed and deformed coordinates of sample points in 3D, we interpolate the nodal degrees of freedom with Lagrange shape functions (see our appendix for details).

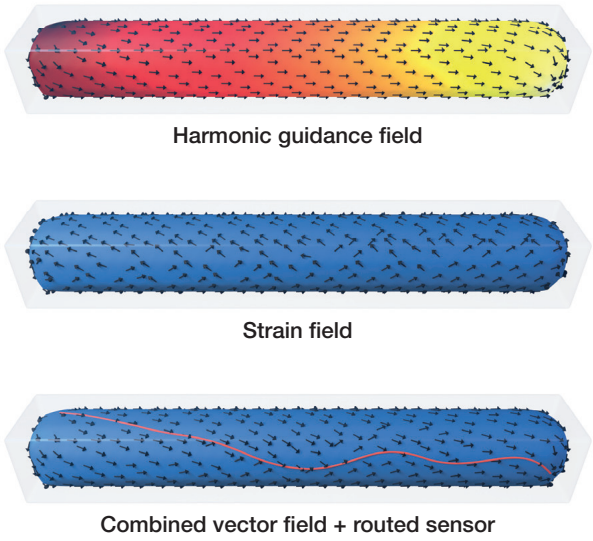

Fig. 3. To initialize a fabricable sensor on an offset surface in the robot, we first compute a harmonic guidance field by specifying start and end points (top). We also compute the strain field, given a target deformation (center). The sensor routing is then computed, taking both fields into account (bottom).

VIII. Initializing Fabricable Sensors

Before applying our sensor selection algorithm described in Sec. VI, we route a large set of candidate sensors through the robot. In all our examples, we found that 200 was a sufficient number of candidate sensors, which traversed the object in sufficiently diverse ways. In the routing of each sensor, we combine three criteria: (i) the routing must account for fabrication constraints; (ii) it must include some randomness; and (iii) to the extent possible, each sensor should follow the strain field of one of the user-specified deformation-force examples, as this will maximize sensitivity of this sensor with respect to the given deformation. To facilitate fabrication as described in Sec. IX, we place sensors at an offset from the outer surface of the robot, and we group all inlets and outlets on a small patch of the surface. The curvature of sensors is bounded, and they are perpendicular to the outer surface at inlets and outlets, to allow for clean and noise-free connections.

We start by computing an offset surface using a distance field. Then, for each input deformation-force example, we sample the tangent strain on the offset surface, and compute the eigenvector corresponding to the largest eigenvalue to define a guidance field. Using this guidance field, we route a number of random sensors for each deformation-force example. For each sensor, we randomly sample an inlet, an outlet, and an intermediate point somewhere on the offset surface (see Fig. 3). By randomly sampling these points, we ensure that the design optimization considers several possible sensors per input deformation example; the rest of the sensor routing process is deterministic. We route each half path of a sensor separately, smoothly connecting them at the intermediate point. Intuitively, we expect peak strains to appear close to the surface of our robot, further motivating the routing of sensors on an offset surface. In the remainder, we discuss the routing between any two points.

Given two end points of a sensor path, we compute a harmonic field on the offset surface, solving a Poisson equation with Dirichlet boundary conditions 0 and 1 at the end points[60]. Following the gradient of the harmonic field, it is possible to trace a path between the two points (compare with Fig. 3). However, while we wish to connect the two points, we also want to align the path with the guidance field[61]. To do so, at every point along the path, we define an angle around the gradient of the harmonic field. If the guidance field falls within this angle, we extend the path along the guidance field. If it does not, we extend the path along the edge of the angle closest to the guidance field. Near the end points, we narrow the angle to ensure that they are reached by the path. Occasionally, the path falls into a local minimum. Then, we take that point as an end point, recompute the harmonic field between this point and the target point, and restart the path generation. Once a sensor's path is complete, we apply a smoothing step to bound the sensor's curvature.

IX. Fabrication

We fabricate our robots using a two-step injection molding process, enabling us to accurately embed the sensors inside the silicone. Our strain sensors are made using silicone tubing (inner diameter 0.3 mm, outer diameter 0.7 mm), similarly to Wall et al.[5]. However, unlike them we require the sensors to be routed through the soft body.

The fabrication process is illustrated in Fig. 4 with a schematic cross-section view of a gripper design (see Sec. X). In the first molding step grooves are added in the silicone where the sensors will be placed (a). We generate the grooves in the mold automatically, and the mold is then 3D-printed. This means that the mold design complexity is independent of the sensor routing. The fabricability of a given sensor design

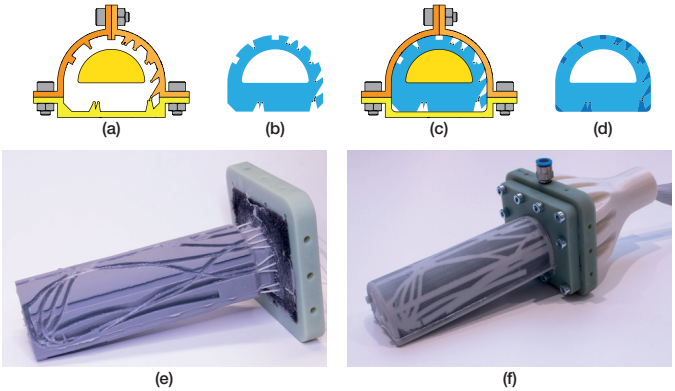

Fig. 4. Gripper fabrication. Top row shows fabrication sequence, using a two-step injection molding process. In the first molding step (a) we add grooves along which the sensors will be placed. Silicone tubes are then routed along the grooves (b), after which a second molding step (c) fills the grooves and embeds the sensors inside the silicone (d). Bottom shows Gripper before second molding step, with silicone tubes placed into the grooves (e), and the finished Gripper (f).

is guaranteed as the initialization process generates sensors which follow the surface of the robot.

The sample is then removed from the mold, and the silicone tube is routed along the grooves and fixed in place with SilPoxy (Smooth-On, Inc) (b). In a second molding step, this is placed inside a closed mold (c) and silicone is injected to fill the grooves. We note that crossing sensors is not a problem: as the sensors are thin, one sensor can be routed over another with negligible impact on performance.

Fig. 4 (e) shows the gripper design before the second molding step, with the silicone tubes routed along the grooves (i.e. Fig. 4 (b)), and Fig. 4 (f) shows the finished Gripper.

