Machine Learning for Soft Robotic Sensing and Control

Keene Chin, Tess Hellebrekers, and Carmel Majidi

Herein, the progress of machine learning methods in the field of soft robotics, specifically in the applications of sensing and control, is outlined. Data-driven methods such as machine learning are especially suited to systems with governing functions that are unknown, impractical or impossible to represent analytically, or computationally intractable to integrate into real-world solutions. Function approximation with careful formulation of the machine learning architecture enables the encoding of dynamic behavior and nonlinearities, with the added potential to address hysteresis and nonstationary behavior. Supervised learning and reinforcement learning in simulation and on a wide variety of physical robotic systems have shown promising results for the use of empirical data-driven methods as a solution to contemporary soft robotics problems.

1. Introduction

Soft robots belong to a diverse category of machines, the borders of which are quite porous depending on the application and field of interest.[1–4] Soft materials are those with a stiffness below some application-specific threshold—typically 1 MPa in the case of systems that interact with biological tissue or mimic soft biological organisms.[9] The mechanical compliance of these materials can be exploited to obviate or simplify many low-level sensing and control problems, e.g., by stabilizing footfalls in legged locomotion[6] or deforming around object geometry in grasping. This ability to encode a desired capability into the mechanical design of the robot itself is one appealing attribute of soft materials.[7–10] Another distinguishing feature is the increased safety of using them in mobile systems due to their lower hardness, increased damping, and lower mass as compared to metals. Because of these attributes, compliant materials are increasingly attractive options for use in modern robotics applications.

As these materials are continuously deformable, their full state is governed by continuum mechanics rather than rigid-body dynamics.[11,12] This means that they are infinite-degree-of-freedom (DOF) systems, which are difficult to represent in closed form. In addition, their dynamics are often highly nonlinear, resulting in systems that are functionally stochastic with respect to a finite state representation. These properties of soft robots, which simplify low-level control and sensing issues, complicate the high-level planning, control, and state estimation of these systems.

Methods based on rigid-body dynamics serve as the backbone of many traditional robotics solutions to control problems requiring heavy modification and approximation to work for soft systems.[13] Among other factors, the difficulty of representing the high-dimensional soft system dynamics means that the gap between the performance of these methods in theory and in practice is usually especially large. High-fidelity models of soft systems have been successfully used in the control of soft systems, but this requires accurate system identification and expensive run-time computation, limiting the potential applications.[14] For soft sensing, this high dimensionality also poses a fundamental challenge as many physical states can produce the same sensor reading as the higher dimensional data is transduced. In addition, these sensors are typically subject to a greater variation in manufacturing outcomes and shapes, have noticeable degrees of hysteresis, nonstationarity, and other mechanical nonlinearities that may be negligible in their rigid counterparts. Approaches to these sensing, actuation, and control problems based on data-driven machine learning provide a parallel path to the continued analytical modeling work and have produced promising results in recent years.

Machine learning methods involve the empirical approximation of an unknown model of some system, e.g., sensor transduction behavior or robot manipulator dynamics.[15] Although this process commonly involves the use of neural networks for function approximation, the term is usefully applied to many optimization strategies and other data-driven approaches. As applied to its use in engineering robotic systems, a critical axis to consider is the degree to which final system performance is dominated by the training data versus an a priori model.[16]

We consider soft sensors to be devices that transduce from some physical phenomenon, e.g., force, to a signal that encodes that information, and which use deformable materials either for the method of sensing or in the mechanical structure in which the sensing element is embedded. Machine learning is used in decoding the signal to reconstruct an intelligible representation of the original phenomenon. For control of soft robotic systems, we look at systems where the physical structure and/or actuators of the robot itself are significantly soft. In the literature, this is overwhelmingly represented by soft continuum manipulators and end effectors, with a few significant standouts. Machine learning
applied to soft robot control involves either an approximation of the kinematic/dynamic model of the system from data or the direct learning of controllers for a given behavior. In this work, we outline the major trends over the last five years for applying machine learning to applications in soft robotics, including common approaches and pitfalls, and potential extensions.

