To grasp or not to grasp: An end-to-end deep-learning approach for predicting grasping failures in soft hands

To grasp or not to grasp: an end-to-end deep-learning approach for predicting grasping failures in soft hands

Visar Arapi[1], Yujie Zhang[1,2], Giuseppe Averta[1,2], Manuel G. Catalano[1,3], Daniela Rus[4], Cosimo Della Santina[4], and Matteo Bianchi[1,2]

***Abstract*— This paper tackles the challenge of predicting grasp failures in soft hands before they happen, by combining deep learning with a sensing strategy based on distributed Inertial Measurement Units. We propose two neural architectures, which we implemented and tested with an articulated soft hand - the Pisa/IIT SoftHand - and a continuously deformable soft hand - the RBO Hand. The first architecture (Classifier) implements a-posteriori detection of the failure event, serving as a test-bench to assess the possibility of extracting failure information from the discussed input signals. This network reaches up to 100% of accuracy within our experimental validation. Motivated by these results, we introduce a second architecture (Predictor), which is the main contribution of the paper. This network works on-line and takes as input a multi-dimensional continuum stream of raw signals coming from the Inertial Measurement Units. The network is trained to predict the occurrence in the near future of a failure event. The Predictor detects 100% of failures with both hands, with the detection happening on average 1.96 seconds before the actual failing occurs - leaving plenty of time to an hypothetical controller to react.**

I. Introduction

The use of compliant and soft elements in robotic hands has proven to be a formidable tool for endowing them with previously unmatched capabilities [1], [2], [3]. At the same time, this new technology has initiated the grand challenge of developing algorithms able to deal with, and even exploit, the intelligence embodied in the hands by the purposeful introduction of these elastic components.

Under a control point of view, the controller should let the hand itself (i.e. the embodied mechanical intelligence) performing the larger part of the grasping/manipulation task. Indeed, the physical adaptability of these systems allows to overcome local uncertainties, requiring only an approximated estimation of the relative hand-object pose. Accordingly, most of the grasp strategies for soft hands proposed so far resort to implementing correct wrist placement and hand pre-shaping [4], [5], [6]. Furthermore, the controller should purposefully react if any unexpected situation occurs. For example, the hand could fail in grasping the object, in which case the controller should be informed in due time, so that a recover routine can be triggered [7]. This makes detecting failures a central goal for developing effective control architectures for soft hands.

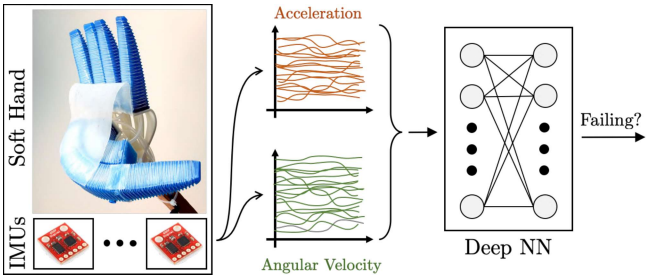


Fig. 1. In this paper two deep learning based architectures are proposed for classifying and predicting failure events in soft robotic hands. Three IMUs are placed on each finger of the hands, for a total of 15 units. The raw readings coming from on-board accelerometers and gyroscopes - 135 signals - are directly fed into the networks.

A commonly used strategy for grasp failure detection - often employed with rigid hands - is to directly measure contact forces with dedicated sensors [8]. With this kind of information, the a posteriori detection of a failure becomes trivial [9]. Slippages can also be directly sensed when tangential forces are measured. More sophisticated usages of these sensors include the prediction of future failure event when the hand is in static conditions [10], and the prediction of slippage even before it starts [11]. These types of techniques use machine learning to perform the prediction. In [12] a similar strategy is adopted within the context of articulated soft hands (i.e. hands with rigid structure but highly deformable joints, also called compliant hands). The failing is here predicted by directly measuring tangential forces through sensors placed at the fingertips. This is however a strategy that is hardly feasible in the generic soft case - especially for what concerns continuum soft hands (i.e. hands made of continuously deformable materials). Indeed, obtaining effective force sensing within the softness constraints is exceptionally challenging [13], [14].

A promising alternative to force sensing is to measure other quantities either tactile or related to other sensory sources - e.g. hand posture [15], audio signals [16], video streams [17] - and infer contact forces through algorithms. Model based techniques are hard to use in these cases, due to the well-known difficulties in formulating reliable models of the hand-object interaction when softness is involved [18]. The use of machine learning to extract force information have therefore been investigated, with promising outcomes [17], [19], [20]. However, we are not aware of any application of these approaches to failure detection or prediction.

This paper moves from the following consideration; *since in any case advanced failure prediction involves the use*

This work has been partially supported by the EU H2020 Research and Innovation program under grant agreement no. 688857 (SoftPro), no. 732737 (Iliad) and no. 871237 (Sophia) and by the Italian Ministry of Education and Research (MIUR) in the framework of the CrossLab project (Department of Excellence).

[1]Centro di Ricerca "Enrico Piaggio", Universita' di Pisa, Largo Lucio Lazzarino 1, 56126 Pisa, Italy {matteobianchi23,visararapi22,manuel.catalano}@gmail.com

[2]Dipartimento di Ingegneria dell'Informazione, Universita' di Pisa, Largo Lucio Lazzarino 1, 56126 Pisa, Italy giuseppe.averta@ing.unipi.it

[3]Soft Robotics for Human Cooperation and Rehabilitation, Fondazione Istituto Italiano di Tecnologia, via Morego, 30, 16163 Genova, Italy

[4]Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA, USA {dsantina,rus}@csail.mit.edu

of machine learning strategies - even when force sensors are available - why do not directly learning an end-to-end mapping from the raw sensors to the detection of the failure event?

To the best of authors' knowledge, the only related approach going in this direction is introduced in [21], which however proposes failure characterization, rather than prediction. Moreover, the strategy is tailored on articulated soft hands only, and experimentally tested with a 2-fingered articulated soft gripper grasping a cylinder.