We used MoldMax 14 NV silicone (Smooth-On), as this has a relatively low viscosity that facilitates injection molding. Moreover, the silicone is tin-curing and is therefore less prone to cure inhibition. Molds were 3D-printed using an Objet Connex 350. Other than standard silicone molding equipment, the injection molding requires a pneumatic cartridge gun, which can be readily obtained at a low cost. Compared to gravity-assisted pouring, the injection molding greatly facilitates void-free molding of geometries such as long and thin features.

After completing the molding steps, we inject eGaIn into the silicone tubes and insert 30 AWG wires into the tube ends to make the electrical connection.

A. Electronics

To obtain the sensor resistance, we pass a constant current through the sensors and measure the voltage drop. After a pre-amplification stage, the voltage is recorded with an Arduino. We use 4-terminal sensing for improved performance, with two current-carrying and two voltage-sensing wires for each strain sensor.

B. Calibration

Although the expected sensor responses could be computed using the tabulated resistivity, silicone tube diameter, and sensor length, performance can be significantly improved with a calibration step. As discussed previously (Sec. VII), the sensor model requires a single parameter R_0 to relate resistance change to strain $\frac{\Delta l}{l}$ (compare with Eq. 14). We note that variations in the rest resistance R_0 are likely due to slight variations in the cross-sectional area of the silicone tube.

We also need to account for the resistance of the connecting wires (R_{con})—this has the effect of adding a constant offset to the resistance we measure. We performed a calibration experiment, where a bar with a single straight sensor running along its length was deformed into a known pose. By measuring the resistance in the rest pose and deformed pose, and comparing to the simulated results, we could solve for both R_0 and R_{con}.

It is reasonable to assume that R_{con} is consistent across different sensors, as the same sensor termination method is used. For a given sensor, we therefore estimate R_0 from the measured resistance in the undeformed configuration R_{meas} as $R_0 = R_{\text{meas}} - R_{\text{con}}$.

X. Results

We demonstrate our pipeline on two fabricated examples (Bar, Gripper) and one simulated example (Tentacle). For simulations of the MoldMax 14 NV silicone, we use a Neo-Hookean material with Poisson's ratio and Young's modulus set to 0.4 and 200 kPa, respectively. We estimated these values using a heuristic formula, documented in the "British Standard 903, Methods of vulcanised rubber" Part 19 (1950) and Part A7 (1957). Comparing simulated to physical deformations, we observe good prediction accuracy for our Shore 14 A durometer silicone. We rely on linear tetrahedral elements to keep the computational complexity within bounds, and use a custom FE implementation for solid simulation as we need to take derivatives of simulations for sensing and design.

A. Bending Bar

The predominant deformation mode in soft robotics is bending. To test how well we can reconstruct large-strain bending deformations, we augmented a 30 mm x 30 mm x 200 mm bar with proprioceptive capabilities (as introduced previously in Fig. 1): the bar is fixed at one end, and interactions are applied at the other end. We generated a set of 8 deformation-interaction pairs (Input, Training poses), and optimized for a set of sensors to match these input deformations (Output). With our sub-selection, we reduced the initial set of 200 (Initialization) to 5 sensors (Output, Optimized sensor set). All in- and outlets are routed through the fixed end. From only 5 integrated values, we can reconstruct the degrees of freedom (DoFs) of a 1'256 element mesh and corresponding interaction forces to a surprisingly good accuracy.

Fig. 5 left shows the reconstruction error for increasing numbers of sensors. Overall, adding more sensors results in a lower error. To evaluate the efficacy of our selection algorithm, we also performed 10 experiments where we selected sensors at random from the same set of initial sensors. The plot shows the mean reconstruction error across the random experiments, and also the maximum and minimum error (in m). We can see that our optimized result is a significant improvement over random, with a more consistent trend.

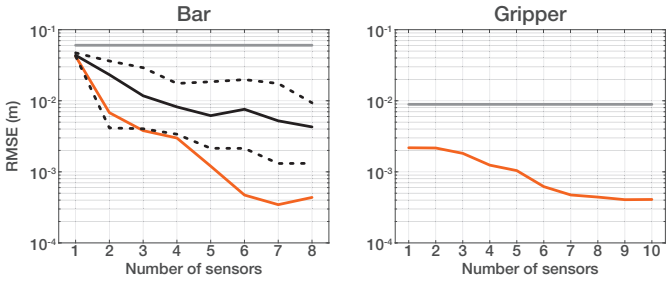

Fig. 5. Reconstruction error for increasing numbers of sensors for the Bar example (left) and Gripper example (right). Orange lines show error with the optimized sensors. Dark grey lines show the reconstruction error with no sensors. For the bar, we also include the mean, minimum, and maximum error from randomly selecting sensors (black lines).

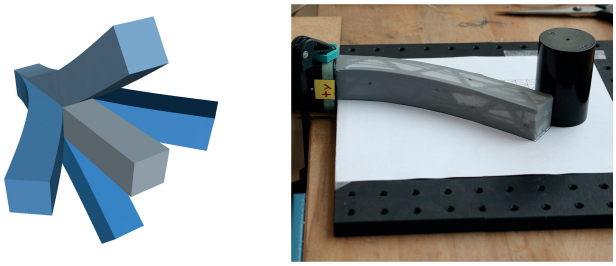

Fig. 6. Bending poses for bar validation experiment. Renders (left) and physical setup (right). The length of the bar is 200 mm.

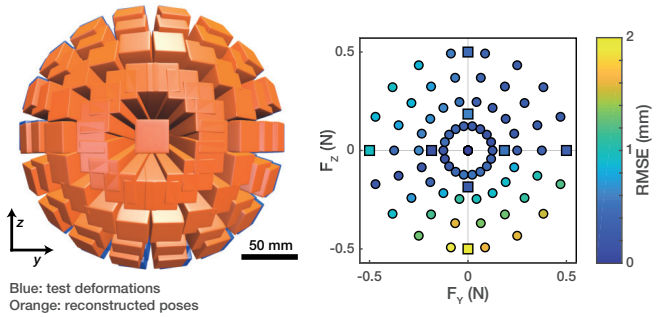

Fig. 7. Extrapolation performance. We apply a set of forces in the y- and z-directions (F_y, F_z) to generate a set of test deformations. Left, the set of reconstructed poses have been overlaid the test deformations. Right, we show the RMSE of the reconstructed deformation for each pose, with test deformations shown as filled circles. We also include the reconstruction performance for the 8 training deformations (filled squares). It can be seen that the performance varies smoothly and the method extrapolates well.

