



Computing Pressure-Deformation Maps for Braided Continuum Robots

David Navarro-Alarcon^{1*}, Omar Zahra¹, Christian Trejo^{1,2}, Ernesto Olguín-Díaz² and Vicente Parra-Vega²

¹ Department of Mechanical Engineering, The Hong Kong Polytechnic University, Kowloon, Hong Kong, ² Robotics and Advanced Manufacturing Group, Center for Research and Advanced Studies of the National Polytechnic Institute Salttillo Unit, Mexico City, Mexico

This paper presents a method for computing sensorimotor maps of braided continuum robots driven by pneumatic actuators. The method automatically creates a lattice-like representation of the sensorimotor map that preserves the topology of the input space by arranging its nodes into clusters of related data. Deformation trajectories can be simply represented with adjacent nodes whose values smoothly change along the lattice curve; this facilitates the computation of controls and the prediction of deformations in systems with unknown mechanical properties. The proposed model has an adaptive structure that can recalibrate to cope with changes in the mechanism or actuators. An experimental study with a robotic prototype is conducted to validate the proposed method.

OPEN ACCESS

Edited by:

Matteo Cianchetti,
Sant'Anna School of Advanced
Studies, Italy

Reviewed by:

Chaoyang Song,
Southern University of Science and
Technology, China
Virgilio Mattoli,
Fondazione Istituto Italiano di
Tecnologia, Italy

*Correspondence:

David Navarro-Alarcon
david.navarro-alarcon@polyu.edu.hk

Specialty section:

This article was submitted to
Soft Robotics,
a section of the journal
Frontiers in Robotics and AI

Received: 21 August 2018

Accepted: 14 January 2019

Published: 05 February 2019

Citation:

Navarro-Alarcon D, Zahra O, Trejo C,
Olguín-Díaz E and Parra-Vega V
(2019) Computing
Pressure-Deformation Maps for
Braided Continuum Robots.
Front. Robot. AI 6:4.
doi: 10.3389/frobt.2019.00004

Keywords: continuum robots, self-organizing maps, adaptive systems, sensorimotor models, neural networks

1. INTRODUCTION

Rigid robotic manipulators have been thoroughly studied and implemented in several applications for more than five decades now (Nof, 1999). More recently, many roboticists have turned their attention to the design and control of manipulators whose mechanical structure is deformable, and therefore, can achieve multiple shapes. This new type of *continuum robots* have many potential applications in fields such as medical robotics (monitoring procedures with flexible endoscopes), industrial robotics (grasping parts with compliant grippers), bio-inspired robotics (generating natural motions with soft limbs), to name a few cases (Hughes et al., 2016; Laschi et al., 2016). When compared to its rigid robot counterpart, methods for analyzing, sensing, and controlling soft robots are still in its infancy.

Pneumatic power is a common actuation method for continuum robots that brings some useful properties such as inherent compliance, motion backdrivability, controllable expansion/contraction of segments, etc. (Marchese et al., 2015; Sadati et al., 2016). However, the nonlinear and dynamic behavior of pneumatically-driven components makes it difficult to derive a closed-form analytical expression relating the driving air pressures and the highly deformable configuration of the robot. This expression is needed to accurately model and control the deformations of a system. Currently, there is no widely adopted approach for computing such sensorimotor relation.

Previous works have addressed this problem (e.g., Trivedi et al., 2008) presents a detailed nonlinear model relating pressures and the deformations due to pneumatic forces; (Shapiro et al., 2011) derives a closed-form relation between the segment's bending angle (measured on a plane) and the driving pressure; (Falkenhahn et al., 2015) presents a lumped parameters model using the Euler-Lagrange formalism; (Marchese et al., 2016) presents a model for multi-segment soft manipulators and experimentally identifies its parameters; (Sadati et al., 2017) derives a

pressure-deformation model based on the principle of virtual work. Note that computing these models requires precise knowledge of the robot’s mechanical properties, which are hardly available in practice due to the complexity of the system.