2. Sensing

There are two common threads found in the literature for applications of machine learning to soft sensors (Figure 1). Although supervised learning techniques are applied in both, they differ in the method of data collection. One approach utilizes sensor data collected in a designed test environment and the other builds upon sensor data collected in a fully integrated robotic system. One reason for these two methods is that soft sensor characterization is largely dependent on the mounting structure and shape. Due to the variety of robotic testbeds and hardware limitations, it is not always possible to accurately measure real-world values in each system implementation. However, not establishing a mapping between real-world values and raw sensor output misses an opportunity for model comparison and results that users can interpret.

2.1. Sensor Characterization

The first technique emphasizes the mapping between raw sensor output and real-world values, such as force in newtons, strain in percentage, or pressure in pascals. This is useful to support novel soft sensor implementations and compare trends against known sensors with similar transduction modes. Setting up controlled experiments with additional verification hardware (e.g., triaxis force sensors) also provides ground truth references to determine the sensor resolution, range, and sensitivity. All these metrics are crucial to engineers and roboticists who are looking to select a sensor to match the desired design requirements for their systems. Controlled dynamic experiments along with a recurrent neural network[17,18] and controlled static experiments paired with feed-forward neural networks have been shown to estimate magnitude (in N) and location of touch (in mm)[19] and object orientation[20] (Figure 2A,B).

2.2. Systems Characterization

The second, more common, technique avoids direct controlled characterization and instead prioritizes higher-level labels, such as successful grasp recognition, slip detection, or object classification. We attribute this largely to the discrepancy in individual sensor characterization and practical use, specifically, the inherent limitation that the soft material sensor response will change with the properties of neighboring structures and mounting equipment. In addition, the real-world values provided by the first method are not necessarily understood by the robotic system. In this case, raw data and real-world calibrated values are equally useful as input to the learning models, and it is desirable to avoid an unnecessary middle step. A handful of systems that use this approach are shown in Figure 2. Temporal data and, therefore, temporal learning methods, have been shown to be effective for identifying grasp success and stability with microfluidic sensors (long short-term memories),[22] as well as classifying gestures with capacitive stretch sensors (convolutional neural network).[23] and object gesture/shape/size/and weight recognition with multimodal optical/pressure inputs.[24] Deep learning is also being used for identifying caging grasps on unknown objects with point cloud data,[25] texture classification with piezoresistive taxels,[21] contact estimation and localization with fiber optic embedded soft pad,[26] object recognition with soft tactile skins and joint forces/torques,[27] and full-body motion estimation.[28] In summary, both approaches are necessary to continue developing both novel sensors with supporting models and practical implementation knowledge.

3. Control

Recent efforts to enable control of soft robots through machine learning have addressed different levels of the control pipeline.
Figure 1. Overview of machine learning strategies common in the field. A) Sensor characterization involves collecting raw sensor data aligned with ground truth in a controlled and monitored environment (e.g., lab setting). This data is then typically provided as input to a neural network to predict further values. B) In contrast, systems characterization collects sensor data in a less controlled environment that mimics their use in the field. In this case, ground truth measurements such as force are more difficult to obtain. Therefore, users often circumvent this by mapping to higher-level classifications, such as grasp success and texture recognition. C) Learning inverse kinematics requires training data composed of matched pairs of robot position and actuator configuration. A function approximator (neural network, etc.) encodes the relationship between sampled data to generalize to arbitrary desired positions. D) Learning forward dynamics collects sets of two sequential positions, as well as the actuator action that caused the transition between them. The time-dependent dynamic behavior is encoded into a neural net, allowing for open-loop following of arbitrary trajectories. E) Learning controllers function by iteratively evaluating the performance of the resultant trajectory for a controller with respect to some cost function and updating parameters to increase that performance via optimization.