We deal instead with the challenge of predicting a failure event with a generic - either continuum or articulated - soft hand. Furthermore, the emphasis is on exploiting temporal features to act on-line and before the event happens, rather than in static conditions. We tackle this challenge through deep learning. First, we propose a deep neural architecture discerning failed grasps from successful ones after the grasp is completed. This serves as ground truth and motivation for our second neural architecture, which is built by stacking three convolutional layers (CNN) and two Long Short Memory Networks (LSTM), and it is able to reliably predict failure events several seconds before they actually happen. The sensing strategy relies on a set of distributed Inertial Measurement Units (IMUs), reading both accelerations and angular velocities. The proposed sensing system and machine learning architecture have been tested with two soft robotic hands: the Pisa/IIT SoftHand [22] and the RBO Hand [23]. The first is an articulated soft hand, the second is continuously deformable. To conclude, this paper contributes with

- the use of IMUs as a reliable source of contact information in soft hands,
- a deep neural network capable of a-posteriori detection of failure events in soft hands,
- a deep neural network capable of on-line prediction of failure events in soft hands,
- the validation of the method with two soft robotic hands.

II. Problem definition

A. Sensing apparatus

We propose here to use a set of Inertial Measurement Units (IMUs) to acquire information on soft hands behavior. A total of 16 IMUs is used here.We place three IMUs for each finger. To get local information, measurements are referred to a IMU placed on the back of the hand palm.

The IMUs are rigidly connected to a deformable glove, which can be easily added to any soft hand[1] without spoiling its softness. Fig. 2 shows the two hands considered in this work, with and without the sensing glove. The first hand is an articulated soft hand - the Pisa/IIT SoftHand [22] - with 19 flexible joints and an under-actuated mechanism implementing a single degree of actuation. The second hand is a continuum soft hand - the RBO Hand [23]. Fully made of silicon rubber, it can undergo continuum deformations. The actuation is pneumatic. From each of these units we read accelerations and angular velocities expressed along the three local axes. We have thus a total of 96 signals. These signals are peculiarly information-rich, carrying both low frequency and high frequency content. The first is related to the change in posture of the hand, and the latter to (micro-)impacts and (micro-)slippage events.

[1]Note that while we consider here anthropomorphic soft hands, the proposed strategy is general in its formulation and thus seamlessly applicable to any soft gripper.

Let us assume a soft robotic hand performing a grasping action. Let us also consider this hand sensorized as discussed above. Sensors measures are collected in the vector $\mathbf{X_t} = [\mathbf{x}_{1,t}^{\mathrm{T}}, \mathbf{x}_{2,t}^{\mathrm{T}}, \ldots, \mathbf{x}_{n,t}^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{nm}$, where $\mathbf{x}_{i,t} \in \mathbb{R}^m$ is the measurement of the i-th sensor at time frame $t \in \mathbb{N}$. m is the dimensionality of the measured quantities and n is the total number of sensors. In the apparatus discussed above these values are $m = 6$ and $n = 16$. Given a set of available measurements at times frames $\{1 \ldots k\}$, we can collect them into the matrix $\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_k] \in \mathbb{R}^{nm \times k}$.

B. Regression problems

We analyze failure detection from two - strictly connected - points of view. First, we tackle the easier case in which the goal is to classify whether one execution resulted in a successful or a failed grasp. The whole evolution of $\mathbf{X_t}$ is considered available as input to the network. As already discussed in the introduction, this would be a trivial problem if contact sensors were available. However, the nature of the sensor input considered here, and their complex and hard-to-formulate relationship with the event we intend to detect, makes this goal already quite challenging. More specifically, we describe this goal as approximating the function

$$\mathbf{y} = \mathscr{C}(\mathbf{X}), \quad \mathscr{P} : \mathbb{R}^{nm \times k} \rightarrow \{0,1\} \tag{1}$$

where $\mathbf{y} \in \{0,1\}$, is a boolean variable that classifies the whole sequence $\mathbf{X}$ between two possible conditions: *success* ($\mathbf{y} = 0$) or *failure* ($\mathbf{y} = 1$).

Second, we consider a more challenging problem - whose solution is the main contribution of this paper - in which the goal is to predict at each time $\bar{t}$ the success or failing of a grasp given the current sensor readings $\mathbf{X}_{\bar{t}}$. More specifically

$$\mathbf{Y} = \mathscr{P}(\mathbf{X}), \quad \mathscr{P} : \mathbb{R}^{nm \times k} \rightarrow \{0,1\}^m \tag{2}$$

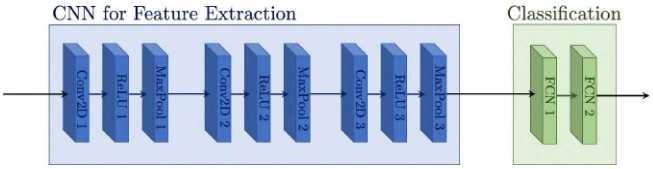
where $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_m]^{\mathrm{T}}$ is the boolean vector collecting the elements $\mathbf{y}_{\bar{t}} \in \{0,1\}$. This value is equal to 1 if the hand-object motions happening at $\bar{t}$ will end into a failure event. $\mathbf{y}_{\bar{t}}$ is 0 otherwise. Therefore, $\mathbf{Y}$ has always the structure of a sequence of zeros, possibly followed by a single sequence of ones - which starts from the time in which the hand-object configuration leading to the failure appears, till the end of the experiment.

Consider here a given ordered set $\mathbf{X}^*$ of measurements $\mathbf{X}$ coming from a number of experiments. We hypothesize the knowledge of $\mathbf{y}^* = \mathscr{C}(\mathbf{X})^*$, $\mathbf{Y}^* = \mathscr{P}(\mathbf{X}^*)$. The pairs $\{\mathbf{X}^*, \mathbf{y}^*\}$ and $\{\mathbf{X}^*, \mathbf{Y}^*\}$ will be referred as training set in the following. This knowledge is provided by an expert labeller, who visually inspects video material of the experiments. Our goal is to learn both $\mathscr{C}$ and $\mathscr{P}$ models by minimizing the difference between the predicted labels $\hat{\mathbf{y}}$, $\hat{\mathbf{Y}}$ and the ground truths $\mathbf{y}^*$, $\mathbf{Y}^*$.