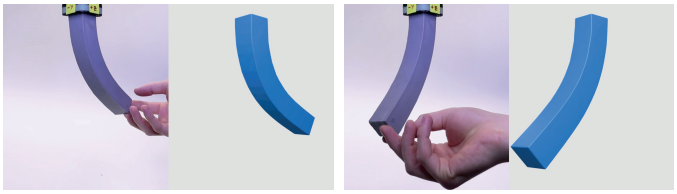

Fig. 8. Frames from bar interaction sequence. The full sequence is included in the supporting video.

We fabricated the bar with 5 sensors, as a trade-off between accuracy and fabrication complexity. If higher accuracy would be required, then more sensors could be added.

It is important to note that the computational cost of evaluating one set of random sensors is significant, as it requires a large set of reconstructions. Thus, a brute-force approach of randomly sampling sets of sensors would in general not perform well. Our evaluation of the gradient of the sensor weights enables a substantially more efficient approach to the problem.

As a validation experiment, we deform the physical bar into 4 known poses (bends in the 4 directions) as shown in Fig. 6. We perform reconstructions using the measured sensor values, and evaluate the error as the distance to the known deformed pose. Across 3 repeats of the 4 poses we obtain a mean reconstruction error of 3.05 mm (RMSE), standard deviation across trials 1.71 mm. Comparing this to the expected error in the ideal case (no sensor noise or fabrication tolerances) from Fig. 5 left, we can see that the error introduced in the fabrication pipeline is similar in magnitude to the numerical error introduced by the optimization.

To evaluate the extrapolation performance of our method, we generate a set of 71 test deformations by applying bending forces to the tip of the bar in a set of different directions. Fig. 7 visualizes the reconstruction error for the set of test poses, and also shows the reconstruction error for the 8 training poses. It can be seen that the reconstruction performance varies smoothly in the space of sampled forces, and that reconstruction performance for the training and test deformations is similar. This illustrates that our method extrapolates well.

We also perform an interaction experiment where we interact with the bar and reconstruct the deformation sequence. The full sequence is shown in the supporting video, and selected frames are shown in Fig. 8. It can be seen that the reconstruction performs well, and that the sensor response is fast. We note that for this example the reconstructions are done offline; a real-time demo would be possible but would significantly lower precision.

B. Pneumatic Gripper

Our single-material pneumatic gripper (see Fig. 9) has an air chamber of complex, asymmetric shape, causing it to bend in a particular direction when inflated. Among others, proprioceptive grippers have applications in grasping, not only detecting that an object has been picked up but also inferring the shape of the grasped object.

As input, we considered the gripper inflated to a fixed pressure, in-contact with a cylinder placed at 4 different locations along its length (Fig. 9 top left). Due to the shape complexity of the internal chamber, a total of 3'766 tetrahedral elements were necessary to avoid low-quality elements. The dimensions of the gripper are 40 mm x 32 mm x 120 mm.

We generated an initial set of 200 fabricable sensors, then ran our design optimization. Fig. 5 right shows the decrease in reconstruction error as the number of sensors increases. A strong monotonic decrease can be seen initially, which levels off between 9 and 10 sensors. As a trade-off between accuracy and complexity, we fabricated the gripper with 6 sensors. The initial set of sensors, and the optimized set can be seen in Fig. 9 bottom left. The physical gripper can also be seen in Fig. 4.

To test the gripper, we performed two inflation sequences: one free inflation and one where the motion is blocked by a cylinder. We recorded the pressure and the sensor measurements, and performed the reconstructions. The contact information with the cylinder was not provided to our reconstruction algorithm. As outlined in Sec. V, we estimate contact forces of unknown magnitude and direction on the complete surface of the gripper. Fig. 9 right shows the physical and reconstructed frames, and we can see that there is good agreement between the physical deformations and the reconstructed results. Note that we can infer the proprioceptive state of the gripper from sensor readings and the measured chamber pressure only. Even though the cylinder is *not* present in simulations, we can reconstruct the deformation under contact forces that differ from the input interactions. We refer to the accompanying video for additional views of the initial and fabricated sensors, and for the gripper inflation sequences.

C. Simulated Tentacle

To illustrate applications in bioinspired robotics, we optimized a thin and long tentacle design of dimensions 29 mm x 23 mm x 298 mm. Our sensor design algorithm received as input an initial set of 200 sensors and only 4 example poses (Fig. 10 top), and the optimization resulted in a 14-sensor design. The optimized result succeeds to reconstruct complex contact scenarios with high accuracy, with a mean reconstruction error of 7.05 mm (RMSE) for the input poses. The deformation model of the tentacle was discretized with a total of 778 elements.

Note that the contacts with the cylinders were explicitly modeled only for the generation of the input examples. During evaluation, the cylinders were *not* explicitly present, and the reconstruction algorithm estimated contact forces on the complete surface of the tentacle. For this thin and long robot design, we can expect the error to accumulate along the length of the tentacle. Nevertheless, the error at the tentacle tip is notably low (see Fig. 10 bottom): we show comparisons of reconstructed deformations to the simulated ground truth, visualized using a heat map. We note that all input and evaluation interactions are spatial. They do not lie in a planar subspace. We refer to the video for the full sequences and for a 360 degree view.

D. Optimization Timings

In Tab. I, we report timings for an average minimization iteration for our simulation and sensing objectives, and the average time it takes to evaluate the analytical gradient of our design objective to decide which sensor to add next. Note that reported timings for a design iteration include the time taken to minimize the sensing and simulation objectives to first-order optimality.

TABLE I
For each model, we report the time complexity of an average iteration when minimizing f_{sim} (avg. iter. sim), f_{sense} (avg. iter. sense), and f_{design} (avg. iter. design).