Figure 2. Examples of soft robotic systems by A) Hellebrekers et al.\cite{19} and B) Han et al. (Reproduced with permission.\cite{17} Copyright 2018, IEEE.) where individual sensors were independently characterized in carefully monitored environments prior to their application. The bottom row shows systems from C) Baishya and Bäuml (Reproduced with permission.\cite{21} Copyright 2016, IEEE.), D) Zimmer et al.\cite{22} and E) Larson et al. (Reproduced with permission.\cite{23} Copyright 2019, Mary Ann Liebert.) in which the sensors are characterized for their specific tasks and setups.
Table 1. Overview of different parameters for learning control. NN, neural network; GMM, Gaussian mixture model.

| Paper                  | Dynamic       | Model representation                  | Hardware training | Simulation training | Adaptive | Feedback |
|-----------------------|---------------|---------------------------------------|-------------------|---------------------|----------|----------|
| Lee et al.\[32\]      | Ensemble of NNs |                                       | ✓                 | ✓                   | ✓        | ✓        |
| Ansari et al.\[33\]   | Linear function approximator |                                       | ✓                 | ✓                   | ✓        | ✓        |
| Thuruthel et al.\[34\]| NN            |                                       | ✓                 | ✓                   | ✓        | ✓        |
| Trumic et al.\[35\]   | Parameterized analytical model |                                       | ✓                 | ✓                   | ✓        | ✓        |
| Chen and Lau\[36\]    | GMM           |                                       | ✓                 | ✓                   | ✓        | ✓        |
| Thuruthel et al.\[37\]| NN            |                                       | ✓                 | ✓                   | ✓        | ✓        |
| Jiang et al.\[38\]    | Hierarchical ensemble of NNs |                                       | ✓                 | ✓                   | ✓        | ✓        |
| Rolf et al.\[39\]     | Constrained extreme learning machine |                                       | ✓                 | ✓                   | ✓        | ✓        |
| Zhang et al.\[40\]    | NN Q-function  |                                       | ✓                 | ✓                   | ✓        | ✓        |
| Thuruthel et al.\[41\]| Recurrent NN  |                                       | ✓                 | ✓                   | ✓        | ✓        |
| Gillespie et al.\[42\]| Deep NN       |                                       | ✓                 | ✓                   | ✓        | ✓        |
| Bern et al.\[43\]     | Spring network energy simulation |                                       | ✓                 | ✓                   | ✓        | ✓        |
| Zhang et al.\[44\]    | Deep NN       |                                       | ✓                 | ✓                   | ✓        | ✓        |
| Schlagenhau et al.\[45\]| Deep NN |                                       | ✓                 | ✓                   | ✓        | ✓        |
| Della Santina et al.\[46\]| Parameterized analytic model |                                       | ✓                 | ✓                   | ✓        | ✓        |

The two most common tactics are learning the inverse kinematics/statics of soft manipulators\[29\] and learning the forward dynamics model to enable predictive control.\[30\] A less common strategy, which has produced interesting results, is the direct learning of controllers for specific behaviors.\[31\] An overview of these efforts is provided in Table 1. The type of function approximation used to encode the kinematic model, dynamic model, or controller is shown, as well as the origin of the training data. Dynamic models, as defined here, do not rely on the quasistatic assumption and have time dependence. Adaptive models can learn from additional data after the initial training, and feedback models produce closed loop controllers.

3.1. Learning Inverse Kinematics

Supervised learning is the basis of much of the machine learning work in learning the kinematics of soft robots. If the system is treated as quasistatic, i.e., it reaches static equilibrium between control steps, a mapping between actuator configuration and task-space position is achievable in open loop. For redundant systems, an injective mapping can be enforced through local optimization, which results in a bijective mapping within the workspace. This is ideal for supervised learning from labeled pairs of training data. To collect training data from which to learn the kinematics function, a random walk through actuation space on the physical hardware (motor babbling) is a straightforward method and has been used for cable-driven continuum manipulators,\[34\] pneumatics continuum manipulators such as the Bionic Handling Assistant (Figure 3A)\[29\] and honeycomb pneumatics network manipulator (Figure 3B),\[38\] and simulated manipulators.\[32,37\] Data can also be collected from human demonstration, by manually replicating important configurations and recording the associated actuator parameters.\[31\]

Data-driven methods have been used to achieve locomotion with soft systems as well,\[41\] but straightforward supervised learning tends to not scale to the dimensionality and dynamic sensitivity of mobile systems.

