Distributed Proprioception of 3D Configuration in Soft, Sensorized Robots via Deep Learning

Ryan L. Truby, Cosimo Della Santina, and Daniela Rus

***Abstract*—Creating soft robots with sophisticated, autonomous capabilities requires these systems to possess reliable, on-line proprioception of 3D configuration through integrated soft sensors. We present a framework for predicting a soft robot's 3D configuration via deep learning using feedback from a soft, proprioceptive sensor skin. Our framework introduces a kirigami-enabled strategy for rapidly sensorizing soft robots using off-the-shelf materials, a general kinematic description for soft robot geometry, and an investigation of neural network designs for predicting soft robot configuration. Even with hysteretic, non-monotonic feedback from the piezoresistive sensors, recurrent neural networks show potential for predicting our new kinematic parameters and, thus, the robot's configuration. One trained neural network closely predicts steady-state configuration during operation, though complete dynamic behavior is not fully captured. We validate our methods on a trunk-like arm with 12 discrete actuators and 12 proprioceptive sensors. As an essential advance in soft robotic perception, we anticipate our framework will open new avenues towards closed loop control in soft robotics.**

***Index Terms*—Modeling, control, and learning for soft robots, soft sensors and actuators, deep learning in robotics and automation.**

I. Introduction

SOFT robotics represents an auspicious new paradigm for designing robots with improved adaptability, resilience, safety, and more by introducing compliance and deformability in robot bodies [1]. However, it is this collection of enabling material properties that complicates approaches to soft robotic control. Despite more than a decade of progress, implementing autonomous behaviors in soft robots remains a long-standing challenge for this interdisciplinary field.

Manuscript received September 10, 2019; accepted January 28, 2020. Date of publication February 26, 2020; date of current version March 9, 2020. This letter was recommended for publication by Associate Editor M. Cianchetti and Kyu-Jin Cho upon evaluation of the reviewers' comments. This work was supported by the NSF EFRI Program Grant 1830901. The work of R. L. Truby was supported by the Schmidt Science Fellows program, in partnership with the Rhodes Trust. *(Ryan L. Truby and Cosimo Della Santina contributed equally to this work.) (Corresponding author: Ryan L. Truby.)*

The authors are with the MIT Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: rltruby@mit.edu; cosimodellasantina@gmail.com; rus@csail.mit.edu).

This letter has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors. The material contains a video overview of all contributions. Data sets and code used for this project are available at github.com/SensoSoRo. The authors thank Dr. Robert Katzschmann for original designs of the 3D soft robotic arm and Ms. Olivia Siegel for assistance in sensor fabrication.

Digital Object Identifier 10.1109/LRA.2020.2976320

Two factors make the control of soft robots difficult. First, the continuum mechanics of these systems makes the implementation of model-based planning and control difficult (though some progress has recently been made on this front [2]). Second, embedded sensing through soft materials is necessary for the next generation of soft robots. However, the design and fabrication of soft sensorized robots is non-trivial and can require specialized methods, materials, and equipment [3]–[8]. Consequently, advances in complex, soft robotic feedback control have typically been achieved through exteroception and exogenous sensing methods, including motion capture vision systems [9], [10] and magnetic tracking [11]. While these approaches are appropriate in the lab for proofs-of-concept, soft robots require integrated soft sensors for more practical, sophisticated, and autonomous capabilities.

Thus, the challenge of soft robotic perception and control is both a materials and a robotics one. Regarding control, learning-based strategies represent an alternative approach that bypasses the challenges of model-based methods [12]. Still, both model-based and data-driven control approaches need accurate, reliable sensing of the robot's posture and extraction of meaningful quantities from soft sensor readings to connect them to the soft robot's complex shape. This requires dealing with the soft sensors' dynamic behaviors and the continuum nature of soft robots, simultaneously.

Recent works have used learning-based methods for dealing with complex sensor characteristics in soft systems [3], [6], [13]–[17]. However, all of them deal with simple systems, often without actuation capabilities and/or clear directions for how these results can be scaled or transferred to the design and control of more complex soft robots. To the best of our knowledge, only two works have predicted some kind of actuated soft robot information from embedded soft sensors [6], [17]. In [6], a soft robotic hand can predict flexion, lateral, and twist deformations in its fingers, and in [17], a recurrent neural network (RNN) is used to learn the tip position and contact forces of a single soft actuator. Neither of these works achieve a description of full 3D configuration, which is essential for a complete understanding of soft robot behavior and, thus, its eventual control.

To address these interwoven challenges, we present a framework for rapidly equipping existing soft robots with distributed, soft, piezoresistive sensors, and enabling them to perceive their 3D configuration via deep learning. Using feedback from the soft sensors and motion capture data as model inputs and ground truth, respectively, we trained long short term memory (LSTM)-based neural networks to predict configuration parameters for a soft robotic arm using a new analytical description of soft robot shape. The arm is a three-segment, modular soft robot with 12 distributed fluidic actuators, each with corresponding sensors (Fig. 1). One of our trained networks, selected for having the

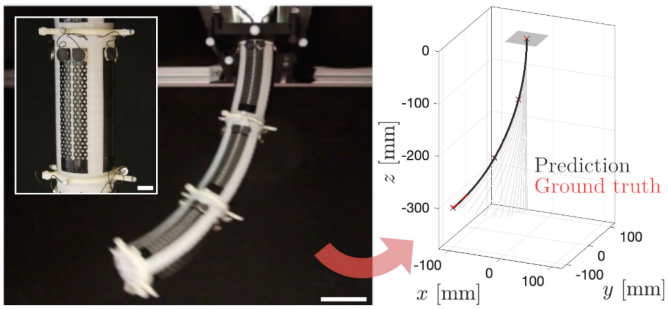

Fig. 1. Proprioception of 3D Configuration in a Soft Robot. The soft robotic arm (left) has a proprioceptive skin of distributed soft sensors (scale bar = 5 cm), which are fabricated via kirigami from off-the-shelf conductive silicone (inset, scale bar = 1 cm). A trained neural network provides a prediction of kinematic parameters from sensor signals, which are used to determine a 3D representation of the soft robot's configuration (right).