Model	avg. iter. sim	avg. iter. sense	avg. iter. design
Bar	15 ms	250 ms	2 h 40 min
Gripper	100 ms	1200 ms	12 h 20 min
Tentacle	15 ms	800 ms	4 h 13 min

XI. Conclusion

We have devised a method that aids the roboticist with the automated sizing of a stretch receptive sensor network, capable of reconstructing the proprioceptive state of soft robots. By introducing a set of selection variables in our sensing objective, we couple the design and sensing problems with a first-order optimality constraint. Intuitively, our design problem measures the performance of our sensing problem under changes to selection variables. Initializing a large set of fabricable sensors, we then use the analytical gradient of our design problem to select a small subset of sensors that results in a good reconstruction performance.

When casting our automated design as a sub-selection problem, we do not make any assumptions on the sensor paradigm. As long as a sensor leads to a reading $\bar{s}$, and this reading can be simulated by evaluating a sufficiently smooth function $s(\mathbf{x}(\mathbf{f}))$ at deformations $(\mathbf{x}, \mathbf{f})$, our sub-selection heuristic could be used to solve the underlying sensor design problem. There is often a duality between sensing and actuation (e.g., when using electro-active polymers). Hence, our formulation is applicable to actuation design, i.e., placing and sizing a discrete set of actuators. Moreover, our method interfaces with general elastic potentials, making it applicable to shells and rods, besides solids. Exciting future applications include the automated sensorization of cloth or wearables.

There are several remaining challenges. While our method interfaces with an arbitrary interaction behavior, it is difficult to model real-world interaction forces accurately[62]. Refining our modeling and design to interface with an *unknown* interaction behavior is an exciting avenue of future work. Similarly to Schumacher et al.[63], uncertainties in interactions could be parameterized, and the worst-case reconstruction error used when calculating gradients for sensor selection.

As our design problem is nonlinear and nonconvex, we cannot give guarantees to find a global optimum. However, as we demonstrate with our three examples, choosing from an initial set of 200 sensors is sufficient to generate robot designs whose proprioceptive state can be reconstructed to high accuracy from only a very few integrated sensor readings.

To compute analytical gradients with respect to selection variables, we cast our design problem as a continuous optimization problem. However, the inherently discrete nature of our design problem prevented us from using this continuous relaxation of our problem directly. An interesting direction of future work is the formulation of a coupled problem where non-zero weights are given a meaning.

Appendix: FEM Simulation

For simulations of our soft robots, we rely on Lagrange shape functions $N_i : \mathbb{R}^3 \to \mathbb{R}$, defined at every node i of the tetrahedral element mesh. By construction, Lagrange shape functions evaluate to one at their corresponding node, and

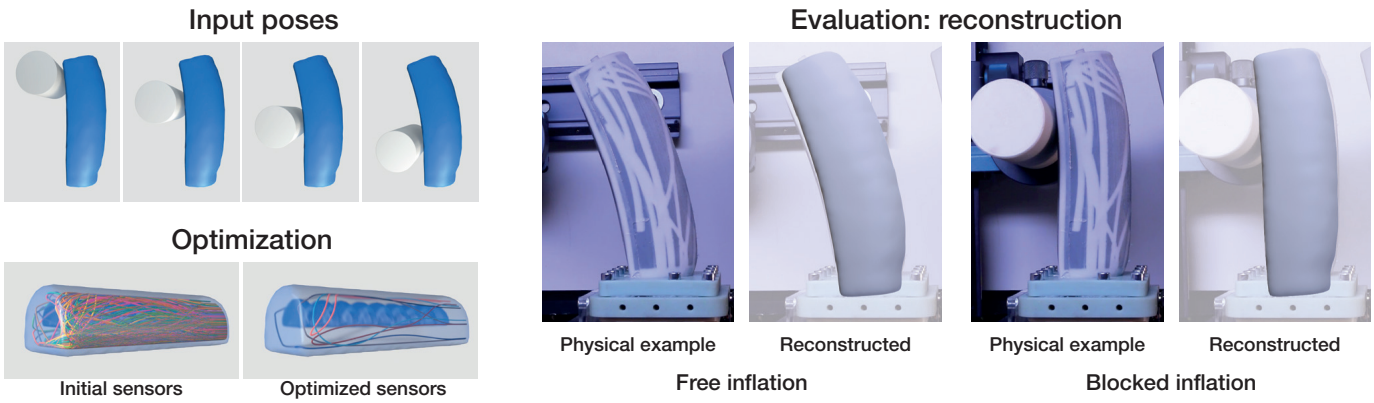

Fig. 9. Gripper example (length: 120 mm). We start with a set of interactions, where the gripper is actuated and comes into contact with a cylinder at different positions along its length (top left). We then generate an initial set of sensors, and run our optimization to arrive at a set of 6 sensors (bottom left). We fabricated the gripper, and conducted an experiment where we performed a free inflation of the gripper along with a blocked inflation, reconstructing the deformation using the measured sensor signals and pressure. The cylinder is *not* present in simulations.

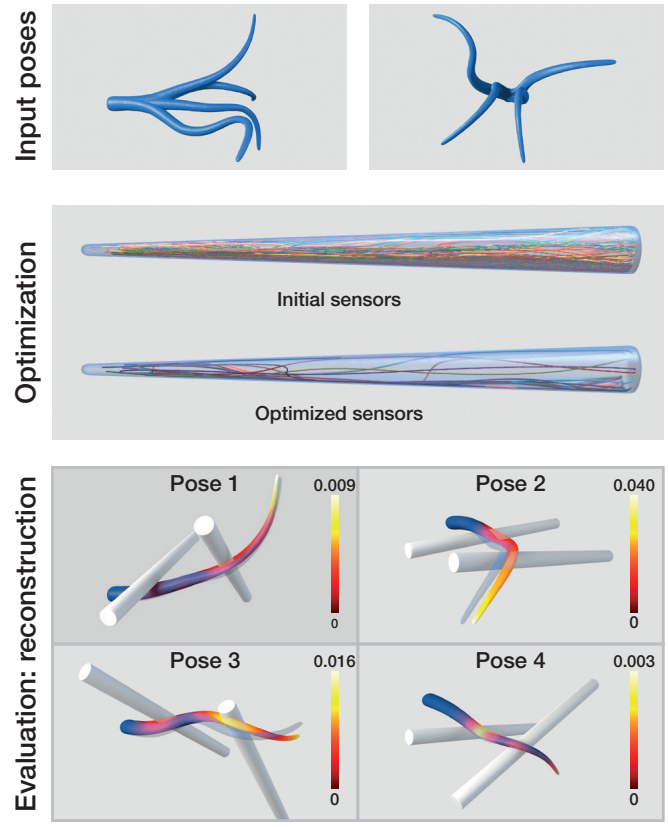

Fig. 10. Tentacle example (length: 298 mm). We start with a set of input interactions (top), from which we generate a set of 200 initial sensors. We run our optimization, and select a set of 14 sensors as this gives a good target matching performance. We then evaluate the reconstruction performance of the tentacle by simulating deformations into the 4 input poses, reconstructing the deformations from the simulated sensor readings (bottom). The ground truth deformations are shown in transparent blue, and the heat map shows the localized reconstruction error in m. We note that the cylinders are used for creating the target deformations, and are *not* included in reconstruction simulations.

zero at every other node, uniquely defining the polynomial coefficients of the shape functions $N_{e,i}$ for every element-node pair.