The objective of this manuscript is to present a new approach for automatically computing the steady-state relation between pressures and deformations. The method is inspired by models in neuroscience. Its topographically ordered structure resembles the way areas in the brain (e.g., the somatosensory and motor cortex) are organized according to related functions (Kohonen, 2001). This method creates discrete configuration clusters that can be used for predicting deformations and computing controls. In contrast with fixed analytical models, the proposed computational model can continuously adapt its structure to cope for new data or changes in the mechanical conditions. To the best of our knowledge, this is the first time this approach has been used for characterizing pneumatic-driven braided continuum robots.

2. METHODS

2.1. Modeling of Braided Continuum Robots

Consider a cylindrical continuum robot (segment) with no radial and torsional deformations, and with constant curvature along its backbone. The configuration of this system is described by the deformation coordinates (Webster and Jones, 2010):

$$\mathbf{q} = [\lambda \ \kappa \ \psi]^T \in \mathcal{Q} \quad (1)$$

where λ denotes the length of the backbone, κ the curvature of the segment, and ψ the angle of the robot’s bending (see **Figure 1A**). These deformation coordinates are determined by the lengths $\mathbf{l} = [l^1, l^2, l^3]^T$ of the three pneumatic chambers. By using simple trigonometry, the relations between these two vectors can be obtained as follows (Sadati et al., 2017):

$$\mathbf{l} = \begin{bmatrix} \lambda(1 - \kappa r \cos(\psi)) \\ \lambda(1 - \kappa r \cos(120^\circ - \psi)) \\ \lambda(1 - \kappa r \cos(120^\circ + \psi)) \end{bmatrix} \quad (2)$$

where r denotes the backbone-chamber distance. The continuum robot is driven by three independent pressures $\mathbf{p} = [p^1, p^2, p^3]^T \in \mathcal{P}$ that are applied to its inner chambers (see **Figure 1B**). The dynamic equations of this system can be derived using Lagrangian-like methods (see e.g., Falkenhahn et al., 2015; Olguin-Diaz et al., 2018). To control the robot’s shape, it is important to know what input pressures are needed to achieve a desired (final) deformation. This steady-state relation is characterized by the nonlinear mapping $\mathbf{p} = f(\mathbf{q}) : \mathcal{Q} \mapsto \mathcal{P}$, which can be found via the principle of virtual work (Hamill, 2014):

$$\delta \mathbf{l} \cdot \left(\frac{\partial U^b}{\partial \mathbf{l}} + \frac{\partial U^e}{\partial \mathbf{l}} + \frac{\partial U^d}{\partial \mathbf{l}} + \frac{\partial U^p}{\partial \mathbf{l}} \right) = 0 \quad (3)$$

where U^b , U^e , U^d , and U^p denote the potential energies resulting from the body loads, external loads, elastic deformations,

and pneumatic pressures, respectively. Note that the fourth term above yields a function that must be solved for \mathbf{p} (by exploiting Equation 2) to obtain the pressure-deformation relation. However, computing the above energy terms is a difficult task that requires the exact identification of the system.

2.2. Self-Organizing Configuration Maps

The complex properties of pneumatic continuum robots make it difficult to derive an expression relating pressures to deformations. In this work we show how an adaptive computational model can be used to approximate such nonlinear relation.

Consider first that the robot performs a series of babbling-like motions (Saegusa et al., 2009) around the workspace of interest described by $\mathcal{P} \times \mathcal{Q}$. Let us assume that a set of T sampling points $\mathbf{p}(\tau)$ and $\mathbf{q}(\tau)$ are collected at the time instant τ and grouped into the following *training* data vector:

$$\mathbf{x}^k = \begin{bmatrix} \mathbf{p}(\tau) \\ \mathbf{q}(\tau) \end{bmatrix} \in \mathcal{P} \times \mathcal{Q} \quad \text{for } k = 1, \dots, T \quad (4)$$

The vectors Equation(4) will be used for training a self-organizing map (SOM) (Kohonen, 2013). An important property of these maps is that they reduce the dimension of the input space into a 2D lattice while preserving its topology. Neighboring neurons represent configurations \mathbf{x}^k that have “similar” pressure-deformation values.