The forward kinematics of soft robots are usually highly nonlinear. For example, in the case of a pneumatic continuum arm, the vector of pressures in segment chambers, i.e., the configuration space, maps to the pose of the manipulator tip, i.e., the task space, through an equilibrium function that depends on local material properties and design geometry. This mapping between actuator configuration space and task space can also be contingent on the environment. For example, the curvature response of a segment to a given pressure can vary significantly in the presence of external forces such as gravity or surface traction. A global representation of the Jacobian is difficult to convey analytically, even without concerns of nonstationarity and stochasticity. Inverting any found Jacobian is even more difficult. Because of this, a common technique in the field is approximating the global Jacobian with a function approximator, which is most commonly a neural network,\[29,32,34,37,38\] but can be of other forms, like linear function approximators,\[33\] constrained extreme learning machines,\[29\] and Gaussian mixture models (GMMs).\[316\] It should be noted that finding this mapping can be done for a given position without learning any model by performing gradient descent over the actuator configuration with the desired position as the target.\[416\] The downside of this method is that it induces undefined motion iteration in the robot between each commanded position, which is potentially unsafe and results in much slower execution. The global inverse Jacobian can also be approximated by aggregating an ensemble of learned local Jacobians sampled from across the full state space.\[32,36\] Explicit global representations are more concise, while ensemble methods allow prioritization of performance in the regions of most interest.

By virtue of their mechanical compliance, the equilibrium state of soft systems is dependent on external forces and tractions. Replacing the purely kinematic mapping between the
robot configuration and task spaces with a function that incorporates the force equilibrium has been shown to be effective for position tracking of continuum manipulators.\cite{29,37} Even without explicitly learning the force balance, the combination of end effector feedback with this compliance can be used for disturbance rejection in simulation.\cite{34} Due to this decoupling of kinematic shape and external force balance, joint position–stiffness controllers can be learned for soft manipulators.\cite{33,35}

### 3.2. Learning Forward Dynamics

Quasistatic approximations of robotic systems are useful for systems with slow time-scales or tasks where timing is not critical. However, for many applications, dynamic control is more appropriate. The forward dynamics of soft systems are even more analytically difficult than the kinematics, so machine learning techniques have been applied to a variety of these problems. Incorporating dynamics into position control has been shown to enable time-sensitive trajectory tracking in a hybrid soft–rigid arm,\cite{35} and a 1 DOF pneumatic finger (Figure 3C).\cite{40} Dynamics models can be learned that are adapted to varying dynamic parameters of the system, e.g., with stiffness-tuning soft manipulators.\cite{44} Exploiting system dynamics to achieve performance beyond quasistatic predictions is a benefit of learning the dynamic system behavior and has been shown to increase manipulator workspace for the soft system of a continuum arm.\cite{160} The work of Thuruthel et al.\cite{45} on using learned dynamic models exemplifies a significant step in the evolution of soft manipulators enabled by machine learning methods. Earlier examples of soft continuum manipulators such as the work of Rolf et al.\cite{29} were more mechanically sophisticated than the system used here, but the full mechanical complexity was not leveraged due to the lack of a useful dynamics model. Thuruthel et al. demonstrate that an approximate dynamics model can be learned and is sufficient for achieving performance beyond what is possible with quasistatic methods.

### 3.3. Direct Controller Learning

Learning kinematic or dynamic models for a soft robot means that, while part of the control pipeline relies on empirically learnt models, the controller itself is still engineered. Reinforcement learning is a machine learning strategy that allows the controller itself to be created through learning from sequential environmental interactions, rather than from previously collected exogenous data. Policy gradient based reinforcement learning converges to a locally optimal controller without an analytical model of the robot dynamics, but requires much more time and data for training than supervised learning. This is due to the need to evaluate the full trajectory produced by following a controller from a specific state before making updates to the model at a given optimization step. A common robotics solution is learning in simulation for many trials, as for the complex tendon-driven humanoid hand in Figure 3E.\cite{31} Trajectory optimization is a common numerical optimization method for generating local controllers, which can be implemented in a machine learning context. Machine learning–based trajectory optimization has been used to generate a library of trajectories for a mobile tensegrity robot (Figure 3F).\cite{42}
and a forward walking trajectory for a cable-driven soft quadruped (Figure 3G). Zhang et al.\textsuperscript{[41]} show that a global controller can be learned by using the outputs of local trajectory stabilizing controllers as labels for supervised learning. The success of local methods lies in the ability to simplify the space needed to explore for data-driven methods. The humanoid soft hand used by Schlagenhauf et al.\textsuperscript{[42]} is much more mechanically expressive than the previous works using continuum soft manipulators. A global dynamic or kinematic model for this system would require an unfeasible amount of data for physical collection. By creating a basis out of a small number of finite poses, the manipulator is capable of performing dexterous motion without a full model of system behavior. Leveraging the ability to create useful behavior without global models will enable an evolution in the complexity of potential soft robotic systems and the performance of existing systems.