Denoting the undeformed and deformed nodes of element e with $\mathbf{X}_{e,i}$ and $\mathbf{x}_{e,i}$, respectively, we use the shape functions to interpolate the undeformed and deformed configurations within elements, and define the deformation gradient for an arbitrary point $\bar{\mathbf{X}}$

$$\mathbf{F}(\bar{\mathbf{X}}) = \left(\sum_i \mathbf{x}_{e,i} \frac{\partial N_{e,i}(\bar{\mathbf{X}})}{\partial \bar{\mathbf{X}}}\right)\left(\sum_i \mathbf{X}_{e,i} \frac{\partial N_{e,i}(\bar{\mathbf{X}})}{\partial \bar{\mathbf{X}}}\right)^{-1}. \tag{15}$$

While we can substitute shape functions of arbitrary order, we rely on *linear* shape functions for computational efficiency. Because their derivatives do *not* depend on the position $\bar{\mathbf{X}}$, the deformation gradient is constant for each element

$$\mathbf{F}_e = \begin{bmatrix} \mathbf{x}_{e,1} - \mathbf{x}_{e,4} & \mathbf{x}_{e,2} - \mathbf{x}_{e,4} & \mathbf{x}_{e,3} - \mathbf{x}_{e,4} \end{bmatrix} \mathbf{B}_e \tag{16}$$
$$\mathbf{B}_e = \begin{bmatrix} \mathbf{X}_{e,1} - \mathbf{X}_{e,4} & \mathbf{X}_{e,2} - \mathbf{X}_{e,4} & \mathbf{X}_{e,3} - \mathbf{X}_{e,4} \end{bmatrix}^{-1}$$

with precomputed, constant matrix $\mathbf{B}_e$.

To model our soft robots, we rely on a hyperelastic material model. While our technique interfaces with arbitrary models, we observe good prediction accuracy with a Neo-Hookean material, parameterized with Lamé constants μ and λ, whose strain energy density

$$\Psi(\mathbf{F}) = \frac{\mu}{2}(I_1 - 3 - 2\ln J) + \frac{\lambda}{2}(\ln J)^2, \tag{17}$$

depends on the two invariants

$$I_1 = \mathrm{tr}(\mathbf{F}^T\mathbf{F}) \quad \text{and} \quad I_3 = J^2 = \det(\mathbf{F}^T\mathbf{F}). \tag{18}$$

Because the deformation gradient, and hence also the strain energy density, are constant for each element, the elastic potential reads

$$E(\mathbf{x}) = \sum_e V_e \Psi(\mathbf{F}_e) \tag{19}$$

where the deformation gradient for element e only depends on the four incident deformed nodes in the $3n$-vector $\mathbf{x}$. The rest volume of an element V_e is conveniently $\det(\mathbf{B}_e)^{-1}$. Note that E is a nonlinear function of the deformed configuration due to the use of the Green strain, and the nonlinear constitutive model. If higher order shape functions are used, numerical quadrature is necessary. However, while the computational complexity of our simulations, sensing, and design would increase with the order of the shape functions, our automated design is generic and independent of the hyperelastic model or the order of the shape functions used.

The uninflated and inflated volumes of pneumatic chambers can be computed from the undeformed and deformed nodes on the chamber surface with the help of Gauss' theorem (see, e.g., supplemental material for[64]).

Appendix: Gradients And Hessians

The sensing and design problems discussed in Secs. V and VI, respectively, are both formulated as constrained optimizations. To compute gradients of the objective functions, we solve the constraints and apply the implicit function theorem. Here, we provide details about the formulation of gradients and Hessians of the sensing and design problems.

The gradient of the sensing objective (8) can be expressed as

$$\mathbf{g}_{\text{sense}}(\mathbf{f}) = \frac{\partial f_{\text{sense}}}{\partial \mathbf{f}} + \frac{\partial f_{\text{sense}}}{\partial \mathbf{x}} \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}\mathbf{f}}, \tag{20}$$

where $\frac{\partial f_{\text{sense}}}{\partial \mathbf{f}} = w_{\text{sense}}\, \mathbf{f}^T$, and

$$\frac{\partial f_{\text{sense}}}{\partial \mathbf{x}} = (\mathbf{s} - \bar{\mathbf{s}})^T \text{diag}(\mathbf{w}) \frac{\partial \mathbf{s}}{\partial \mathbf{x}} = \mathbf{w}^T \text{diag}(\mathbf{s} - \bar{\mathbf{s}}) \frac{\partial \mathbf{s}}{\partial \mathbf{x}}. \tag{21}$$

By imposing first-order optimality on simulations, we can apply the implicit function theorem to obtain

$$\begin{aligned} \mathbf{g}_{\text{sim}}(\mathbf{x}) = 0 \quad \Rightarrow \quad & \frac{\partial \mathbf{g}_{\text{sim}}}{\partial \mathbf{f}} + \mathbf{H}_{\text{sim}} \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}\mathbf{f}} = 0 \\ & \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}\mathbf{f}} = -\mathbf{H}_{\text{sim}}^{-1} \frac{\partial \mathbf{g}_{\text{sim}}}{\partial \mathbf{f}} = \mathbf{H}_{\text{sim}}^{-1}. \end{aligned} \tag{22}$$

As a result, we obtain expression (7) for the sensing gradient.