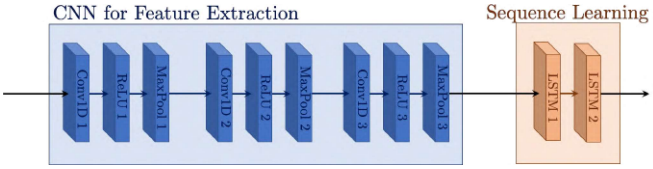
It is very important to underline that while the labeling is performed in a non-causal way - i.e. by looking if the motions at time t will produce a failure in the future - we ask $\mathscr{P}(\mathbf{X})$ to be causal, i.e. that the $\bar{t}-$th component $\mathbf{y}_{\bar{t}} = \mathscr{P}_{\bar{t}}(\mathbf{X})$ is actually only function of $\{\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_{\bar{t}}\}$. This is equivalent to introducing the hypothesis that the current readings carry enough information on the hand state to allow for a prediction

(a) P/I SoftHand, no sens. (b) RBO Hand, no sensors (c) P/I SoftHand, with sens. (d) RBO Hand, with sensors

Fig. 2. Panels (a,b) show the two robotic hands considered in this study, without sensors. The first is an articulated soft hand [22], the second a continuum soft hand [23]. Panels (c,d) show the same hands when the proposed sensing apparatus is used.



(a) Classifier



(b) Predictor

Fig. 3. The two deep neural networks proposed in this paper. Panel (a) shows the Classifier. Its inputs are entire sequences $\mathbf{X} \in \mathbb{R}^{nm \times k}$ and its output a single boolean value $\{0,1\}$. Panel (b) shows the Predictor. This network approximates (2) by breaking down $\mathbf{X} \in \mathbb{R}^{nm \times k}$ into its samples $\mathbf{X}_t \in \mathbb{R}^{nm}$, and it uses them as inputs. To any sequence of inputs $\{1, \ldots, \mathbf{X}_{\bar{t}}\}$ corresponds a single output in $\mathbf{y}_{\bar{t}} \in \{0,1\}$. Note therefore that the first stage - CNN for Feature Extraction - has only apparently the same structure in the two cases. Indeed, the one in Panel (a) is a CNN for 2-D feature extraction, while the architecture in Panel (b) is a CNN for 1-D feature extraction.

of the failure event, based on an internal representation of hand-object physics.

III. Grasp failure detection via Deep Neural Networks

Finding $\mathscr{C}$ and $\mathscr{P}$ can be regarded as pattern recognition (PR) problems. We consider here the use of Deep Learning (DL) to achieve this goal. Compared to the other traditional PR approaches - see e.g. decision trees [24], hidden Markov models [25], support vector machines [26]) - DL techniques can learn features automatically from data, thus being more appropriate for our highly unstructured framework. Furthermore, DL techniques can extract on-line high-level feature details in deep layers.

In this work we propose two deep architectures as depicted in (Fig.3), named respectively "*Classifier*" - approximating (1) - and "*Predictor*" - approximating (2).

We will describe these architectures already referring to their optimal structures - in terms of hyperparameters - which we learned from data. We will provide details on this learning process in Sec. V.

TABLE I
Structure of the CNN used in the classifier.

type	# of kernels (b)	kernel size (f)	stride (s)
Conv 1	64	11	4
MaxPool 1	-	3	2
Conv 2	96	5	1
MaxPool 2	-	3	2
Conv 3	96	5	1
MaxPool 3	-	3	2

A. Classifier

We attack the problem through a Convolutional Neural Network [27] (CNN). Note that input $\mathbf{X}$ - being a matrix - can be regarded as analogous to a 2D image. The input is processed by the CNN, which acts as a feature extractor. Note that the convolution in time is essentially related to Fourier transforms. So through this technique we can at the same time isolate spatial [2] relationship between signals, and - even more importantly - separate high frequency and low frequency information (see Sec. II).

The first stage of the CNN comprises three sequences of convolution, rectified linear units, and pooling. Convolution layers - 'Conv2D i" - extract features from the input data by means of a convolution operation of the input. Number of kernels b, of dimension f, and stride s of each layer if reported in Tab. I. . Each Convolution layer is followed by a standard Rectified linear units (ReLU) - saturating the outcomes of convolutional layers - and a standard pooling layer - down-sampling the inputs. Finally two fully connected layers (FCN 1 and FCN 2) are added, each one including 512 neurons. In this way, features describing specific time intervals and/or local events within the soft hand can be combined to achieve global features - in this way reasoning on the whole hand, in the whole time period. Finally, a softmax function - not shown in figure - capitalizes on these fixed size representations to generate a probability distribution over $\{0,1\}$.

B. Predictor

This network cannot extract temporal features by means of convolutions, since they are non causal operator. Therefore, we split spatial and temporal features extraction into two

[2]Note that $\mathbf{X}$ has been defined so that signals coming from a same IMU are adjacent, and signals coming from close IMUs are close.