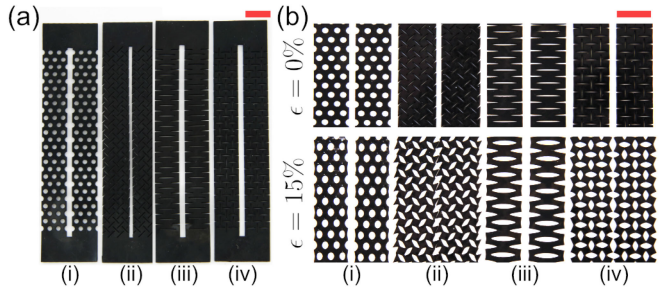

Fig. 2. Kirigami Sensors. (a) Four kirigami cut patterns (*circle* (i), *45/−45* (ii), *1D* (iii), *0/90* (iv)) are used to fabricate soft sensors from off-the-shelf sheets of electrically conductive silicone. (b) Sensors are shown at 0 and 15% strain. (Scale bars = 1 cm).

lowest overall root mean square error ($RMSE$) in predicted kinematic parameters, provided predictions of 3D configuration that reasonably captured the steady-state geometry of the soft robot arm, even with inputs from our soft sensors exhibiting highly hysteretic, non-monotonic behaviors. Current predictions do not completely capture the full dynamic 3D configuration of the arm. Still, our framework - which is modular and universal in the sense that it can be easily extended to a variety of soft systems - achieves an essential step in on-line, proprioception of body shape in soft robots. Altogether, this letter contributes:

1) A kirigami-inspired design and fabrication strategy for rapidly integrating piezoresistive silicone sensors onto existing soft robots as a proprioceptive skin,
2) A new kinematic description connecting the whole soft robot's shape to low dimensional features that can be easily learned by a data-driven strategy, and
3) A deep learning approach to predicting an approximation of the complete 3D continuum shape of a complex, soft sensorized robot arm.

II. Design Overview and Rationale

A. Sensorization Strategy and Soft Robot Design

Molded fluidic elastomer actuators (FEAs) popularly used in soft robotics can be sensorized with flexible curvature sensors [3], liquid metal sensors [6], soft piezoresistive composites [7], elastomeric waveguides [5], or ionically conductive gels [4], [8]. Soft robotic sensors offer important trade-offs between design, performance, and manufacturability. However, they must all be integrated into the actuator during fabrication. We created a sensor design and fabrication strategy that enables us to bypass any modifications to existing soft robotic fabrication procedures. Our approach for designing soft, piezoresistive silicone sensors that can easily be distributed across the surface of a soft robot body is enabled by kirigami (Fig. 2). Sensors are laser cut from off-the-shelf sheets of conductive silicone elastomers used in electromagnetic interference shielding. They can be covalently bonded without adhesives to the surface of a soft robot by plasma treatment [18]. Neither materials synthesis/handling nor specialized equipment beyond a hand-held corona generator and laser cutter are needed, allowing others to easily adopt our technique.

We distributed 12 kirigami sensors on an elephant trunk-inspired soft robotic arm, as shown in Fig. 1. The arm is based on a previously reported, sensorless design, for which a feedback controller based on a rigorous mathematical model has already been developed [10]. Four kirigami patterns - called *circle, 45/−45, 1D,* and *0/90* - were used as cut designs for the soft sensors (Fig. 2). The soft arm consists of three individual silicone segments with four embedded FEAs arranged in a cross-like configuration. Complex systems with distributed actuation networks with 3D maneuverability can be achieved by serially adding multiple segments together. With our methods, any number of sensors can be distributed across the soft robot's body in practice, and more sophisticated sensory skins can be achieved in the future through more complex kirigami forms, materials, and composite structures. For this study, we arbitrarily configured each segment such that sensors with one of each of the four unique kirigami patterns lie over the segment's four embedded actuators. Segments are assembled such that the same sensor designs are arranged identically across segments. Because an investigation into how specific sensor designs influence configuration prediction are beyond the scope of the present paper and warrant a separate study, we maintain this arbitrary sensor configuration.

B. Encoding the Geometry of the Soft Robot: 3D Kinematic Description

The goal of the kinematic model is to connect orientation and position of each point along the robot to a reduced set of variables, thus analytically encoding the geometric features of the robot. Knowing these variables' values at a given time and the kinematic description is equivalent to knowing the complete (approximated) shape of the robot. This reduced set of variables (configuration hereinafter) can then be learned by the deep learning part of our method, a task that would be unfeasible with the complete continuum shape.

We consider the central axis of the robot divided in n_{segment} segments, corresponding to the soft robot design discussed above. Fig. 3 depicts a schematic representation of the robot with main quantities highlighted. $\{S_0\}$ is the base frame of the robot. $\{S_i\}$ is the frame fixed to the tip of the $i-$th segment, in such a way that $\{S_i\}$ and $\{S_{i-1}\}$ are aligned when the robot is in a straight configuration, the local z axis is tangent to the robot, and the local x axis points in the direction of the first (i.e., *circle*) sensor. We call $s_i \in [0, 1]$ the coordinate parametrizing points along the $i-$th segment. The distance of a point with coordinate

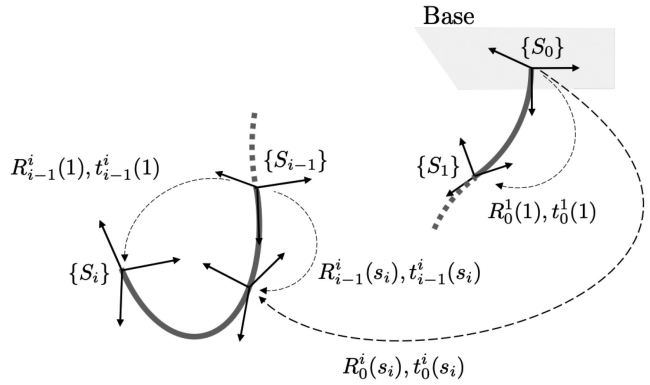

Fig. 3. Kinematic Model for the Soft Robotic Arm. Local reference frames are indicated by their principal axes. Transformations between reference frames are depicted as dashed arrows. The rotation matrix and the translation vector implementing the transformation are reported close to the arrow. $\{S_i\}$ are the main reference frames along the robot, which are determined with our motion capture system.

s_i from the base of the segment is s_i times the length of the segment itself: $s_i = 0$ indicates the beginning of the segment, and $s_i = 1$ its end. Each material point has its own orientation and position in space, which we express using a rotation matrix[1] and a triple of real numbers. We call them $R^i_0(s_i) \in SO(3)$ and $t^i_0(s_i) \in \mathbb{R}^3$ when expressed in base coordinates $\{S_0\}$, and $R^i_{i-1}(s_i) \in SO(3)$ and $t^i_{i-1}(s_i) \in \mathbb{R}^3$ when expressed in local coordinates $\{S_{i-1}\}$.

To derive such a description, we consider four lengths on the $i-$th segment running parallel with each sensor: $L_{i,1}, L_{i,2}, L_{i,3}, L_{i,4} \in \mathbb{R}^+$. These quantities can be linearly combined in the following manner: $\Delta_{\mathrm{x},i} = \frac{L_{2,i}-L_{1,i}}{2}$, $\Delta_{\mathrm{y},i} = \frac{L_{4,i}-L_{3,i}}{2}$, $\delta L_i = \frac{1}{4}\sum_{j=1}^{4} L_{j,i} - L_{0,i}$, where $L_{0,i} \in \mathbb{R}$ is the length of the segment when the robot is at rest. Intuitively, $\Delta_{\mathrm{x},i}$ carries information on the extent of segment bending in the x direction in $\{S_{i-1}\}$ coordinates. Similarly, $\Delta_{\mathrm{y},i}$ quantifies segment bending toward the local y. Finally, δL_i measures the segment's net change in length.