The gradient of the design objective (9) can be expressed as

$$\mathbf{g}_{\text{design}}(\mathbf{w}) = \frac{\partial f_{\text{design}}}{\partial \mathbf{w}} + \sum_k \frac{\partial f_{\text{design}}}{\partial \mathbf{x}_k} \frac{\mathrm{d}\mathbf{x}_k}{\mathrm{d}\mathbf{f}_k} \frac{\mathrm{d}\mathbf{f}_k}{\mathrm{d}\mathbf{w}}, \tag{23}$$

where $\frac{\partial f_{\text{design}}}{\partial \mathbf{w}} = 0$ in our case, $\frac{\mathrm{d}\mathbf{x}_k}{\mathrm{d}\mathbf{f}_k}$ is obtained from the first-order optimality constraint of the simulation problem as shown in (22) above, and

$$\frac{\partial f_{\text{design}}}{\partial \mathbf{x}_k} = (\mathbf{x}_k - \bar{\mathbf{x}}_k)^T. \tag{24}$$

By imposing first-order optimality on each sensing problem, we can apply the implicit function theorem to obtain

$$\begin{aligned} \mathbf{g}_{\text{sense}}(\mathbf{f}_k) = 0 \quad \Rightarrow \quad & \frac{\partial \mathbf{g}_{\text{sense}}}{\partial \mathbf{w}} + \mathbf{H}_{\text{sense}} \frac{\mathrm{d}\mathbf{f}_k}{\mathrm{d}\mathbf{w}} = 0 \\ \frac{\mathrm{d}\mathbf{f}_k}{\mathrm{d}\mathbf{w}} = -\mathbf{H}_{\text{sense}}^{-1} \frac{\partial \mathbf{g}_{\text{sense}}}{\partial \mathbf{w}} = -\mathbf{H}_{\text{sense}}^{-1} \mathbf{H}_{\text{sim}}^{-1} & \frac{\partial \mathbf{s}_k}{\partial \mathbf{x}_k}^T \text{diag}(\mathbf{s}_k - \bar{\mathbf{s}}_k). \end{aligned} \tag{25}$$

As a result, we obtain expression (11) for the design gradient.

The Hessian of the sensing problem can be obtained as

$$\mathbf{H}_{\text{sense}} = \frac{\partial \mathbf{g}_{\text{sense}}}{\partial \mathbf{f}} + \frac{\partial \mathbf{g}_{\text{sense}}}{\partial \mathbf{x}} \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}\mathbf{f}} = w_{\text{sense}}\, \mathbf{I} + \frac{\partial \mathbf{g}_{\text{sense}}}{\partial \mathbf{x}} \mathbf{H}_{\text{sim}}^{-1}. \tag{26}$$

To avoid tensor notation, we assemble the columns of $\frac{\partial \mathbf{g}_{\text{sense}}}{\partial \mathbf{x}}$ separately:

$$\begin{aligned} \frac{\partial \mathbf{g}_{\text{sense}}}{\partial x_i} = \mathbf{H}_{\text{sim}}^{-1} \Bigg(& \frac{\partial}{\partial x_i} \left(\frac{\partial \mathbf{s}}{\partial \mathbf{x}} \right)^T \text{diag}(\mathbf{w})(\mathbf{s} - \bar{\mathbf{s}}) \\ & + \left(\frac{\partial \mathbf{s}}{\partial \mathbf{x}} \right)^T \text{diag}(\mathbf{w}) \frac{\partial \mathbf{s}}{\partial x_i} - \frac{\partial \mathbf{H}_{\text{sim}}}{\partial x_i} \mathbf{g}_{\text{sense}}^T \Bigg). \end{aligned} \tag{27}$$

Acknowledgment

We thank the anonymous reviewers for their helpful comments; Carolina Ferrari for model design; Héctor Barreiro and Igor Santesteban for rendering assistance. This work has been supported by the projects SOMA (European Commission, Horizon 2020 Framework Programme, H2020-ICT-645599) and TouchDesign (ERC Consolidator Grant no. 772738).

References

[1] Y.-L. Park, C. Majidi, R. Kramer, P. Bérard, and R. J. Wood, "Hyperelastic pressure sensing with a liquid-embedded elastomer," *Journal of Micromechanics and Microengineering*, vol. 20, no. 12, p. 125029, 2010.

[2] B. O'Brien, T. Gisby, and I. A. Anderson, "Stretch sensors for human body motion," in *Electroactive Polymer Actuators and Devices (EAPAD) 2014*, vol. 9056. International Society for Optics and Photonics, 2014, p. 905618.

[3] Y. Mengüç, Y.-L. Park, H. Pei, D. Vogt, P. M. Aubin, E. Winchell, L. Fluke, L. Stirling, R. J. Wood, and C. J. Walsh, "Wearable soft sensing suit for human gait measurement," *The International Journal of Robotics Research*, vol. 33, no. 14, pp. 1748–1764, 2014.

[4] U. Culha, S. G. Nurzaman, F. Clemens, and F. Iida, "Svas3: strain vector aided sensorization of soft structures," *Sensors*, vol. 14, no. 7, pp. 12 748–12 770, 2014.

[5] V. Wall, G. Zöller, and O. Brock, "A method for sensorizing soft actuators and its application to the rbo hand 2," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 4965–4970.

[6] M. C. Yuen, R. Kramer-Bottiglio, and J. Paik, "Strain sensor-embedded soft pneumatic actuators for extension and bending feedback," in *2018 IEEE International Conference on Soft Robotics (RoboSoft)*. IEEE, 2018, pp. 202–207.

[7] K. Suzumori, S. Iikura, and H. Tanaka, "Development of flexible microactuator and its applications to robotic mechanisms," in *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*. IEEE, 1991, pp. 1622–1627.

[8] F. Ilievski, A. D. Mazzeo, R. F. Shepherd, X. Chen, and G. M. Whitesides, "Soft robotics for chemists," *Angewandte Chemie*, vol. 123, no. 8, pp. 1930–1935, 2011.

[9] F. Connolly, C. J. Walsh, and K. Bertoldi, "Automatic design of fiber-reinforced soft actuators for trajectory matching," *Proceedings of the National Academy of Sciences*, vol. 114, no. 1, pp. 51–56, 2017.

[10] M. A. Robertson and J. Paik, "New soft robots really suck: Vacuum-powered systems empower diverse capabilities," *Science Robotics*, vol. 2, no. 9, p. eaan6357, 2017.

[11] A. Miriyev, K. Stack, and H. Lipson, "Soft material for soft actuators," *Nature Communciations*, 2017.