TABLE II
STRUCTURE OF THE CNN USED IN THE PREDICTOR.

type	# of kernels (b)	kernel size (f)	stride (s)
Conv1D 1	64	3	1
MaxPool1D 1	-	2	1
Conv1D 2	64	3	1
MaxPool1D 2	-	2	1
Conv1D 3	64	3	1
MaxPool1D 3	-	2	1

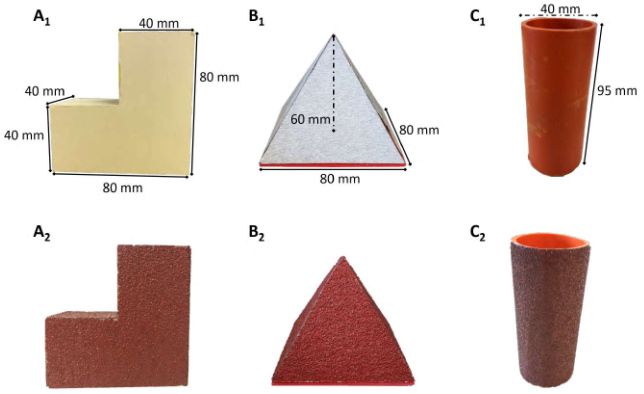


Fig. 4. *Eigen*-Objects considered for the experiments presented in this paper. Three different fundamental shapes and two values of surface roughness are considered (smooth on the top images, rough in the bottom images).

separate stages. The latter assumes also the role that was of the fully connected layers in the Classifier, i.e. to correlate features which are physically and temporally far from each others.

This results in an architecture consisting of two fundamental ingredients: (i) a Convolutional Neural Network , and (ii) Recurrent Neural Networks, of the Long Short Time Memory (LSTM) type [28]. The input of the CNN is 1D, being the vector $\mathbf{X}_t$. Details on the CNN architecture are reported in Tab. II. Input channels are processed by three Conv1D layers, each of them followed by ReLU and MaxPool layers as illustrated in Fig.3. The CNN outputs are connected to the *Sequence Learning* module, implemented by 2 LSTM layers including 128 memory cells each.

IV. EXPERIMENTS AND DATA COLLECTION

With the goal of acquiring a dataset to be used for training and validation, we carried out a series of experiments in which an expert user maneuvered the soft hands through a handle.

Fig. 5 shows the experimental setup. A camera is used to record the scene and to keep track of the time. The hand is equipped with the IMUs, which are used to record the accelerations and the gyroscope values of the hand fingers. During the experiments, a custom routine was used to record synchronously the IMUs signals and the video.

We executed the grasp of three different objects: a cylinder (H = 95mm, R = 40mm), a pyramid (H = 60mm, L = 80mm) and a L-shape polyhedron (L_1 = 40mm, L_2 = 80mm). Each object was presented two times, first covered with a rough surface (high roughness object), and then with a smooth

Fig. 5. Experimental Setup used in this paper. Two soft robotic hands (namely RBO Hand and Pisa/IIT SoftHand) were used to grasp six objects with different shape and surface roughness. A camera was used to record the whole experimental execution. Accelerations of the fingers were recorded through a custom glove endowed with 16 IMUs.

surface (low roughness object) - see Fig. 4 for details. The idea is to provide the network with examples of prototypical shapes and roughnesses. More realistic objects - see below - can then be seen as a combination of these fundamental ingredients. For this reason we refer to these objects as *eigen*-objects, in analogy to *eigen*-vectors of linear spaces. For the sake of space, we cannot provide in this paper a more formal analysis of this idea, which will provided in future work.

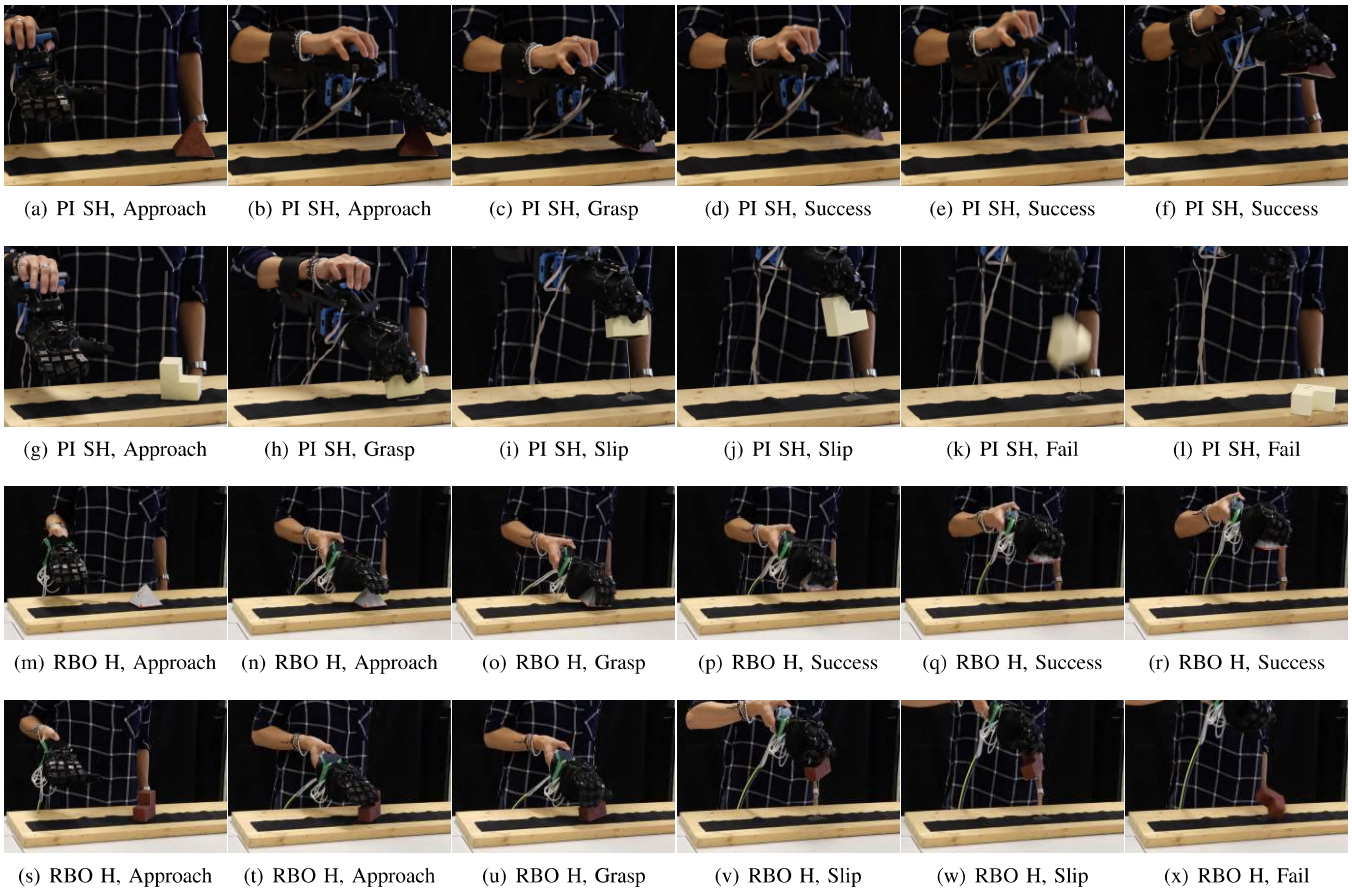
Each acquisition was repeated five times, in two different experimental conditions: i) object free to move; ii) object fastened to the surface through a fixed-length wire (see Fig. 4). This resulted in the generation - in a controlled and repeatable fashion - of successful grasps in case i) and failed grasps in case ii). It is worth noticing that, given the non-zero length of the wire, the initial part of the acquisition for case ii) is expected to be analogous, under a phenomenological point of view, to the acquisitions of case i), while differences are - by experimental design - arising only when the action of the wire is effectively introducing an external wrench on the object (please refer also to Fig. 6).