It can be proven that $\Delta_{\mathrm{x},i}$, $\Delta_{\mathrm{y},i}$, δL_i provide a complete parametrization of the configuration manifold of the $i-$th soft segment under the hypothesis of locally constant curvature and homogeneous elongation within the segment. In other words, these variables carry all the information we need for describing the configuration of the $i-$th segment $q_i = [\Delta_{\mathrm{x},i},\ \Delta_{\mathrm{y},i},\ \delta L_i]^{\mathrm{T}} \in \mathbb{R}^3$. The formal proof of this statement is provided in Ref. [19]. The actual kinematic description of the segment is provided by $R^i_{i-1}(s_i; q_i) =$

$$\begin{bmatrix} 1+\frac{\Delta^2_{\mathrm{x},i}}{\Delta^2}\mathrm{C}_i(s_i;\Delta_i) & \frac{\Delta_{\mathrm{x},i}\Delta_{\mathrm{y},i}}{\Delta^2}\mathrm{C}_i(s_i;\Delta_i) & -\frac{\Delta_{\mathrm{x},i}}{\Delta}\mathrm{S}_i(s_i;\Delta_i) \\ \frac{\Delta_{\mathrm{x},i}\Delta_{\mathrm{y},i}}{\Delta^2}\mathrm{C}_i(s_i;\Delta_i) & 1+\frac{\Delta^2_{\mathrm{y},i}}{\Delta^2}\mathrm{C}_i(s_i;\Delta_i) & -\frac{\Delta_{\mathrm{y},i}}{\Delta}\mathrm{S}_i(s_i;\Delta_i) \\ \frac{\Delta_{\mathrm{x},i}}{\Delta}\mathrm{S}_i(s_i;\Delta_i) & \frac{\Delta_{\mathrm{y},i}}{\Delta}\mathrm{S}_i(s_i;\Delta_i) & 1+\mathrm{C}_i(s_i;\Delta_i) \end{bmatrix},$$

$$t^i_{i-1}(s_i;q_i) = \frac{d_i s_i (L_{0,i}+\delta L_i)}{\Delta^2} \begin{bmatrix} \mathrm{C}_i(s_i;\Delta_i)\Delta_{\mathrm{x},i} \\ \mathrm{C}_i(s_i;\Delta_i)\Delta_{\mathrm{y},i} \\ \mathrm{S}_i(s_i;\Delta_i)\Delta_i \end{bmatrix},$$

where $\Delta = \sqrt{\Delta^2_{\mathrm{x},i} + \Delta^2_{\mathrm{y},i}}$, $\mathrm{S}_i(s_i;\Delta_i) = \sin(\frac{s_i\Delta}{d_i})$, $\mathrm{C}_i(s_i;\Delta_i) = (\cos(\frac{s_i\Delta}{d_i}) - 1)$, and $d_i \in \mathbb{R}^+$ is the radius of a segment's cross-section. These equations connect the configuration q_i to the posture of each point s_i, expressed in coordinates $\{S_{i-1}\}$. They can be derived from the corresponding elements in the standard parametrization [20], by exploiting a transition map connecting the bending and curvature variables used there with the $\Delta_{\mathrm{x},i}$, $\Delta_{\mathrm{y},i}$ introduced here. Ref. [19] provides more details.

R^i_{i-1} and t^i_{i-1} can be further reorganized into the $SE(3)$ element[2]

$$T^i_{i-1}(s_i;q_i) = \begin{bmatrix} R^i_{i-1}(s_i;q_i) & t^i_{i-1}(s_i;q_i) \\ 0\ \ 0\ \ 0 & 1 \end{bmatrix}. \tag{1}$$

The posture of the point s_i on the $i-$th segment of the soft robot, in base coordinates, can then be derived using the standard $SE(3)$ algebra

$$T^i_0(s_i,q) = T^1_0(1,q_1)\ldots T^{i-1}_{i-2}(1,q_{i-1})T^i_{i-1}(s_i,q_i). \tag{2}$$

Finally, the continuum shape of the segment in configuration q_i is represented by the sub-set $\mathcal{S}_i = \{T^i_0(s_i;q),\ \forall s_i \in [0,1]\}$, while the shape of the whole robot $\mathcal{S}$ is the collection of these sets $\forall i \in \{1\ldots n_{\mathrm{segment}}\}$.

C. Inversion Problem

The ultimate goal of our framework is to estimate the robot's continuum shape, $\mathcal{S}_i$, through the information that can be retrieved from the soft sensor readings, v_i. This problem is studied in dynamic system theory as system inversion [21]. However, existing techniques in the field are not suitable for soft robotics, since they require reliable and relatively simple models of the dynamic relationship between input and output. Such models would take the form of the dynamical system $\dot{\xi}_i = F_i(\xi_i, \mathcal{S}_i)$, $v_i = H_i(\xi_i, \mathcal{S}_i)$, to be connected in series to the actual dynamics of the soft robot. The variable of interest $\mathcal{S}_i$ serves here as the input, and the sensor readings v_i as output. ξ_i are internal states, taking into account short and long term memory effects. The former include low pass behaviors common to many soft sensing systems. The latter include the hysteretic behaviors of soft sensor composites. Finally, F_i, H_i are generic applications, describing the internal dynamics and the output characteristics, respectively.

The complexity of the inversion problem can be strongly reduced by relying on salient geometric features that can be extracted from the kinematic model, as introduced in the previous subsection. Thus, we substitute the parametrization of $\mathcal{S}_i$ in terms of q_i into F_i and H_i, getting the following set of standard ordinary differential equations

$$\dot{\xi}_i = f_i(\xi_i, q_i), \quad v_i = h_i(\xi_i, q_i), \tag{3}$$

where $q_i \in \mathbb{R}^3$ is the parametrization of the $i-$th segment, as introduced in the previous section. $f_i : \mathbb{R}^{n_{\mathrm{internal}}} \times \mathbb{R}^3 \to \mathbb{R}^{\mathrm{internal}}$ defines the internal dynamics of the sensor, and $h_i : \mathbb{R}^{n_{\mathrm{internal}}} \times \mathbb{R}^3 \to \mathbb{R}^{n_{\mathrm{sensor}}}$ is the output function.

Finally, our goal is to regress the low dimensional robot configuration q_i from the sensor readings v_i, without having a direct knowledge of the internal states ξ_i. Deriving adequately precise

[1]Representation of one element of the Special Orthogonal group of dimension 3, $SO(3)$.

[2]Element of the Special Euclidean group of dimension 3.