[12] T. Umedachi, V. Vikas, and B. A. Trimmer, "Highly deformable 3-d printed soft robot generating inching and crawling locomotions with variable friction legs," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013, pp. 4590–4595.

[13] B. N. Peele, T. J. Wallin, H. Zhao, and R. F. Shepherd, "3d printing antagonistic systems of artificial muscle using projection stereolithography," *Bioinspiration & biomimetics*, vol. 10, no. 5, p. 055003, 2015.

[14] J. Morrow, S. Hemleben, and Y. Menguc, "Directly fabricating soft robotic actuators with an open-source 3-d printer," *IEEE Robotics and Automation Letters*, vol. 2, no. 1, pp. 277–281, 2017.

[15] D. Drotman, M. Ishida, S. Jadhav, and M. T. Tolley, "Application-driven design of soft, 3d printed, pneumatic actuators with bellows," *IEEE/ASME Transactions on Mechatronics*, 2018.

[16] A. D. Marchese, C. D. Onal, and D. Rus, "Autonomous soft robotic fish capable of escape maneuvers using fluidic elastomer actuators," *Soft Robotics*, vol. 1, no. 1, pp. 75–87, 2014.

[17] M. A. McEvoy and N. Correll, "Materials that couple sensing, actuation, computation, and communication," *Science*, vol. 347, no. 6228, p. 1261689, 2015.

[18] M. Wehner, R. L. Truby, D. J. Fitzgerald, B. Mosadegh, G. M. Whitesides, J. A. Lewis, and R. J. Wood, "An integrated design and fabrication strategy for entirely soft, autonomous robots," *Nature*, 2016.

[19] J. Shintake, V. Cacucciolo, D. Floreano, and H. Shea, "Soft robotic grippers," *Advanced Materials*, p. 1707035, 2018.

[20] K. C. Galloway, K. P. Becker, B. Phillips, J. Kirby, S. Licht, D. Tchernov, R. J. Wood, and D. F. Gruber, "Soft robotic grippers for biological sampling on deep reefs," *Soft robotics*, vol. 3, no. 1, pp. 23–33, 2016.

[21] E. W. Hawkes, L. H. Blumenschein, J. D. Greer, and A. M. Okamura, "A soft robot that navigates its environment through growth," *Science Robotics*, vol. 2, no. 8, p. eaan3028, 2017.

[22] M. A. Estrada, S. Mintchev, D. L. Christensen, M. R. Cutkosky, and D. Floreano, "Forceful manipulation with micro air vehicles," *Science Robotics*, vol. 3, no. 23, p. eaau6903, 2018.

[23] M. Cianchetti, T. Ranzani, G. Gerboni, T. Nanayakkara, K. Althoefer, P. Dasgupta, and A. Menciassi, "Soft robotics technologies to address shortcomings in today's minimally invasive surgery: the stiff-flop approach," *Soft robotics*, vol. 1, no. 2, pp. 122–131, 2014.

[24] R. Deimel and O. Brock, "A novel type of compliant and underactuated robotic hand for dexterous grasping," *The International Journal of Robotics Research*, vol. 35, no. 1-3, pp. 161–185, 2016.

[25] P. Polygerinos, Z. Wang, K. C. Galloway, R. J. Wood, and C. J. Walsh, "Soft robotic glove for combined assistance and at-home rehabilitation," *Robotics and Autonomous Systems*, vol. 73, pp. 135–143, 2015.

[26] R. S. Diteesawat, T. Helps, M. Taghavi, and J. Rossiter, "High strength bubble artificial muscles for walking assistance," in *2018 IEEE International Conference on Soft Robotics (RoboSoft)*. IEEE, 2018, pp. 388–393.

[27] R. Mutlu, G. Alici, M. in het Panhuis, and G. M. Spinks, "3d printed flexure hinges for soft monolithic prosthetic fingers," *Soft Robotics*, vol. 3, no. 3, pp. 120–133, 2016.

[28] H. Hauser, A. J. Ijspeert, R. M. Füchslin, R. Pfeifer, and W. Maass, "Towards a theoretical foundation for morphological computation with compliant bodies," *Biological cybernetics*, vol. 105, no. 5-6, pp. 355–370, 2011.

[29] A. Michael-Titus, P. Revest, and P. Shortland, *The Nervous System*, 2nd ed. Churchill Livingstone, 2010.

[30] H. Wang, M. Totaro, and L. Beccai, "Toward perceptive soft robots: Progress and challenges," *Advanced Science*, p. 1800541, 2018.

[31] P. Polygerinos, N. Correll, S. A. Morin, B. Mosadegh, C. D. Onal, K. Petersen, M. Cianchetti, M. T. Tolley, and R. F. Shepherd, "Soft robotics: Review of fluid-driven intrinsically soft devices; manufacturing, sensing, control, and applications in human-robot interaction," *Advanced Engineering Materials*, vol. 19, no. 12, p. 1700016, 2017.

[32] M. Bächer, B. Hepp, F. Pece, P. G. Kry, B. Bickel, B. Thomaszewski, and O. Hilliges, "Defsense: Computational design of customized deformable input devices," in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, ser. CHI '16. New York, NY, USA: ACM, 2016, pp. 3806–3816.

[33] Y. L. Park, B. R. Chen, and R. J. Wood, "Design and fabrication of soft artificial skin using embedded microchannels and liquid conductors," *IEEE Sensors Journal*, vol. 12, no. 8, pp. 2711–2718, Aug 2012.

[34] N. Farrow and N. Correll, "A soft pneumatic actuator that can sense grasp and touch," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 2317–2323.

[35] J. C. Case, E. L. White, and R. K. Kramer, "Sensor enabled closed-loop bending control of soft beams," *Smart Materials and Structures*, vol. 25, no. 4, p. 045018, 2016.

[36] E. L. White, J. C. Case, and R. K. Kramer, "Multi-mode strain and curvature sensors for soft robotic applications," *Sensors and Actuators A: Physical*, vol. 253, pp. 188–197, 2017.

[37] J. T. Muth, D. M. Vogt, R. L. Truby, Y. Mengüç, D. B. Kolesky, R. J. Wood, and J. A. Lewis, "Embedded 3d printing of strain sensors within highly stretchable elastomers," *Advanced Materials*, vol. 26, no. 36, pp. 6307–6312, 2014.