The same experiments were performed with the Pisa/IIT SoftHand, and the RBO Hand. Note that, the strong design differences between the two models are a key point for the development of this work and also an important point of strength of the overall proposed architecture. Indeed, these differences have an effect on the distributed vibrations along the fingers and, hence, introduce non trivial differences in the recorded signals.

Finally, experiment are repeated when grasping the objects in Fig. 7. These are objects of common day use, and data extracted from them will be used to test the capabilities of the network to generalize in complex ways. These experiments were performed using the RBO hand.

V. RESULTS

The two proposed architectures are tested on three datasets described in the previous section: one containing data from Pisa/IIT SoftHand experiments only, one containing data from RBO Hand only, and a mixed dataset obtained by merging the first two. More specifically, we use both successful and failure grasp datasets in the case of the grasp execution classification problem, while in the grasp execution prediction we employ



(a) PI SH, Approach (b) PI SH, Approach (c) PI SH, Grasp (d) PI SH, Success (e) PI SH, Success (f) PI SH, Success

(g) PI SH, Approach (h) PI SH, Grasp (i) PI SH, Slip (j) PI SH, Slip (k) PI SH, Fail (l) PI SH, Fail

(m) RBO H, Approach (n) RBO H, Approach (o) RBO H, Grasp (p) RBO H, Success (q) RBO H, Success (r) RBO H, Success

(s) RBO H, Approach (t) RBO H, Approach (u) RBO H, Grasp (v) RBO H, Slip (w) RBO H, Slip (x) RBO H, Fail

Fig. 6. Stills from four examples of experiments. Panels (a-f) show the Pisa/IIT SoftHand SH successfully grasping a rough pyramid object. Panels (g-l) show the Pisa/IIT SoftHand failing in grasping a smooth L-shaped object. Panels (m-r) show the RBO Hand successfully grasping a smooth pyramidal object. Panels (s-x) show the RBO hand failing to grasp a rough L-shaped object.

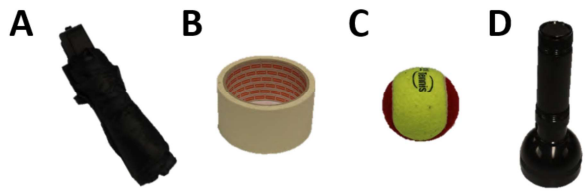


Fig. 7. Objects used for the evaluation of the proposed network, outside the *eigen*-objects set. Panel (a) shows an umbrella, Panel (b) a tape, Panel (c) a ball, Panel (d) a torch.

only failure grasp datasets. This choice has been carried out to avoid to polarize the network to success examples. Indeed, failure examples already incorporate a certain amount of success information (the failure starts happening only when the tendon starts producing external wrenches). However, rather than discarded, all the success examples are collected in a second validation set, on which we will test the network after training.

Both architectures are implemented using *keras library* and are trained from scratch. The library is written in python, and we adopt *Tensorflow* as back-end. Training and testing procedures are executed on a NVIDIA GTX1080 with 8GB of on board memory.

We use hold out cross-validation to ensure the generalization and robustness of both the networks. The goal is to estimate the expected level of model predictive accuracy independently from the data used to train the model. During training process we reduce over-fitting risk applying *dropout* technique in both networks. Thereby, each neuron composing the last "FCN 2" layer (or each memory cell composing "LSTM 2" layer) is disconnected with probability p_{drop}. This action enhances network variability and minimizes weights co-adaption.

1) Classifier: The three datasets we consider in this case are composed by: 100 acquisitions for both Pisa/IIT Softhand and RBO hand and 200 acquisitions for the mixed dataset. We randomly split the three datasets discussed above in: 80% for training and 20% for testing. We trained 40 different network configurations to discover both network hyper-parameters (i.e., CNN depth and width) and learning hyper-parameters (i.e., batch size, learning rate, number of epochs and dropout). Each configuration was obtained by varying number of convolutional layers in $\{2,3,4\}$, number of kernels for each conv layer in $\{32,64,96,128\}$, number of FCN layers in $\{1,2,3\}$, number of neurons for each FCN layer in $\{256,512,1024\}$, batch size in $\{5,10,15,20\}$, learning rate in $\{10^{-2},10^{-3},10^{-4}\}$, number of epochs in $\{10,20,30,40\}$ and dropout $p_{\text{drop}} \in \{0.4,0.5,0.6\}$.