The network has N computing units arranged in a 2D lattice. Each neuron has an associated weight vector \mathbf{w}^j of the same dimension as \mathbf{x}^k . Training is done by sequentially presenting each input pattern \mathbf{x}^k to the network and finding the weight vector that best matches its values. This “winning” neuron satisfies:

$$i = \arg \min_j \|\mathbf{w}^j - \mathbf{x}^k\| \quad (5)$$

where i denotes its index over the N lattice nodes. The best matching neuron is placed at the center of a neighborhood of cooperating nodes. Let h^{ij} denote the neighborhood function centered at i for an active neuron j . To make the excitation proportional to the distance from the center, a common choice is to use the Gaussian function:

$$h^{ij} = \exp \left(-\frac{\|\mathbf{r}^i - \mathbf{r}^j\|^2}{2\sigma_t^2} \right) \quad (6)$$

where \mathbf{r}^i and \mathbf{r}^j denote the i th and j th node’s 2D position within the lattice (e.g., $\mathbf{r}^i = [20, 15]$). The variable $\sigma_t > 0$ specifies the effective cooperation width, i.e., the influence that neuron i exerts over the nearby neurons. The j th weight vector is computed with the following update rule (Sakamoto et al., 2004):

$$\mathbf{w}_{t+1}^j = \mathbf{w}_t^j - \gamma_t h^{ij} (\mathbf{w}_t^j - \mathbf{x}^k) \quad (7)$$

for a learning gain $\gamma_t > 0$. By using Equation 6, the degree of adaptation of \mathbf{w}_t^j exponentially decreases with its separation from the center.

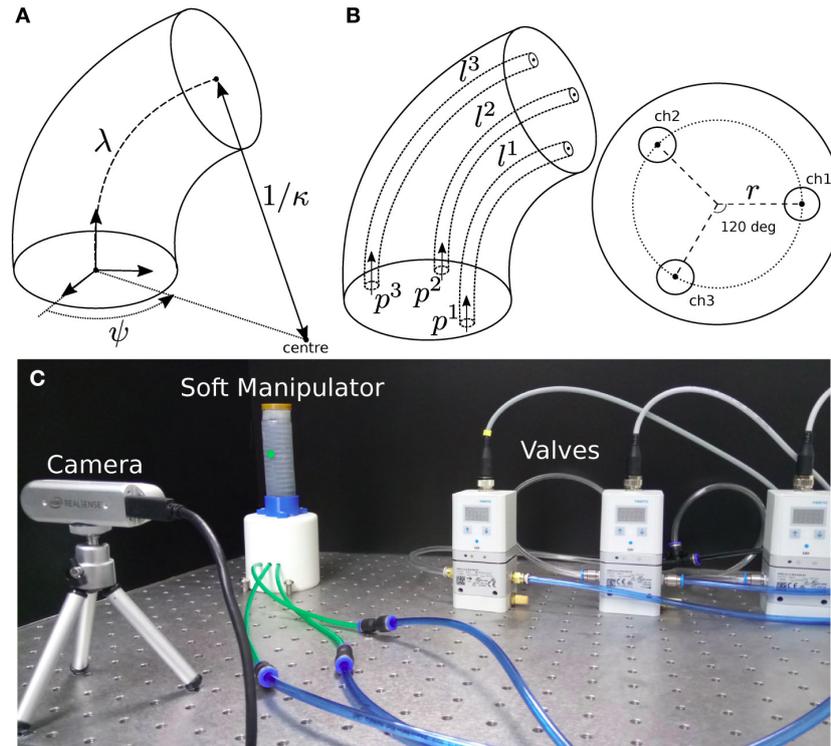


FIGURE 1 | (A) Geometric representation of the robot's deformation coordinates $\mathbf{q} = [\lambda, \kappa, \psi]^T$. **(B)** Details of the internal pneumatic chambers to control the robot's configuration. **(C)** The experimental setup composed of an elastomer-based manipulator, pneumatic valves from Festo (VPPE-3-1-1), a RealSense camera from Intel, and a Linux-based PC with an i7-7700K processor.

The variables σ_t and γ_t control the network's plasticity. These are first given large values to allow for coarse adaptations, and then decreased to fine tune the network's structure as follows:

$$\sigma_{t+1} = \sigma_t - \frac{1}{\eta} (\sigma_t - \tilde{\sigma}) \quad \gamma_{t+1} = \gamma_t - \frac{1}{\eta} (\gamma_t - \tilde{\gamma}) \quad (8)$$

for $\tilde{\sigma}, \tilde{\gamma} > 0$ as the fine tuning parameters, and $\eta > 0$ as its time constant.