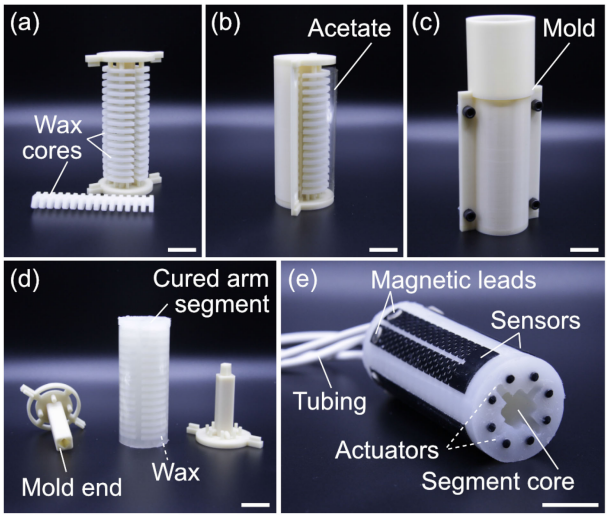

Fig. 4. Sensorized, Soft Robotic Arm Segment Fabrication. (a) Molded wax cores are loaded into a 3D printed mold. (b) An acetate film lines the mold's inner wall. (c) Silicone is cast into the mold and cured. (d) A cured arm segment is removed, and inner wax cores are melted out. (e) The final arm segment with plasma bonded sensors is ready for arm assembly. (Scale bars = 2 cm.)

models of f_i and h_i remains difficult, if not impossible. Instead, we turn to black box nonlinear system identification [22], specifically deep learning. Since these techniques ignore the causal relationship between data [23], we can directly regress an approximation of the inverse system instead of estimating f_i and h_i and inverting them:

$$\dot{\psi}_i = \kappa_i(\psi_i, v_i), \quad \tilde{q}_i = \iota_i(\psi_i, v_i),$$
$$\text{s.t. } \tilde{q}_i \simeq q_i, \quad \text{if } v_i = h_i(\xi_i, q_i). \tag{4}$$

where $\psi \in \mathbb{R}^{n_{\text{internal}}}$ is the internal state of the inverse model, and $\tilde{q}_i \in \mathbb{R}^3$ the estimation of the segment configuration. κ_i and ι_i are the functions to be learned from data, specifying the internal dynamics of the inverse model and its output function, respectively. Given the dynamic, multi-scale nature of (3) and (4), we use LSTM-based neural networks as approximators. We do not report the structure of κ_i and ι_i for LSTM architectures (see [24] for details).

III. Methods

A. Soft Sensor and Soft Sensorized Robot Fabrication

Soft strain sensors with *circle, 45/−45, 1D,* and *0/90* cut patterns are fabricated from 0.5 mm-thick, electrically conductive silicone sheets (Shore A hardness of 65+/−5, Silex Silicones Ltd) via laser cutting (CO_2 laser, Universal Laser Systems). The conductive silicone is filled with carbon black and has an approximate volume resistivity of 5 Ω-cm. All sensors are cleaned by alternating 15-min intervals of sonication in isopropanol and acetone.

Arm segments are molded from Dragon Skin 30 (Smooth-On) using lost-wax casting methods with 3D printed molds [10] (Fig. 4(a)–(d)). Molds are lined with 0.1 mm-thick acetate films to create a smooth surface that facilitates plasma bonding of sensors to the arm segments (Fig. 4(b)). After curing, the kirigami strain sensors are plasma bonded to the surface of the segments using a hand-held, atmospheric corona treater (Electro-Technic Products) [18]. The corona treated sensor is firmly pressed against the arm segment and left undisturbed. After 24 hr of bonding at room temperature, Kapton tape is laid over all sensors, and the wax is removed through melting by heating segments in a 90 °C oven, followed by complete immersion in boiling water, both for 1 hr [10]. The Kapton tape is removed, and neodymium rare earth magnets are glued at the ends of each strain sensor using a conductive, silver-filled silicone adhesive (SS-26, Silicone Solutions). Steel disks are soldered to 28 gauge wire and placed as break-away leads on each magnet. Silicone tubing is glued into the four actuator inlets of each segment using Sil-Poxy (Smooth-On) (Fig. 4(e)). All leads and actuator tubing are threaded through the segment cores, and the segments are glued together to form the arm.

Algorithm 1: Extract Posture q From One Set of Motion Capture Readings $T_0^1(1) \dots T_0^{n_{\text{segment}}}(1)$.

$T_0^0(1) \leftarrow I_{4\times4}$ ▷ Initialization
for $i \leftarrow 1, n_{\text{segment}}$ **do**
  $T_0^i(1) \leftarrow \text{ReadMotionCapture}(i)$
for $i \leftarrow 1, n_{\text{segment}}$ **do** ▷Posture extraction
  $T_{i-1}^i(1) \leftarrow (T_0^{i-1}(1))^{-1} T_0^i(1)$
  $\Delta_i \leftarrow \arccos(R_{i-1}^i[3,3]) d_i$
  $\delta L_i \leftarrow t_{i-1}^i[3]\Delta_i / [d_i \sin(\Delta_i/d_i)] - L_0$
  $\Delta_{\text{x},i} \leftarrow t_{i-1}^i[1]\ \Delta_i^2 / [d_i(L_{0,i} + \delta L_i)(R_{i-1}^i[3,3] - 1)]$
  $\Delta_{\text{y},i} \leftarrow t_{i-1}^i[2]\ \Delta_i^2 / [d_i(L_{0,i} + \delta L_i)(R_{i-1}^i[3,3] - 1)]$

B. Soft Robot Operation and Characterization

The test bed for operating and characterizing the soft robot includes a 16-line pressure manifold for pneumatic actuation (Festo Corporation), readout electronics, digital acquisition unit (DAQ, USB-6212, National Instruments), and motion capture setup (OptiTrack). Inflation pressures of 0 to 120 kPa are used for actuation. Sensor feedback is obtained with voltage dividers, whose output voltages are recorded with the DAQ. Each sensor's resistance, R_{S}, is given by $R_{\text{S}} = \rho(l/A)$, where ρ is the resistivity of the conductive silicone, l is the sensor length, and A is the sensor cross-sectional area. As a sensor deforms, R_{S} changes. The voltage divider converts R_{S} to an output voltage, V, where $V = V_{\text{CC}} R_2/(R_{\text{S}} + R_2)$. V_{CC} and R_2 are 9 V and 7.5 MΩ, respectively. If V_0 is a sensor's initial voltage, then ΔV decreases as R_{S} increases (where $\Delta V = V - V_0$). Optical tracking markers are placed at the arm's base and at the end of each arm segment to designate $\{S_i\}$. `Motive` motion capture software (OptiTrack) coordinates all motion capture data and sensor inputs from the DAQ. Camera and sensor sampling rates are 100 frames/sec and 1000 Hz, respectively.

The motion capture provides direct measurements of position and posture of each segment's end, expressed in base coordinates. To extract q from these data, we express the postures in local coordinates and invert (1) for $s_i = 1$. The operations needed to perform this task are reported in Algorithm 1 as pseudo-code. We call $t_{i-1}^i[j]$ the $j-$th element of the translation part of $T_{i-1}^i(1)$, and $R_{i-1}^i[j,k]$ the element (j,k) of the rotation part of the same matrix.