Among all the architectures resulting from training, we select the one that provides the highest accuracy on the mixed

TABLE III

CONFUSION MATRICES RELATIVE TO GRASP CLASSIFICATION. FROM LEFT TO RIGHT: PISA/IIT SOFTHAND, RBO HAND, AND MIXED DATASETS.

		Predicted	
		Success	Fail
True	Success	93	7
	Fail	7	93

		Predicted	
		Success	Fail
True	Success	100	0
	Fail	0	100

		Predicted	
		Success	Fail
True	Success	97	3
	Fail	7	93

test dataset. This network configuration has been exhaustively described in Sec. V. The selected hyper-parameters are: batch size 10, Adam optimizer with learning rate 10^{-3}, number of epochs 30 and dropout 0.5. With such learning parameters, the network classifies the two classes (*successful* and *failure*) with an accuracy of 93% in the Pisa/IIT SoftHand test dataset, 100% in the RBO Hand test dataset and 95% in the mixed test dataset, as also reported in Tab. III.

2) Predictor: The dataset that we consider here are composed of: 50 acquisitions for both Pisa/IIT Softhand and RBO hand, and 100 acquisitions for the mixed dataset. Here as well the three datasets are randomly divided in 80% for training and 20% for testing. We train 40 different network configurations, obtained by varying the hyper-parameters, within the same intervals discussed for the Classifier. Looking at the results of each simulation, we select the configuration that provides the highest accuracy on the test mixed dataset. The network configuration has been described by Sec. III. The selected learning hyper-parameters are: batch size 8, RMSprop optimizer with learning rate 10^{-4}, number of epochs 30 and dropout $p_{\text{drop}} = 0.5$. In this configuration the network predicts the two classes (no failure and ongoing failure) with an accuracy of 85.8% in the Pisa/IIT Softhand test dataset, 94.2% in the RBO Hand test dataset and 90.6% in the mixed test dataset, as also reported in Tab. IV. Despite being trained only on failure examples, the network is remarkably good in not producing false positives. Indeed the accuracy tested on success examples is 83.3%. Fig. 8 shows the results expressed in time for the mixed dataset, showing that the algorithm is able to predict the failure event 100% of times, with an average anticipation of 1.96 seconds. Similar results can be obtained for the other validations sets. Two examples - one success and one failure - of measured signals, together with predicted and ground truth labels, are shown in Fig. 10, for the RBO hand and Fig. 9.

Finally the network is tested on failure prediction within the objects of common use shown in Fig. 7. This is done to test the ability of the network to generalize to less standardized objects, with more complex surface characteristics. This provides a first validation of the above-introduced idea of learning fundamental features from *eigen*-objects. This learned feature extraction capability is then exploited while dealing with unseen objects that can be considered intuitively as combinations of these basic components. Note that objects (a) and (c) have a partially deformable nature, while (b) and (d) are rigid. The results on the whole validation set are provided in Tab. V. Two conditions are tested; network (i) not retrained, and (ii) retrained exclusively using experiments from objects (a) and (b). Also in this case the failure is predicted in 100% of cases, with the network identifying the failure several seconds before it happened. Two examples of measured signals, together with predicted and ground truth labels, are shown in Fig. 11. The first example shows the most common case, in which the two outputs of the retrained and not retrained networks are similar, with the first

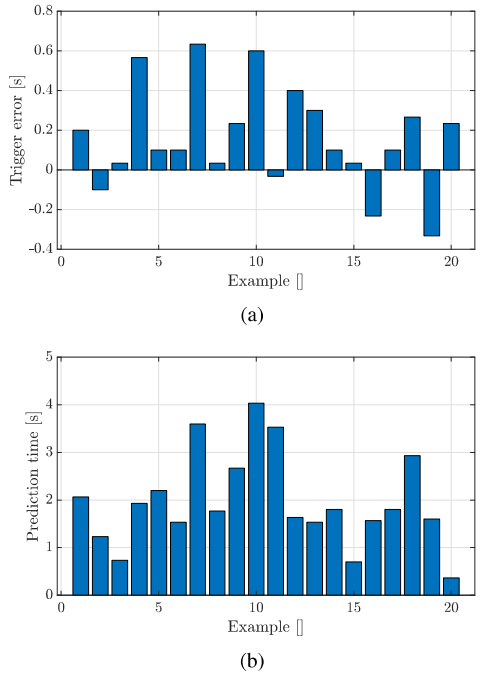


Fig. 8. Performance of the network measured in time for the failures in the mixed validation set. Panel (a) shows the difference between the time in which the change of label happens in the ground truth, and when it occurs in the network's output. Negative values identify and anticipation. Panel (b) shows how much in advance the network predicts the failure. The average value is 1.96 seconds.

presenting only slightly improved performance. The second shows a behavior which can be spotted in few experiments, where the untrained network presents some spikes before the actual failing starts. This behavior always disappears after the training.

VI. CONCLUSIONS AND FUTURE WORK

In this paper we presented a deep learning based framework to predict grasp failures in soft robotic hands. The idea is based on three main pillars: i) the usage of soft hands to exploit the vibrations induced by objects' sliding and transferred to the fingers; ii) an IMU-based sensing setup to record such vibrations; iii) a high level intelligence that predicts the object sliding by looking at IMUs reading. The system has been first tested on a set of six *eigen*-objects, with different size and surface roughness, and with two different hand designs, i.e. a continuum soft hand (RBO Hand) and an articulated soft hand (Pisa/IIT SoftHand). Then, we applied the Predictor on data acquired with the continuum hand, grasping four objects of daily use. Results show promising performances of the overall framework, for both tasks of classification and sliding prediction. In our future work we will integrate this algorithm within the control architecture [6]. Another major direction of investigation will be formalizing the *eigen*-object concept, here only preliminary introduced.