3. RESULTS

3.1. Setup

We fabricated an elastomer-based soft manipulator (Agarwal et al., 2016; Schmitt et al., 2018) with three inner chambers that are independently controlled with pneumatic servo-valves (Festo) via an analog board (Phidgets). The robot's configuration is measured with a camera (see **Figure 1C**). All computations are performed in a Linux PC using OpenCV (Bradski, 2000).

In this study, we restrict our attention to the case of planar robot motions¹. To produce plane deformations, we couple the controls $p^2 = p^3$ such that a virtual pressure separated by 180° from p^1 is enforced. To differentiate between left/right bendings,

we use signed curvature values and define a 2D deformation vector $\mathbf{q} = [\lambda, \text{sgn}(\psi)\kappa]^T$, where κ is measured with vision as in Navarro-Alarcon et al. (2014).

3.2. Experiments

The robot first performs a series of *slow* bending/stretching motions (by commanding ramp pressure signals to the valves) from which $T = 734$ data points $\mathbf{x}^k = [p^1, p^{2,3}, \lambda, \text{sgn}(\psi)\kappa]^T$ are collected. This pressure/deformation data set is then used for training the network. The **Supplementary Material Video S1** depicts the conducted experiments.

Note that if the network has too few neurons, the model has problems in separating distinct configurations (e.g., left and right bendings). The U-matrix is a useful method to visualize boundaries between different configurations (Ultsch and Siemon, 1990). It assigns high/low values if its weight vector is different/similar from those of Neighboring nodes. **Figures 2A-C** depict the U-matrices obtained by considering lattices of 10×10 , 30×30 , and 50×50 neurons, respectively. These figures show that boundaries (i.e., red stripes) start to appear with a higher number of nodes². For this study, we have selected a

¹This situation removes 1-DOF from \mathbf{q} as independent changes in ψ cannot be determined by its 2D image measurements.

²The left/right arc symbols are used for representing positive/negative curvatures, respectively, whereas the vertical line symbol is used for central configurations without prominent bendings.