### 3.4. Hybrid Approaches

Systems do not need to be purely data driven for machine learning to be helpful. The dichotomy between empirical, data-driven knowledge and understanding drawn from first principles is not an exclusive one. Rather, the strengths of each strategy can reinforce the other. Hybrid approaches allow the leveraging of existing knowledge so only the most intractable system components need to be learned. Learning the parameters of an analytical dynamics model, similarly to traditional adaptive control methods, has been shown to be fast and effective if such a model can be constructed with enough fidelity.\textsuperscript{[135]} It is also possible to decompose control of multiactuator systems into analytic kinematic targets, where each actuator achieves the final shape through a system-level controller\textsuperscript{[29]} or individual actuator-level controllers.\textsuperscript{[35]} Model predictive control strategies can be used on systems with uncertain models by using neural networks to encode approximate models of system dynamics.\textsuperscript{[40]}

### 4. Upcoming Challenges

The use of machine learning to enable soft robotic functionality has produced many promising results, but there remain several challenges to overcome and unknowns to explore, both from properly training general machine learning models and handling the spatial and temporal nonlinearities of soft systems. Many machine learning techniques that have produced good results in rigid-body robot systems have yet to be fully explored in the soft robot domain. Domain randomization involves augmenting simulated training data by perturbing the simulator’s estimates of the physical parameters of the real task. This strategy combined with a deep neural network controller trained with reinforcement learning has allowed a humanoid robotic hand to solve Rubik’s cubes without specialized hardware or encoded human knowledge.\textsuperscript{[37]} The stochastic and nonlinear dynamics of soft systems is a major difficulty in their control. Similarly difficult dynamics have been learned using machine learning, including a low-cost manipulator with many of the same compliance and nonstationarity concerns of soft systems,\textsuperscript{[48]} as well as an industrial manipulator using a knife to cut deformable vegetables,\textsuperscript{[49]} which can be considered a dual problem to controlling a deformable robotic system in a rigid environment.

#### 4.1. Hysteresis and Nonstationarity

There are unique considerations for soft systems that hint at the potential for unique strategies that come primarily from the synergy between soft systems and data-driven methods. Hysteresis and nonstationarity are inherent side effects of the use of elastomers and other soft materials in the construction of soft robot systems. Current methods often ignore these effects or treat them as unmodeled noise.\textsuperscript{[29, 50]} Hysteresis can be addressed by conditioning learned models on system state history. This explicit time dependence can be encoded in structures such as recurrent neural networks\textsuperscript{[30]} or by simply concatenating multiple time-steps of state data as the input to the model. These types of solutions are not wholly incompatible with quasistatic approaches but lend themselves to more dynamic conceptualizations of the system behavior. Similarly, nonstationary behavior, e.g., baseline drift of raw sensor data due to polymer degradation, induces a need for models that can adapt; otherwise retraining/recalibration is periodically required.\textsuperscript{[49]}

#### 4.2. Model Bias and Overfitting

Using machine learning to enable the sensing and control of soft robots can lead to unpredictable effects if the limitations of machine learning strategies are not explicitly addressed. Model bias and overfitting to the training set are fundamental considerations in the machine learning literature. Model bias results from attempting to fit the training data to an a priori model that does not adequately represent the true behaviour of the system dynamics, and is one of the motivations for using model-free machine learning methods. Given an architecture capable of encoding the dataset information, learned models can match the distribution of training data closely, as machine learning excels at uncovering unknown structure. However, these data-driven methods can encode artifacts of the training data rather than the underlying general features, leading to overfitting to the training data, i.e., loss of generality. Even if the underlying features are learned appropriately over the space of the training set distribution, extrapolations beyond the range of that distribution can produce unreliable results.