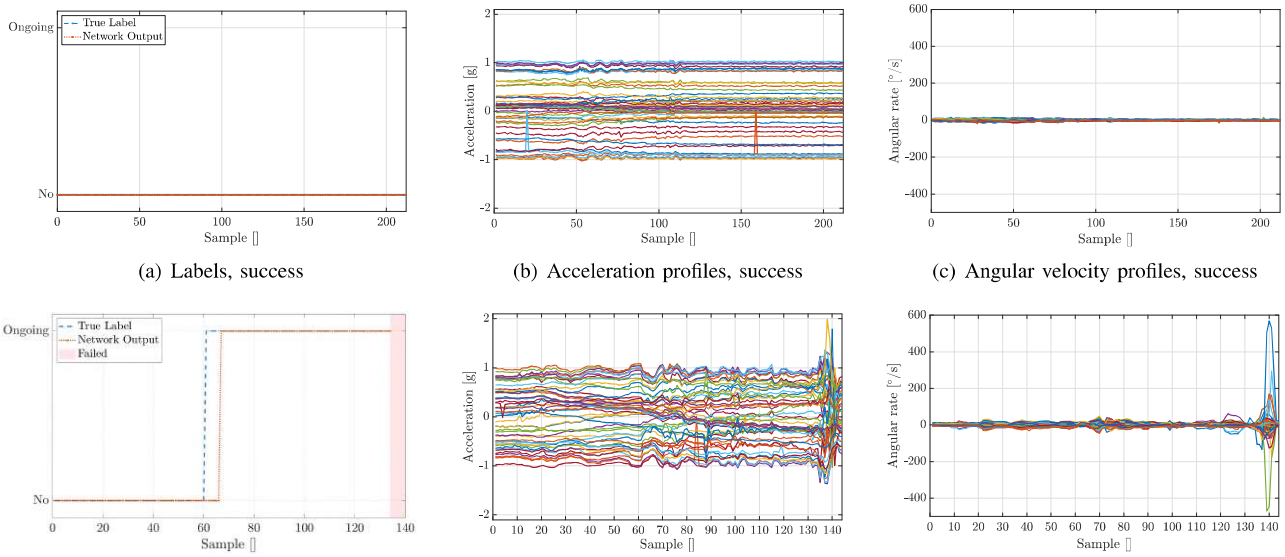
TABLE IV

CONFUSION MATRICES RELATIVE TO OBJECT FAILURE PREDICTION. FROM LEFT TO RIGHT; PISA/IIT SOFTHAND, RBO HAND, AND MIXED DATASETS.

		Predicted	
		No Fail.	Ongoing
True	No Fail.	86.3	13.7
	Ongoing	14.6	85.4

		Predicted	
		No Fail.	Ongoing
True	No Fail.	93.2	6.8
	Ongoing	4.7	95.3

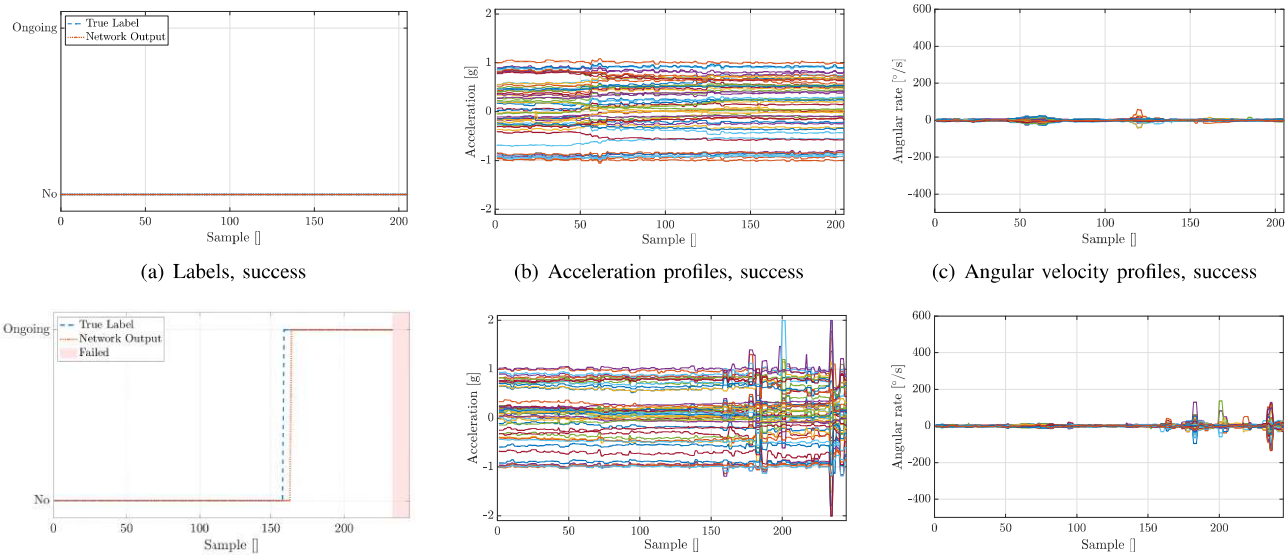
		Predicted	
		No Fail.	Ongoing
True	No Fail.	91.6	8.4
	Ongoing	10.3	89.7



(a) Labels, success (b) Acceleration profiles, success (c) Angular velocity profiles, success

(d) Labels, failure (e) Acceleration profiles, failure (f) Angular velocity profiles, failure

Fig. 9. Labels, classification, and sensor readings for one success experiment - Panels (a-c) respectively - and one failure experiment - Panels (e-f) respectively - with the Pisa/IIT SoftHand. When compared to the success case, clear patterns - even if high dimensional and supposedly non-linear- can be visually recognized in both acceleration and angular velocities while the object slips, and even more clearly when the object falls.



(a) Labels, success (b) Acceleration profiles, success (c) Angular velocity profiles, success

(d) Labels, failure (e) Acceleration profiles, failure (f) Angular velocity profiles, failure

Fig. 10. Labels, classification, and sensor readings for one success experiment - Panels (a-c) respectively - and one failure experiment - Panels (e-f) respectively - with the RBO hand. When compared to the success case, clear patterns - even if high dimensional and supposedly non-linear- can be visually recognized in both acceleration and angular velocities while the object slips, and even more clearly when the object falls.