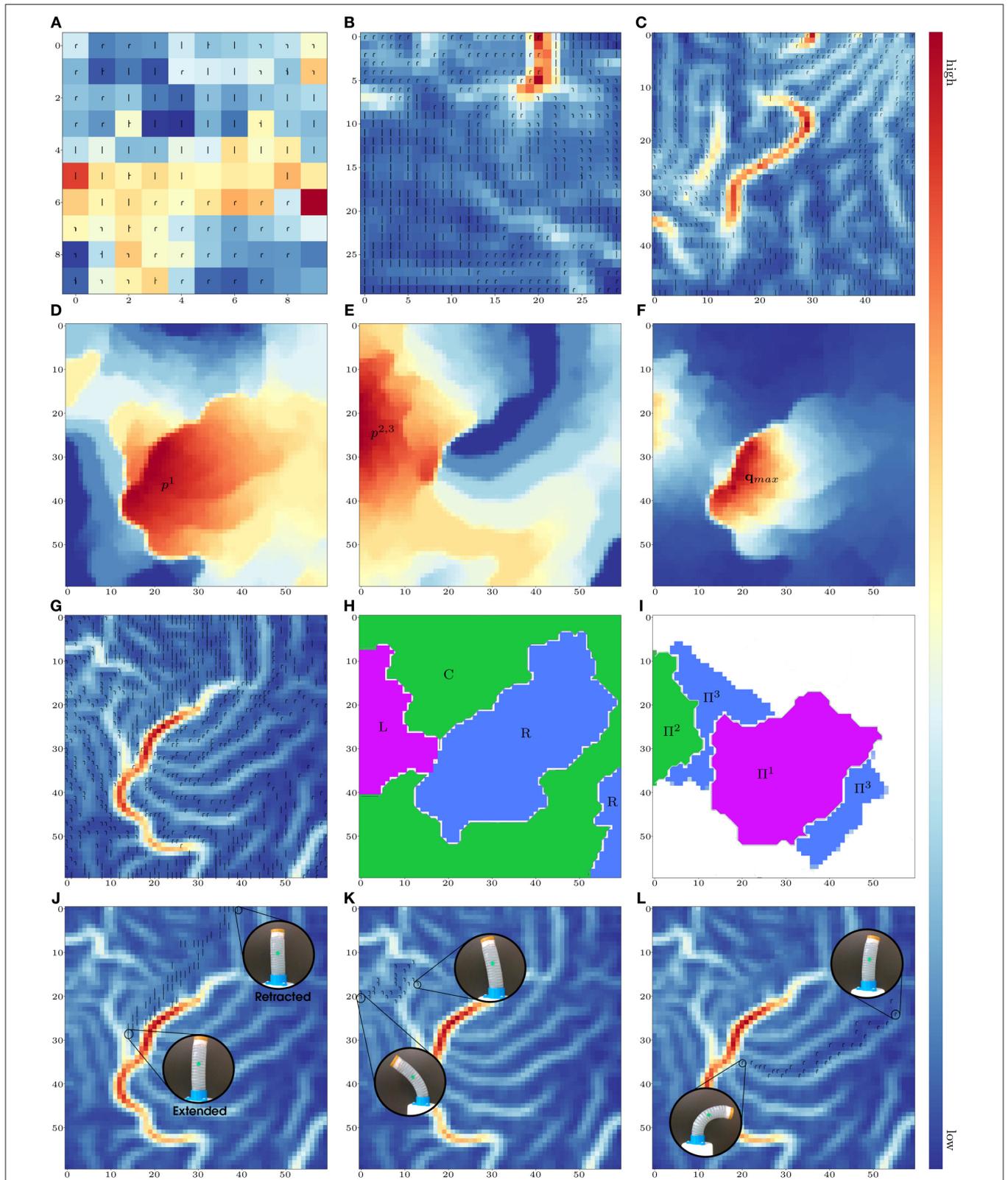


FIGURE 2 | U-matrices with: (A) 10 × 10, (B) 30 × 30, and (C) 50 × 50 neurons. Magnitude of: (D) w^{1j} (activated with p^1), (E) w^{2j} (activated with $p^{2,3}$), and (F) $[w^{3j}, w^{4j}]$ (activated with q). (G) U-matrix with 60 × 60 neurons. (H) Deformation clusters. (I) Pressure clusters. Curves for (J) extension, (K) left bending, and (L) right bending.

network with 60×60 neurons, which will be used for the results reported here.

Figures 2D,E depict the areas that are activated with the input pressures p^1 and $p^{2,3}$. High values can be interpreted as deformations that are “mostly” enforced by either p^1 or $p^{2,3}$ (i.e., right bending, and left bending). The map presents little overlapping of pressure regions; this corresponds to extension/retraction motions that require coordinated actions from all inputs. **Figure 2F** shows the group of neurons that encode the maximum value q_{max} for deformations (i.e., the longest and roundest configurations), which for our case corresponds to right bendings. The identified maximum region depends on the configurations available in the data set (in our experiments, left bendings were not so prominent).

Figure 2G shows the U-matrix computed for the 60×60 network with symbols denoting the dominant deformation state associated with each neuron. From this data, we can create clusters to represent the most numerically distinct configurations of the robot, namely, left and right large bendings, and central positions³. The resulting *deformation clusters* are shown in **Figure 2H** (where L, C, and R denote the left, center, and right robot configurations, respectively). We can also create the *pressure clusters* shown in **Figure 2I**, where Π^1 , Π^2 , and Π^3 denote areas of “mostly” p^1 , “mostly” $p^{2,3}$, and combined pressure actions, respectively.

The network can be used to relate a target deformation \bar{q} with its required pressure values \bar{p} . By presenting a partial vector $\bar{x}^T = [*, *, \bar{q}^T]^T$ (for * as unimportant terms) to the network, we can find the weight w^j that best matches \bar{q} , as done in Equation 5. The corresponding pressure values are simply located in the other coordinates of $w^j = [w^{1j}, w^{2j}, *, *]^T \approx [\bar{p}^T, *, *]^T$. Using this *prediction* approach, our trained network showed a maximum coordinate error of (3.57%, 5.38%) for the \bar{q} and of (17.3%, 15.1%) for \bar{p} .