We use two discrete actuation patterns to characterize our soft sensorized robot: *inflation sweeps* and *inflation steps*. During inflation sweeps, segment S_3 is inflated to 120 kPa then deflated

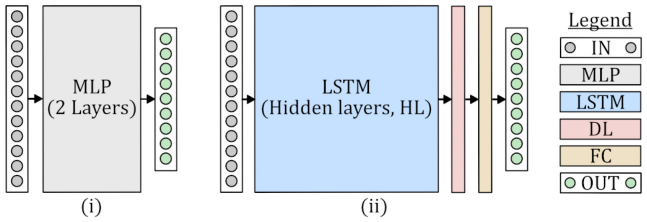

Fig. 5. Neural Network Designs. Schematics for Architectures 1 (i) and 2 (ii), indicating input (IN), multi-layer perceptron (MLP), long short term memory (LSTM, with HL hidden layers), fully connected (FC), dropout (DL), and output (OUT) layers.

back to 0 kPa at 10-kPa increments held for 3 sec each. The inflation pressure is swept sequentially in Actuators 1, 2, 3, and 4 of segment S_3 (i.e., the actuators beneath the *circle, 45/−45, 1D,* and *0/90* sensors, respectively). During inflation steps, the same actuator is inflated in all segments to 100 kPa and held for 15 sec before deflation to 0 kPa, which is held for 15 sec. This step inflation is applied sequentially to Actuators 1 through 4 in all segments.

For training and validation purposes, we also collect data from the arm as it undergoes several additional actuation sequences. These actuation motifs are called *random actuations*, *swing inflation*, and *extension* sequences. In random actuations, all 12 actuators in the arm are randomly actuated at inflation pressures of 50 to 120 kPa and changed every 3 sec for 100 cycles. Swing inflation sequences are identical to inflation steps, except two neighboring actuators are inflated simultaneously, cycling between Actuators 1+2, 2+3, 3+4, and 4+1. Fig. 1 shows the arm with Actuators 3+4 inflated at 100 kPa during a swing inflation sequence. Extension sequences involve inflating all actuators in S_1 to 50 kPa for 15 sec and then deflating to 0 kPa for 15 sec, repeating for S_2 then S_3.

C. Neural Network Design and Training

All neural network design, training, and validation was performed with `MATLAB`'s `Deep Learning Toolbox`. Two types of neural networks, noted Architectures 1 and 2, were designed (Fig. 5). The input layer features of each architecture are time series responses from a single sensor, while the regression output layers provides values corresponding to predictions of the q_i parameters. With Architecture 1, we explore a quasi-static simplification of the inversion problem discussed in Section II C, i.e. $v_i \simeq h_i(q_i)$, which ignores the dynamic, hysteretic behavior of the soft sensors. This leads to the simpler static inverse model $\tilde{q}_i = \iota_i(v_i) : \mathbb{R}^{n_{\text{sensors}}} \rightarrow \mathbb{R}^{3n_{\text{segments}}}$ such that $\iota_i(h_i(q_i)) \simeq q_i$. To model this, Architecture (Arch.) 1 is based on a shallow MLP layer with two hidden layers, followed by a set of low pass filters. In contrast, Arch. 2 contains an LSTM layer, rendering it sensitive to the sensors' dynamic responses, followed by dropout and fully connected layers. Both architectures have 12 inputs and 9 outputs (i.e., 12 sensor signals and 9 q_i parameters, respectively).

Training and validation sets are comprised of data collected from unique sequences of inflation steps (ISt), inflation sweeps (ISw), random actuations (RAs), swing inflations (SwI), and extension (Ext). The training data set includes a concatenated sequence of the following: RAs, ISt*, Ext, RAs, ISw, ISt*, RAs, RAs, ISt*, RAs, ISw, ISt*, SwI*, ISt, ISt, SwI, SwI*, ISt, SwI, ISt, RAs, ISt, SwI (where * indicates that the sequence was only performed on S_3). The validation data set includes a concatenated sequence of the following: RAs, ISw, ISt. While we use repeat actuation sequences in the training and validation sets, note that the sensor response is different in each due to the dynamics and deformation history-dependent nature of the soft sensors' responses. Also, RAs are the longest sequences, especially compared to the ISt and ISw motifs, ensuring that random behaviors from the arm are prominently featured in the data sets. Training and validation data sets amount to approximately 80 and 13 min of data, respectively, with both training and validation sets including data over intervals where the arm was at rest. Following the practices used in [17], we use the validation set as our test set for evaluating neural network performance, due to the challenge of collecting extensive data sets with soft robots.

Three networks of each architecture were trained with various sets of hyperparameters. Networks are trained using the Adam optimizer for a maximum of 1000 epochs with a validation frequency of 10. Over-fitting is minimized by L2 regularization and a *patience* of 20. In Arch. 1, the number of hidden neurons in both the first and second layers were varied in [8, 16, 24, 32], in all combinations. In Arch. 2, the dropout layer rates were varied in $[0.1, 0.2, 0.5]$, and the number of hidden states were varied in [10, 100, 200]. The root mean square error on the prediction from the test (i.e., validation) set is evaluated for each network as $RMSE = \sqrt{\sum_{j=0}^{12} \sum_{k=0}^{n_{\text{sample}}} \frac{||q[j,k]-\tilde{q}[j,k]||_2^2}{12 n_{\text{sample}}}}$ where $n_{\text{sensor}} n_{\text{segment}} = 12$, $q[j,k]$ is the $j-$th element of the configuration at the sample k, and $\tilde{q}[j,k]$ its estimation.

IV. Experimental Results

A. Soft Robot Characterization

Fig. 6(a) illustrates the steps of an inflation sweep actuation pattern in segment S_3. Figs. 6(b) and 6(c) show ΔV for all sensors and q_3 parameters ($\Delta_{x,3}$, $\Delta_{y,3}$, and δL_3) as a function of inflation pressure, respectively, during inflation sweeps in each actuator. All sensors exhibit hysteretic and non-monotonic behavior (Fig. 6(b)). In general, the sensor responses vary significantly for different kirigami cut patterns. Also, all sensors decrease in R_S (i.e., increase in ΔV) when the opposing actuator is inflated and compresses the sensor. Otherwise, all other trends are less obvious.

First, all sensors exhibit a similar trend when the actuators they lay above are inflated: the sensors begin to decrease in V, but begin increasing in V around 80 to 100 kPa (Fig. 6(b)). One would expect that increasing sensor strain during inflation-induced bending would lead to monotonically increasing R_{S} (i.e., decreasing V). However, the structural dynamics of the percolated networks of conductive carbon black particles in the sensors can result in a changing ρ, yielding non-monotonic sensor responses [25]. Monotonic changes in R_{S} are observed when sensors lie perpendicular to bending direction. Finally, we observe expected changes in $\Delta_{x,3}$, $\Delta_{y,3}$, and δL_3: increasing inflation pressures at Actuators 1, 2, 3, and 4 induce bending in the direction along the $-x$, $-y$, x, and y axes of $\{S_2\}$, respectively (Fig. 6(c)), and $\delta L_3 > 0$ during bending.