[38] R. L. Truby, M. Wehner, A. K. Grosskopf, D. M. Vogt, S. G. Uzel, R. J. Wood, and J. A. Lewis, "Soft somatosensitive actuators via embedded 3d printing," *Advanced Materials*, vol. 30, no. 15, p. 1706383, 2018.

[39] J.-B. Chossat, Y.-L. Park, R. J. Wood, and V. Duchaine, "A soft strain sensor based on ionic and metal liquids," *IEEE Sensors Journal*, vol. 13, no. 9, pp. 3405–3414, 2013.

[40] C. Larson, B. Peele, S. Li, S. Robinson, M. Totaro, L. Beccai, B. Mazzolai, and R. Shepherd, "Highly stretchable electroluminescent skin for optical signaling and tactile sensing," *Science*, vol. 351, no. 6277, pp. 1071–1074, 2016.

[41] W. Felt, M. Suen, and C. D. Remy, "Sensing the motion of bellows through changes in mutual inductance," in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. IEEE, 2016, pp. 5252–5257.

[42] Y. Tenzer, L. P. Jentoft, and R. D. Howe, "The feel of mems barometers: Inexpensive and easily customized tactile array sensors," *IEEE Robotics & Automation Magazine*, vol. 3, no. 21, pp. 89–95, 2014.

[43] Y. Zhang, G. Laput, and C. Harrison, "Electrick: Low-cost touch sensing using electric field tomography," in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, 2017, pp. 1–14.

[44] V. Wall, G. Zöller, and O. Brock, "Acoustic sensing for soft pneumatic actuators," in *Intelligent Robots and Systems (IROS), 2018 IEEE/RSJ International Conference on*. IEEE, 2018.

[45] C. Chorley, C. Melhuish, T. Pipe, and J. Rossiter, "Development of a tactile sensor based on biologically inspired edge encoding," in *Advanced Robotics, 2009. ICAR 2009. International Conference on*. IEEE, 2009, pp. 1–6.

[46] Z. Zhang, J. Dequidt, and C. Duriez, "Vision-based sensing of external forces acting on soft robots using finite element method," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1529–1536, 2018.

[47] T. Helps and J. Rossiter, "Proprioceptive flexible fluidic actuators using conductive working fluids," *Soft robotics*, vol. 5, no. 2, pp. 175–189, 2018.

[48] S. Coros, B. Thomaszewski, G. Noris, S. Sueda, M. Forberg, R. W. Sumner, W. Matusik, and B. Bickel, "Computational design of mechanical characters," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 83:1–83:12, Jul. 2013.

[49] M. Skouras, B. Thomaszewski, S. Coros, B. Bickel, and M. Gross, "Computational design of actuated deformable characters," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 82:1–82:10, Jul. 2013.

[50] M. Skouras, B. Thomaszewski, P. Kaufmann, A. Garg, B. Bickel, E. Grinspun, and M. Gross, "Designing inflatable structures," *ACM Trans. Graph.*, vol. 33, no. 4, pp. 63:1–63:10, Jul. 2014.

[51] V. Megaro, J. Zehnder, M. Bächer, S. Coros, M. Gross, and B. Thomaszewski, "A computational design tool for compliant mechanisms," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 82:1–82:12, Jul. 2017.

[52] V. Megaro, E. Knoop, A. Spielberg, D. I. W. Levin, W. Matusik, M. Gross, B. Thomaszewski, and M. Bächer, "Designing cable-driven actuation networks for kinematic chains and trees," in *Proceedings of the ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, ser. SCA '17. New York, NY, USA: ACM, 2017, pp. 15:1–15:10.

[53] J. Pérez, B. Thomaszewski, S. Coros, B. Bickel, J. A. Canabal, R. Sumner, and M. A. Otaduy, "Design and fabrication of flexible rod meshes," *ACM Trans. Graph.*, vol. 34, no. 4, pp. 138:1–138:12, Jul. 2015.

[54] E. Sifakis and J. Barbic, "Fem simulation of 3d deformable solids: A practitioner's guide to theory, discretization and model reduction," in *ACM SIGGRAPH 2012 Courses*, ser. SIGGRAPH '12. New York, NY, USA: ACM, 2012, pp. 20:1–20:50.

[55] M. Pozzi, E. Miguel, R. Deimel, M. Malvezzi, B. Bickel, O. Brock, and D. Prattichizzo, "Efficient fem-based simulation of soft robots modeled as kinematic chains," in *IEEE International Conference on Robotics and Automation 2018*, 2018.

[56] Y. Nesterov, "A method of solving a convex programming problem with convergence rate o (1/k2)," *Soviet Mathematics Doklady*, vol. 27, no. 2, 1983.

[57] B. O'Donoghue and E. Candès, "Adaptive restart for accelerated gradient schemes," *Foundations of Computational Mathematics*, vol. 15, no. 3, pp. 715–732, Jun 2015.

[58] G. Goh, "Why momentum really works," *Distill*, 2017. [Online]. Available: http://distill.pub/2017/momentum

[59] S. Kim, J. Oh, D. Jeong, W. Park, and J. Bae, "Consistent and reproducible direct ink writing of eutectic gallium–indium for high-quality soft sensors," *Soft Robotics*, vol. 5, no. 5, pp. 601–612, 2018, pMID: 29975584.

[60] T. Pereira, S. Rusinkiewicz, and W. Matusik, "Computational light routing: 3d printed optical fibers for sensing and display," *ACM Trans. Graph.*, vol. 33, no. 3, pp. 24:1–24:13, Jun. 2014.

[61] P. Pérez, M. Gangnet, and A. Blake, "Poisson image editing," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 313–318, Jul. 2003.

[62] T.-H. Pham, A. Kheddar, A. Qammaz, and A. A. Argyros, "Towards force sensing from vision: Observing hand-object interactions to infer manipulation forces," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 2810–2819.
[63] C. Schumacher, J. Zehnder, and M. Bächer, "Set-In-Stone: Worst-case optimization of structures weak in tension," *ACM Trans. Graph.*, vol. 37, no. 6, pp. 252:1–252:13, Nov. 2018.
[64] M. Bächer, E. Whiting, B. Bickel, and O. Sorkine-Hornung, "Spin-it: Optimizing moment of inertia for spinnable objects," *ACM Trans. Graph.*, vol. 33, no. 4, pp. 96:1–96:10, Jul. 2014.