#### 4.3. Validation and Reproducibility

Due to the approximate nature of machine learning models, a priori analytical verification of model performance is not generally possible. Specific model validation techniques for machine learning have been developed and should be implemented when using these methods. Tools such as dropout, k-fold cross-validation, and input perturbation can prevent models from overfitting to the training data and allow greater generalization. Metrics for evaluating the training efficacy of learned models can be used to allow better comparison between systems. Rather than reporting the performance metrics of the final learned model in isolation, the relationship between training set performance and testing set performance should be tracked as the
model is trained. A high training error after the learned model converges (underfitting) indicates the presence of model bias. A low training error and a high testing error means the model has overfitted to the training data. In these cases, either training for less time or using a model that matches the complexity of the underlying governing equations can be a solution. Cross-study reproducibility is another important component in the scientific study of these robotic systems. When learned models are used as part of the system, conveying a study’s experimental parameters does not only consist of describing the physical system and the type of machine learning technique used. The training data itself is an integral part of the system’s eventual behavior, and collaboration and validation would be greatly enhanced by the sharing of such data between researchers.

4.4. Increasing System Complexity

Continuum robots and other systems that rely on the constant curvature assumption still dominate the literature[51] due to their simplicity of analysis. However, when the kinematic and dynamic behavior is being learned empirically, an analytically simple system and an analytically intractable system do not necessarily require different approaches to learn. Complex functions that are not easily represented analytically can be encoded by training function approximators with sampled data from physical systems. Given the ability of machine learning models to discover the underlying structure in the data without a full a priori model, the physical system that produces the data need not be analytically feasible to model. Because of this, riskier and more esoteric robot morphologies can be explored, leveraging the rich diversity of design that has been generated within the soft robotics literature. An expansion in capabilities of control and sensing could also enable open-source, user-fabricated platforms with limited standardization such as those proposed by Della Santina et al[52] to proliferate. More complex systems that feature multifunctional materials, computationally optimized geometry,[53] or an intrinsic interplay between structure and control[54] (morphological computation) are all potentially quite difficult to model analytically but have properties that are nonetheless highly enabling and advance the field of soft robotics. Using machine learning to aid in sensing and control of these systems is a promising synergy that can allow the novel capabilities of soft materials and novel soft robotics hardware to impact the space of what is possible with intelligent robotic systems.

Acknowledgements

This work was in part supported by the National Science Foundation (NSF) Graduate Research Fellowship Program (GRFP) under Grant No. DGE 1252522 and the National Oceanographic Partnership Program (NOPP) under Grant No. N000141812843 (PM: Dr. Reginald Beach). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF or NOPP.

Conflict of Interest

The authors declare no conflict of interest.

Keywords

control, machine learning, neural networks, sensing, soft robotics

Received: December 10, 2019
Revised: January 10, 2020
Published online: March 3, 2020