Figures 2J–L depict examples of how the topologically preserving network can capture incremental deformations of the robot into adjacent neurons. These figures show that the end points of the “deformation trajectories” correspond to the high/low values of the performed motion. The computed maps allow the characterization of the robot’s properties, including fabrication errors, unbalanced pneumatic chambers (see the slight bending in **Figure 2J**), or any other variation in the sensorimotor conditions.

4. DISCUSSION

This brief research report presents a computational model that approximates the steady-state pressure-deformation relations

³We used arbitrary threshold values to define these configurations.

(as described by Equation 3) of pneumatic continuum robots. The method is based on a self-organizing network that discretises the configuration space while preserving its topology. This allows us to represent motions with contiguous nodes whose associated weights smoothly change along the trajectory. To evaluate the method’s performance, we conducted experiments with a robotic prototype⁴.

The proposed method can be interpreted as an adaptive lookup table that automatically organizes data according to the similarity of its coordinates. Once the network is trained, each weight vector provides an approximated relation between deformations and pressures. Such relation can be re-trained when necessary (e.g., based on metrics Polzlbauer, 2004), by reinitializing the parameters in Equation 8 or by using dynamic SOMs (Rougier and Boniface, 2011).

There are some limitations with this approach, e.g., the accuracy of the sensorimotor model is directly dependant on the dimension of the network and the representativeness of the training data. Also, note that the computed model does not capture any dynamical properties of the mechanism; it only describes the sensorimotor relations in a static manner (see Sang et al., 2009 for an SOM to represent dynamical systems).

As future work, we plan to evaluate the method with 3D motions of a soft manipulator (using a 3D camera). To enable the use of sensor-based controls, we are currently developing a similar network that associates local Jacobian-like transformations to each node.

AUTHOR CONTRIBUTIONS

DN-A conceived the study, coordinated the project, and drafted the manuscript. OZ developed the adaptive algorithm, processed the data, and created the visualizations. CT fabricated the continuum robot and conducted the experiments. EO-D and VP-V developed the mathematical model and revised the paper.

FUNDING

This work is supported in part by the Research Grants Council (RGC) of Hong Kong under grant PolyU 142039/17E, in part by The Hong Kong Polytechnic University under grant 4-ZZHJ, in part by the CONACYT under grant 2592.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: <https://www.frontiersin.org/articles/10.3389/frobt.2019.00004/full#supplementary-material>

Video S1 | Experiments.

⁴Note that if the data is collected from multiple (possibly heterogeneous) robots, the resulting map will average its properties.