Figs. 7(a) and 7(b) provide the sensor readings and the posture parameters, respectively, for all segments during inflation step actuation patterns. From Fig. 7(a), the sensors' nonlinear, deformation history-dependent responses and their evolution when the robot is at rest are clearly shown. In Fig. 7(b), the oscillatory

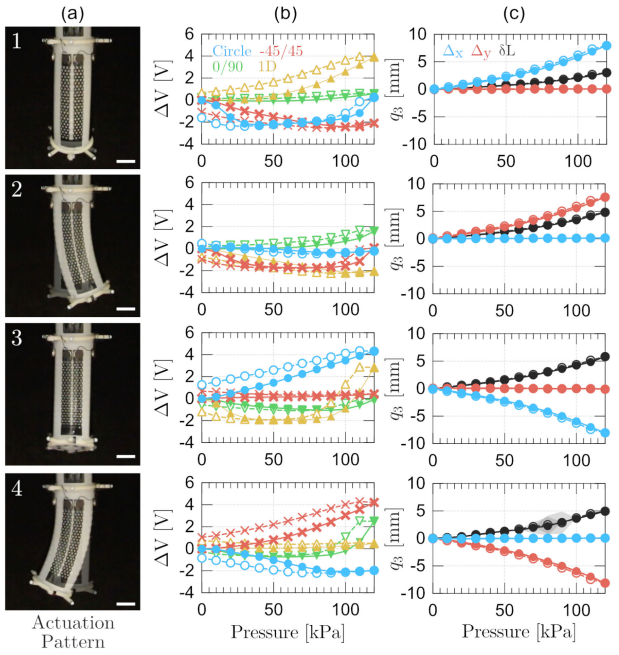

Fig. 6. Characterization of a Single Arm Segment. (a) Photographs show the arm's third segment (S_3) actuating at 120 kPa inflation pressure, sequentially applied from Actuator 1 (top) to Actuator 4 (bottom). Transparent overlays of the segment at 0 kPa are included for visual reference of deflated state. (Scale bars = 2 cm). (b) Voltage change, ΔV, for each sensor and (c) configuration values versus inflation pressure are provided for each actuation pattern shown in (a). In both (b) and (c), shaded error bands indicate standard deviation ($n = 3$); filled lines with solid markers and dashed lines with empty markers indicate data from inflation and deflation cycles, respectively.

TABLE I
RMSE OF VALIDATION SET PREDICTIONS FOR ARCH. 1 (mm)

	Hidden neurons, 1st layer												
	8			16			32			64			
2nd layer 8	5.9	5.7	6.4	5.9	6.0	11	7.2	7.2	6.4	6.1	6.7	6.3	
16	5.6	6.0	6.3	8.2	6.2	6.5	7.1	6.9	8.0	8.6	8.6	10	
32	6.1	6.0	7.3	7.8	8.4	8.3	9.0	10	9.0	11	10	8.1	
64	7.7	7.4	7.1	9.0	8.6	7.6	9.0	8.0	11	11	15	10	

TABLE II
RMSE OF VALIDATION SET PREDICTIONS FOR ARCH. 2 (mm)

	Hidden layers									
	10			100			200			
Dropout 0.1	1.93	2.22	2.03	1.54	1.52	1.67	1.27	1.70	1.46	
0.2	2.23	1.77	2.01	1.50	1.42	1.71	1.68	1.37	1.37	
0.5	2.22	2.10	2.11	1.45	1.53	1.35	1.46	1.53	1.43	

swinging of the actuated arm is captured. Due to inertia and the arms' mechanical properties, the swinging of the arm should be most pronounced at S_1 and minimal for S_3. These features are correctly reflected in our kinematic model. We note that only some of the sensors' responses in Fig. 7(a) reflect the arm's swinging motion.

B. Neural Network Design and Training

Tables I and II provide $RMSE$ values (in units of mm) for all trained networks. Overall, LSTM-based Arch. 2 provides lower $RMSE$ than MLP-based Arch. 1, presumably due to Arch. 2's sensitivity to the sensors' dynamic responses over time. For Arch. 2, changes in dropout rates did not have clear

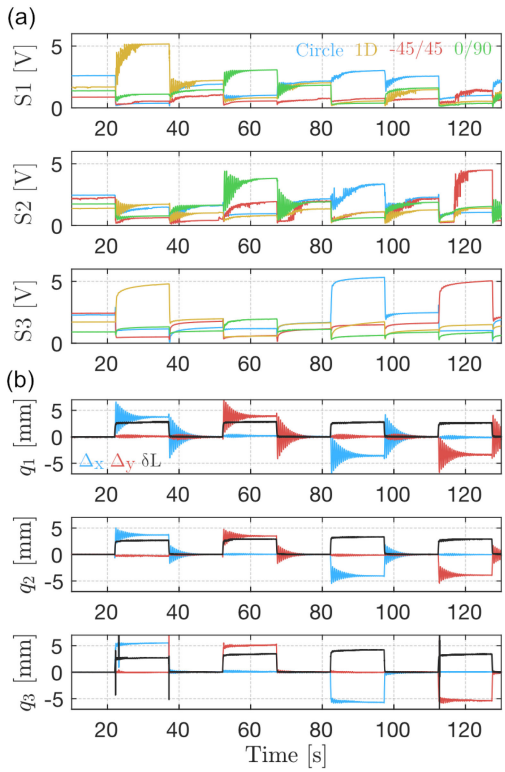

Fig. 7. Characterization of the Full Arm. In a sequential pattern with all actuators, moving from inflating Actuator 1 of all segments to Actuator 4 of all segments, the arm is inflated for 15 sec at 100 kPa and deflated for 15 sec. (a) The output voltage from sensors in the first ($S1$), second ($S2$), and third ($S3$) segments and (b) the configuration values for the first (q_1), second (q_2), and third (q_3) segments are provided versus time (all plots share the same x-axis.)

trends for influence on $RMSE$. Finally, for Arch. 2 networks with a given dropout rate, more hidden layers in the LSTM layer typically resulted in lower $RMSE$. An Arch. 2 network with a dropout rate of 0.1 and 200 hidden layers in the LSTM layer provided the lowest $RMSE$ of all trained networks, at 1.27 mm. We emphasize that these hyperparameters are optimal for our specific validation data set here, not, necessarily, for a generic new set of data.

C. Validation of 3D Configuration

With our trained neural network (Arch. 2 with $RMSE$ of 1.27 mm), sensor voltage inputs are used to predict $\tilde{q}_i$ for all segments during inflation steps (Fig. 8) and random actuation cycles (Fig. 9). Fig. 8(a) shows four near-steady state configurations after swinging about, for each actuation step of the inflation step sequence. Fig. 8(b) shows the corresponding ground truth 3D configurations computed from motion capture data compared with the predicted configurations. Fig. 8(c) shows ground truth parameters, q_i, and estimated ones, $\tilde{q}_i$, over time for the second step inflation, in which Actuator 2 in all segments is inflated to 100 kPa. $\tilde{q}_i$ is in relatively close agreement with q_i during step inflation cycles. The network dampens the amplitude of the oscillatory, dynamic motion in q_1 and q_2 (Fig. 9(c)), though it does predict the general steady-state behavior of the arm's motion during these actuation steps. Thus, the network satisfactorily predicts the steady-state configuration of the arm but not its full dynamics.