[1] D. Trivedi, C. D. Rahn, W. M. Kier, I. D. Walker, Appl. Bionics Biomech. 2008, 5, 99.
[2] G. M. Whitesides, Angew. Chem. Int. Ed. 2018, 57, 4258.
[3] C. Majidi, Adv. Mater. Technol. 2019, 4, 1800477.
[4] D. Rus, M. T. Tolley, Nature 2015, 521, 467.
[5] C. Majidi, Soft Robot. 2014, 1, 5.
[6] G. Buondonno, J. Carpentier, G. Saurel, N. Mansard, A. De Luca, J.-P. Laumond, in IEEE/RJS Int. Conf. on Intelligent Robots and Systems, IEEE, Piscataway, NJ 2017, p. 705.
[7] K. Man, A. Damasio, Nat. Mach. Intell. 2019, 1, 446.
[8] D. Howard, A. E. Eiben, D. F. Kennedy, P. Valencia, D. Winkler, Nat. Mach. Intell. 2019, 1, 12.
[9] K. Nakajima, H. Hauser, T. Li, R. Pfeifer, Soft Robot. 2018, 5, 339.
[10] S. Haddadin, K. Krieger, A. Albu-Schaffer, T. Lilge, IEEE Trans. Robot. 2018, 34, 91.
[11] K. M. de Payrebrune, O. M. O’Reilly, Extreme Mech. Lett. 2016, 8, 38.
[12] E. Ceevoet, T. Morales-Biezze, F. Largilliere, Z. Zhang, M. Thieffry, M. Sanz-Lopez, B. Carrez, D. Marchal, O. Goury, J. Desquidt, C. Duriez, Adv. Robot. 2017, 31, 1208.
[13] D. Ross, M. P. Nemitz, A. A. Stokes, Soft Robot. 2016, 3, 170.
[14] C. Duriez, in Proc. IEEE Int. Conf. on Robotics and Automation, IEEE, Piscataway, NJ 2013, p. 3982.
[15] T. M. Mitchell, Machine Learning, McGraw-Hill Education, New York 1997.
[16] C. Bishop, Pattern Recognition and Machine Learning, 1st ed., Springer, New York 2006.
[17] S. Han, T. Kim, D. Kim, Y. L. Park, S. Jo, IEEE Robot. Automat. Lett. 2018, 3, 873.
[18] T. G. Thurfjelld, B. Shih, C. Laschi, M. T. Tolley, Sci. Robot. 2019, 4, eaav1488.
[19] T. Hellebrekers, O. Kroemer, C. Majidi, Adv. Intell. Syst. 2019, 1, 1900025.
[20] M. H. Rosle, R. Kojima, K. Or, Z. Wang, S. Hirai, IEEE Robot. Automat. Lett. 2019, 5, 159.
[21] S. S. Baishya, B. Bäuml, in IEEE Int. Conf. on Intelligent Robots and Systems, IEEE, Piscataway, NJ 2016, p. 8.
[22] J. Zimmer, T. Hellebrekers, T. Asfour, C. Majidi, O. Kroemer, Predicting Grasp Success with a Soft Sensing Skin and Shape-Memory Actuated Gripper, IEEE, Piscataway, NJ 2019, pp. 7120–7127.
[23] C. Larson, J. Sjøt, R. Knepper, R. Shepherd, Soft Robot. 2019, 6, 611.
[24] H. Huang, J. Lin, L. Wu, B. Fang, Z. Wen, F. Sun, Machine Learning-Based Multi-Modal Information Perception for Soft Robotic Hands, Science Robotics, Washington, DC.
[25] C. Choi, W. Schwarting, J. Delpreto, D. Rus, IEEE Robot. Automat. Lett. 2018, 3, 2370.
[26] J. I. Kim, D. Kim, M. Krebs, Y. S. Park, Y.-L. Park, IEEE Robot. Automat. Lett. 2019, 4, 3481.
[27] A. Schmitz, Y. Bansho, K. Noda, H. Iwata, T. Ogata, S. Sugano, in IEEE-RAS Int. Conf. on Humanoid Robots, IEEE, Piscataway, NJ 2015, p. 1044.
[28] D. Kim, J. Kwon, S. Han, Y. L. Park, S. Jo, IEEE/ASME Trans. Mechatron. 2019, 24, 56.
[29] M. Rolf, K. Neumann, J. F. Queißer, R. F. Reinhart, A. Nordmann, J. J. Steil, Adv. Robot. 2015, 29, 847.
[30] T. G. Thuruthel, E. Falotico, F. Renda, C. Laschi, *Bioinspiration and Biomimetics* 2017, 12, 066003.

[31] C. Schlagenhauf, D. Bauer, K.-H. Chang, J. P. King, D. Moro, S. Coros, N. Pollard, *Control of Tendon-Driven Soft Foam Robot Hands*, Tech. Rep., IEEE, Piscataway, NJ.

[32] K. H. Lee, D. K. Fu, M. C. Leong, M. Chow, H. C. Fu, K. Althoefer, K. Y. Sze, C. K. Yeung, K. W. Kwok, *Soft Robot*. 2017, 4, 324.

[33] Y. Ansari, M. Manti, E. Falotico, M. Cianchetti, C. Laschi, *IEEE Robot. Automat. Lett.* 2018, 3, 108.