REFERENCES

- Agarwal, G., Besuchet, N., Audergon, B., and Paik, J. (2016). Stretchable materials for robust soft actuators towards assistive wearable devices. *Sci. Rep.* 6:34224. doi: 10.1038/srep34224
- Bradski, G. (2000). The OpenCV Library. *Dobbs J. Softw. Tools* 25, 122–125. Available online at: <http://www.drdoobs.com/open-source/the-opencv-library/184404319>
- Falkenhahn, V., Mahl, T., Hildebrandt, A., Neumann, R., and Sawodny, O. (2015). Dynamic modeling of bellows-actuated continuum robots using the euler-lagrange formalism. *IEEE Trans. Robot.* 31, 1483–1496. doi: 10.1109/TRO.2015.2496826
- Hamill, P. (2014). *A Student's Guide to Lagrangians and Hamiltonians*. Cambridge, UK: Cambridge University Press.
- Hughes, J., Culha, U., Giardina, F., Guenther, F., Rosendo, A., and Iida, F. (2016). Soft manipulators and grippers: a review. *Front. Robot. AI* 3:69. doi: 10.3389/frobot.2016.00069
- Kohonen, T. (2001). *Self-Organizing Maps*. Oxford, UK: Springer Berlin Heidelberg.
- Kohonen, T. (2013). Essentials of the self-organizing map. *Neural Netw.* 37, 52–65. doi: 10.1016/j.neunet.2012.09.018
- Laschi, C., Mazzolai, B., and Cianchetti, M. (2016). Soft robotics: Technologies and systems pushing the boundaries of robot abilities. *Sci. Rob.* 1, 1–11. doi: 10.1126/scirobotics.aah3690
- Marchese, A. D., Katzschmann, R. K., and Rus, D. (2015). A recipe for soft fluidic elastomer robots. *Soft Robot.* 2, 7–25. doi: 10.1089/soro.2014.0022
- Marchese, A. D., Tedrake, R., and Rus, D. (2016). Dynamics and trajectory optimization for a soft spatial fluidic elastomer manipulator. *Int. J. Robot. Res.* 35, 1000–1019. doi: 10.1177/0278364915587926
- Navarro-Alarcon, D., Liu, Y.-H., Romero, J. G., and Li, P. (2014). On the visual deformation servoing of compliant objects: uncalibrated control methods and experiments. *Int. J. Robot. Res.* 33, 1462–1480. doi: 10.1177/0278364914529355
- Nof, S. (1999). *Handbook of Industrial Robotics. Number v. 1 in Electrical and Electronic Engineering*. Wiley.
- Olguin-Diaz, E., Trejo, C., Parra-Vega, V., and Navarro-Alarcon, D. (2018). “On the modelling of soft-robots as quasi-continuum Lagrangian dynamical systems with well-posed input matrix,” in *Proceedings of Workshop on Soft Robotic Modeling and Control, IROS 2018* (Madrid), 1–4.
- Polzbauer, G. (2004). “Survey and comparison of quality measures for self-organizing maps,” in *Proceedings of Fifth Workshop on Data Analysis* (Vysoké Tatry: Elfa Academic Press), 130–136.
- Rougier, N., and Boniface, Y. (2011). Dynamic self-organising map. *Neurocomp.* 74, 1840–1847. doi: 10.1016/j.neucom.2010.06.034
- Sadati, S., Naghibi, S. E., Shiva, A., Noh, Y., Gupta, A., Walker, I. D., et al. (2017). A geometry deformation model for braided continuum manipulators. *Front. Robot. AI Lausanne* 4:22. doi: 10.3389/frobot.2017.00022
- Sadati, S. H., Shiva, A., Ataka, A., Naghibi, S. E., Walker, I. D., Althoefer, K., et al. (2016). “A geometry deformation model for compound continuum manipulators with external loading,” in *IEEE International Conference on Robotics and Automation* (Lausanne), 4957–4962.
- Saegusa, R., Metta, G., Sandini, G., and Sakka, S. (2009). “Active motor babbling for sensorimotor learning,” in *International Conference on Robotics and Biomimetics* (Guilin), 794–799.
- Sakamoto, Y., Kuriyama, S., and Kaneko, T. (2004). “Motion map: image-based retrieval and segmentation of motion data,” in *Proceedings of the ACM Symposium on Computer Animation* eds R. Boulic, D. K. Pai (Grenoble), 259–266.
- Sang, H., Gelfand, A., Lennard, C., Hegerl, G., and Hewitson, B. (2009). Interpreting self-organizing maps through space–time data models. *Ann. Appl. Stat.* 2, 1194–1216. doi: 10.1214/08-AOAS17
- Schmitt, F., Piccin, O., Barb, L., and Bayle, B. (2018). Soft robots manufacturing: a review. *Front. Robot. AI* 5:84. doi: 10.3389/frobot.2018.00084
- Shapiro, Y., Wolf, A., and Gabor, K. (2011). Bi-bellows: pneumatic bending actuator. *Sens. Actuators A Phys.* 167, 484–494. doi: 10.1016/j.sna.2011.03.008
- Trivedi, D., Lotfi, A., and Rahn, C. D. (2008). Geometrically exact models for soft robotic manipulators. *IEEE Trans. Robot.* 24, 773–780. doi: 10.1109/TRO.2008.924923
- Ultsch, A., and Siemon, H. P. (1990). “Kohonen’s self organizing feature maps for exploratory data analysis,” in *IEEE International Conference on Neural Networks* (Paris: Kluwer Academic Press), 305–308.
- Webster, R. J., and Jones, B. A. (2010). Design and kinematic modeling of constant curvature continuum robots: a review. *Int. J. Robot. Res.* 29, 1661–1683. doi: 10.1177/0278364910368147

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2019 Navarro-Alarcon, Zahra, Trejo, Olguin-Diaz and Parra-Vega. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.