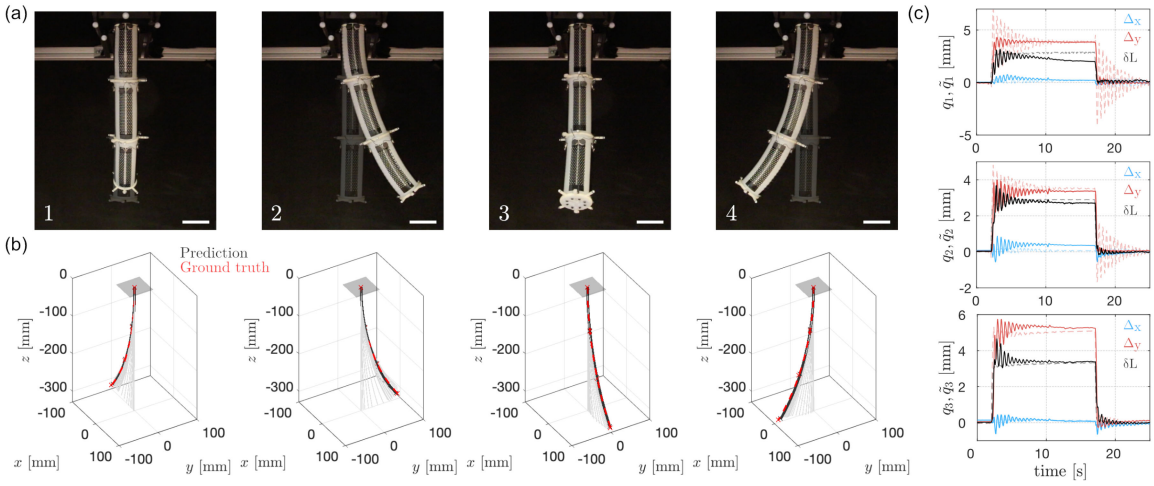

Fig. 8. Validation Results on Full Arm Swings. (a) Photographs and (b) respective plots of ground truth and predicted configurations are provided for the same actuation pattern in Figure 7, in which the arm swings in different directions by inflating one of the same actuator networks in each segment from 0 to 100 kPa for 15 sec. In (b), the grey curves represent ground truth configurations over several seconds; the red and black curves represent the near-steady state configuration and predicted configuration from the neural network, respectively. (c) Plots of ground truth (dashed lines, q_i) and predicted configuration parameters (solid lines, $\tilde{q}_i$) versus time are provided for segments S_1 (top), S_2 (middle), and $S3$ (bottom) during the actuation shown in the second column of (a) and (b). Scale bars = 5 cm.)

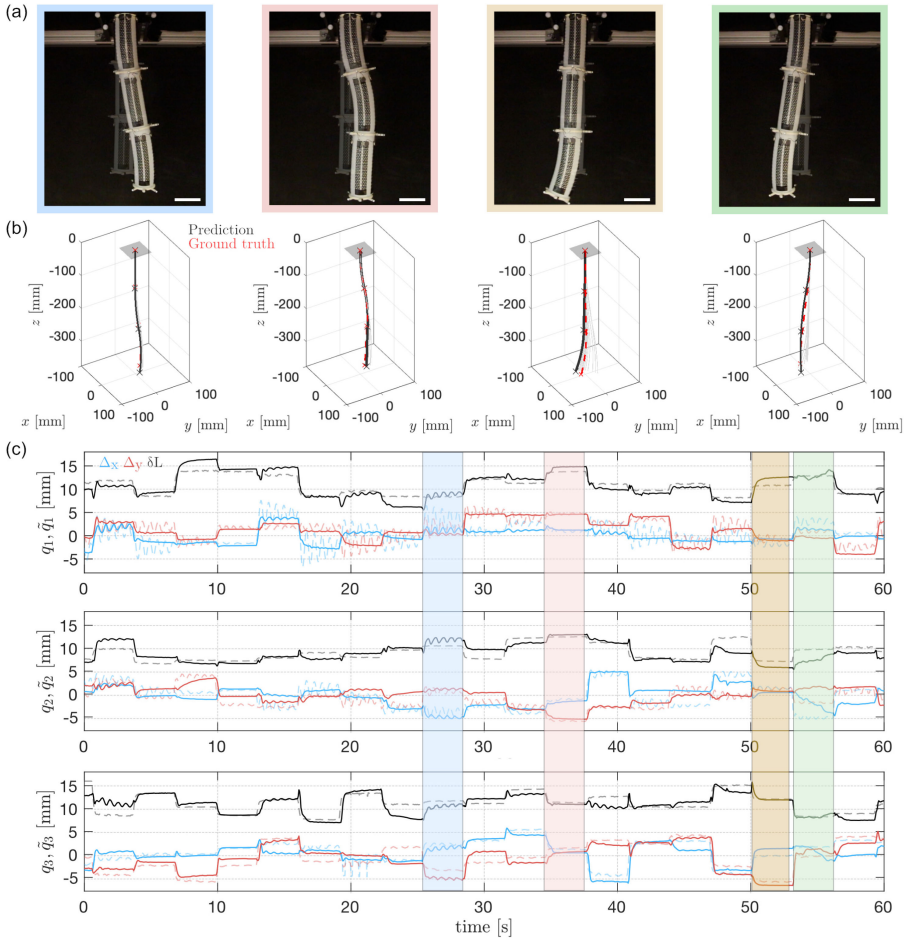

Fig. 9. Validation Results on Random Arm Actuations. (a) Photographs and (b) respective plots of ground truth and predicted configurations are provided for representative soft robot poses during random actuation cycles. In (b), the grey curves represent ground truth configurations over several seconds; the red and black curves represent the near-steady state configuration and predicted configuration from the neural network, respectively. (c) Plots of ground truth (dashed lines, q_i) and predicted configuration parameters (solid lines, $\tilde{q}_i$) versus time are provided for segments S_1 (top), S_2 (middle), and S_3 (bottom) as the arm is actuating by random changes of inflation pressure at 3 sec intervals. Colored regions correspond to the poses indicated by frame color for the photographs in (a). (Scale bar = 5 cm.)

Fig. 9 provides data regarding predictions of $\tilde{q}_i$ as the arm is randomly actuated at inflation pressures of 50 to 120 kPa changed every 3 sec (Fig. 9(a)). Again, as for Fig. 8(b), Fig. 9(b) shows the computed and predicted 3D configurations using posture parameters q_i and $\tilde{q}_i$. Fig. 9(c) shows q_i and $\tilde{q}_i$ for the course of this study, with colored regions indicating the specific postures shown in Figs. 9(a) and 9(b). A slight phase delay between $\tilde{q}_i$ and q_i is observed in Fig. 9(c). This behavior, not observed in Fig. 8(c), is likely due to the higher level of dynamicity during random actuations. We again see agreement between q_i and $\tilde{q}_i$ during random actuations, with the network better predicting steady-state configuration.