[34] T. G. Thuruthel, E. Falotico, M. Manti, A. Pratesi, M. Cianchetti, C. Laschi, *Soft Robot*. 2017, 4, 285.

[35] M. Trumic, K. Jovanovic, A. Fagiolini, *Kernel-Based Nonlinear Adaptive Control of Stiffness and Position for Soft Robots Actuators*, Tech. Rep., IEEE, Piscataway, NJ.

[36] J. Chen, H. Y. Lau, in *Proc. of the 2nd Int. Conf. on Control, Automation and Robotics*, IEEE, Piscataway, NJ 2016, p. 103.

[37] T. G. Thuruthel, E. Falotico, M. Cianchetti, F. Renda, C. Laschi, in 13th Int. Conf. on Informatics in Control, Automation and Robotics, Doctoral Consortium, Vol. 2. Science and Technology Publications, Setubal, Portugal 2016, p. 303.

[38] H. Jiang, Z. Wang, X. Liu, X. Chen, Y. Jin, X. You, X. Chen, in *Proc. IEEE Int. Conf. on Robotics and Automation*. IEEE, Piscataway, NJ 2017, p. 6127.

[39] H. Zhang, R. Cao, S. Zilberstein, F. Wu, X. Chen, in *Proc. 10th International Conference on Intelligent Robotics and Applications (ICIRA)*, Vol. 10462. Springer, Cham, Switzerland 2017, pp. 173–184.

[40] M. T. Gillespie, C. M. Best, E. C. Townsend, D. Wingate, M. D. Killpack, in *IEEE Int. Conf. on Soft Robotics*, RoboSoft 2018, 2018, p. 39.

[41] J. M. Bern, P. Banzet, R. Poranne, S. Coros, in *Proc. Robotic Science and Systems*, Robotics: Science and Systems 2019.

[42] M. Zhang, X. Geng, J. Bruce, K. Caluwaerts, M. Vespignani, V. Sunspiral, P. Abbeel, S. Levine, in *Proc. IEEE Int. Conf. on Robotics and Automation*, IEEE, Piscataway, NJ 2017, p. 634.

[43] C. Della Santina, M. Bianchi, G. Grioli, F. Angelini, M. Catalano, M. Garabini, A. Bicchi, *IEEE Robot. Automat. Mag.* 2017, 24, 75.

[44] R. F. Reinhart, J. J. Steil, *Proc. Technol.* 2016, 26, 12.

[45] T. G. Thuruthel, E. Falotico, M. Manti, C. Laschi, *IEEE Robot. Automat. Lett.* 2018, 3, 1292.

[46] R. K. Katzschmann, A. D. Marchese, D. Rus, *Soft Robot*. 2015, 2, 155.

[47] OpenAI, I. Akkaya, M. Andrychowicz, M. Chociej, M. Litwin, B. McGrew, A. Petron, A. Paino, M. Plappert, G. Powell, R. Ribas, J. Schneider, N. Tezak, J. Tworek, P. Welinder, L. Weng, Q. Yuan, W. Zaremba, L. Zhang, *Solving Rubik’s Cube with a Robot Hand*. 2019.

[48] M. P. Deisenroth, C. E. Rasmussen, D. Fox, *Robot. Sci. Syst.* 2011, 5, 57.

[49] I. Lenz, R. A. Knepper, A. Saxena, *Robot. Sci. Syst.* 2015, 11.

[50] C. Lee, M. Kim, Y. J. Kim, N. Hong, S. Ryu, H. J. Kim, S. Kim, *Int. J. Control Automat. Syst.* 2017, 15, 3.

[51] T. George Thuruthel, Y. Ansari, E. Falotico, C. Laschi, *Soft Robot*. 2018, 5, 149.

[52] C. Della Santina, C. Piazza, G. M. Gasparri, M. Bonilla, M. G. Catalano, G. Grioli, M. Garabini, A. Bicchi, *IEEE Robot. Automat. Mag.* 2017, 24, 48.

[53] N. Cheney, R. MacCurdy, J. Clune, H. Lipson, *ACM SIGEVOlution* 2014, 7, 11.

[54] N. Cheney, J. Bongard, V. SunSpiral, H. Lipson, *J. R. Soc. Interface* 2018, 15, 20170937.