V. Conclusions

We have developed a framework for learning 3D configuration in a soft robot through distributed proprioception enabled by a soft sensor skin. We have shown an example of a trained RNN that reasonably predicts the steady-state configuration of our soft robotic arm during both prescribed and random actuation sequences, even with feedback from non-monotonic, hysteretic, soft piezoresistive sensors. Our neural network's inability to predict all details of the arm's full dynamic motion is likely the result of two important features: (i) the soft sensor's current sensitivity is insufficient for capturing small dynamic, oscillatory motions, and (ii) the use of voltage dividers as convenient readout electronics can unintentionally suppress dynamic details in sensor signals. The former represents a general limitation and challenge for the development of soft robotic sensors from soft materials, while the latter speaks to a general question of how data engineering also influences our approach. We are now working towards two follow-up investigations of how sensor and neural network designs influence our ability to predict the full, dynamic, 3D configuration of soft sensorized robots. First, we are exploring how our different kirigami sensor designs influence the accuracy of configuration prediction. Second, we are interested in exploring new neural network architectures that enable us to look at sensory feedback in both the time and frequency domains and/or better identify important sensor signal features in order to predict subtle configuration dynamics.

Overall, our results represent a fundamental step towards learning-based, soft robotic proprioception, which is necessary for addressing key challenges in closed-loop feedback control for soft robotics. We hope our framework can be readily used by others in the field to model, sensorize, and learn 3D configuration in other soft robots, expediting advances in soft robotic perception and control.

References

[1] D. Rus and M. T. Tolley, "Design, fabrication and control of soft robots," *Nature*, vol. 521, no. 7553, pp. 467–475, 2015.

[2] C. Della Santina, R. K. Katzschmann, A. Bicchi, and D. Rus, "Model-based dynamic feedback control of a planar soft robot: Trajectory tracking and interaction with the environment," *Int. J. Robot. Res.*, 2020. [Online]. Available: https://doi.org/10.1177/0278364919897292

[3] B. S. Homberg, R. K. Katzschmann, M. R. Dogar, and D. Rus, "Haptic identification of objects using a modular soft robotic gripper," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2015, pp. 1698–1705.

[4] C. Larson *et al.*, "Highly stretchable electroluminescent skin for optical signaling and tactile sensing," *Science*, vol. 351, no. 6277, pp. 1071–1074, 2016.

[5] H. Zhao, K. O'Brien, S. Li, and R. F. Shepherd, "Optoelectronically innervated soft prosthetic hand via stretchable optical waveguides," *Sci. Robot.*, vol. 1, no. 1, 2016, Art. no. eaai7529.

[6] V. Wall, G. Zoller, and O. Brock, "A method for sensorizing soft actuators and its application to the RBO hand 2," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2017, pp. 4965–4970.

[7] B. Shih *et al.*, "Custom soft robotic gripper sensor skins for haptic object visualization," in *Proc. IEEE Int. Conf. Intell. Robots Syst.*, 2017, pp. 494–501.

[8] R. L. Truby, R. K. Katzschmann, J. A. Lewis, and D. Rus, "Soft robotic fingers with embedded ionogel sensors and discrete actuation modes for somatosensitive manipulation," in *Proc. IEEE Int. Conf. Soft Robot.*, 2019, pp. 322–329.

[9] D. B. Camarillo, C. R. Carlson, and J. K. Salisbury, "Configuration tracking for continuum manipulators with coupled tendon drive," *IEEE Trans. Robot.*, vol. 25, no. 4, pp. 798–808, Aug. 2009.

[10] R. K. Katzschmann, C. Della Santina, Y. Toshimitsu, A. Bicchi, and D. Rus, "Dynamic motion control of multi-segment soft robots using piecewise constant curvature matched with an augmented rigid body model," in *Proc. IEEE Int. Conf. Soft Robot.*, Apr. 2019, pp. 454–461.

[11] A. Bajo, R. E. Goldman, and N. Simaan, "Configuration and joint feedback for enhanced performance of multi-segment continuum robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 2905–2912.

[12] T. G. Thuruthel, E. Falotico, F. Renda, and C. Laschi, "Model-based reinforcement learning for closed-loop dynamic control of soft robotic manipulators," *IEEE Trans. Robot.*, vol. 35, no. 1, pp. 124–134, 2018.

[13] G. Soter, A. Conn, H. Hauser, and J. Rossiter, "Bodily aware soft robots: Integration of proprioceptive and exteroceptive sensors," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2018, pp. 2448–2453.

[14] I. M. Van Meerbeek, C. M. De Sa, and R. F. Shepherd, "Soft optoelectronic sensory foams with proprioception," *Sci. Robot.*, vol. 3, no. 24, 2018, Art. no. eaau2489.

[15] S. Han, T. Kim, D. Kim, Y. Park, and S. Jo, "Use of deep learning for characterization of microfluidic soft sensors," *IEEE Robot. Autom. Lett.*, vol. 3, no. 2, pp. 873–880, Apr. 2018.

[16] C. Larson, J. Spjut, R. Knepper, and R. Shepherd, "A deformable interface for human touch recognition using stretchable carbon nanotube dielectric elastomer sensors and deep neural networks," *Soft Robot.*, vol. 6, no. 5, pp. 611–620, 2019.

[17] T. G. Thuruthel, B. Shih, C. Laschi, and M. T. Tolley, "Soft robot perception using embedded soft sensors and recurrent neural networks," *Sci. Robot.*, vol. 4, no. 26, 2019, Art. no. eaav1488.

[18] K. Haubert, T. Drier, and D. Beebe, "PDMS bonding by means of a portable, low-cost corona system," *Lab Chip*, vol. 6, pp. 1548–1549, 2006.

[19] C. Della Santina, A. Bicchi, and D. Rus, "On an improved state parametrization for soft robots with piecewise constant curvature and its use in model based control," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 1001–1008, Apr. 2020.

[20] R. J. Webster III and B. A. Jones, "Design and kinematic modeling of constant curvature continuum robots: A review," *Int. J. Robot. Res.*, vol. 29, no. 13, pp. 1661–1683, 2010.

[21] R. Hirschorn, "Invertibility of multivariable nonlinear control systems," *IEEE Trans. Autom. Control*, vol. 24, no. 6, pp. 855–865, 1979.

[22] L. Ljung, "Perspectives on system identification," *Annu. Rev. Control*, vol. 34, no. 1, pp. 1–12, 2010.

[23] G. Marcus, "Deep learning: A critical appraisal." [Online]. Available: https://arxiv.org/abs/1801.00631, Accessed on: Sep. 1, 2019.

[24] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[25] W. Luheng, D. Tianhuai, and W. Peng, "Influence of carbon black concentration on piezoresistivity for carbon-black-filled silicone rubber composite," *Carbon*, vol. 47, no. 14, pp. 3151–3157, 2009.