REFERENCES

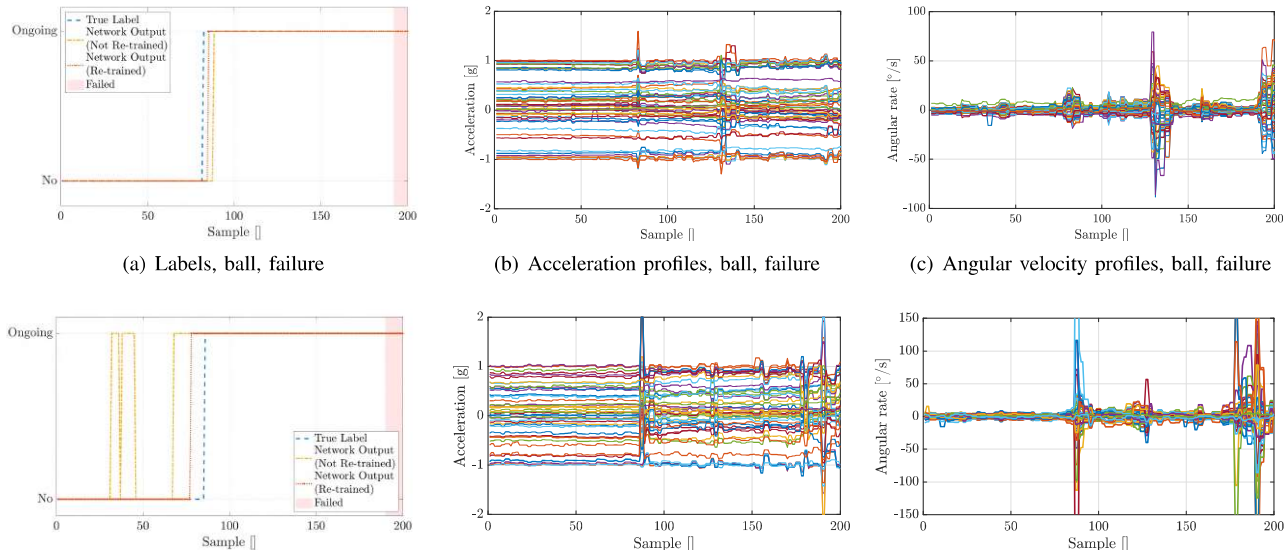
[1] J. Hughes, U. Culha, F. Giardina, F. Guenther, A. Rosendo, and F. Iida, "Soft manipulators and grippers: a review," *Frontiers in Robotics and AI*, vol. 3, p. 69, 2016.

[2] J. Shintake, V. Cacucciolo, D. Floreano, and H. Shea, "Soft robotic grippers," *Advanced Materials*, vol. 30, no. 29, p. 1707035, 2018.

[3] C. Piazza, G. Grioli, M. Catalano, and A. Bicchi, "A century of robotic hands," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. (In Press), 2019.

[4] C. Choi, W. Schwarting, J. DelPreto, and D. Rus, "Learning object grasping for soft robot hands," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2370–2377, 2018.

[5] M. Pozzi, G. Salvietti, J. Bimbo, M. Malvezzi, and D. Prattichizzo, "The closure signature: a functional approach to model underactuated compliant robotic hands," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2206–2213, 2018.



(a) Labels, ball, failure (b) Acceleration profiles, ball, failure (c) Angular velocity profiles, ball, failure

(d) Labels, torch, failure (e) Acceleration profiles, torch, failure (f) Angular velocity profiles, torch, failure

Fig. 11. Labels, classification, and sensor readings with the RBO hand or two of the experiments with the second object sets. Results prior to retraining and after retraining are shown. These example are selected as representative of the two situations that we observe in all the validation set. In Panels (a-c) the result is already good and the retrain only marginally improves it, in Panels (d-f) without retrain the network produces some short time false positive before the actual detection. This behavior always disappears after the retraining.

TABLE V

CONFUSION MATRICES FOR THE SECOND OBJECT SET. WITHOUT (TOP) AND WITH (BOTTOM) RETRAINING.

		Predicted	
		No Fail.	Ongoing
True	No Fail.	71.1	28.9
	Ongoing	24.7	75.3

		No Fail.	Ongoing
True	No Fail.	86.7	13.3
	Ongoing	11.6	88.4

[6] C. Della Santina, V. Arapi, G. Averta, F. Damiani, G. Fiore, A. Settimi, M. G. Catalano, D. Bacciu, A. Bicchi, and M. Bianchi, "Learning from humans how to grasp: a data-driven architecture for autonomous grasping with anthropomorphic soft hands," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1533–1540, 2019.

[7] S. Kitagawa, K. Wada, K. Okada, and M. Inaba, "Learning-based task failure prediction for selective dual-arm manipulation in warehouse stowing," in *International Conference on Intelligent Autonomous Systems*. Springer, 2018, pp. 428–439.

[8] Z. Kappassov, J.-A. Corrales, and V. Perdereau, "Tactile sensing in dexterous robot hands," *Robotics and Autonomous Systems*, vol. 74, pp. 195–220, 2015.

[9] L. Pinto and A. Gupta, "Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours," in *2016 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2016, pp. 3406–3413.

[10] R. Krug, A. J. Lilienthal, D. Kragic, and Y. Bekiroglu, "Analytic grasp success prediction with tactile feedback," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 165–171.

[11] X. Song, H. Liu, K. Althoefer, T. Nanayakkara, and L. D. Seneviratne, "Efficient break-away friction ratio and slip prediction based on haptic surface exploration," *IEEE Transactions on Robotics*, vol. 30, no. 1, pp. 203–219, 2013.

[12] A. Ajoudani, E. Hocaoglu, A. Altobelli, M. Rossi, E. Battaglia, N. Tsagarakis, and A. Bicchi, "Reflex control of the pisa/iit softhand during object slippage," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 1972–1979.

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

[14] 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 *2019 2nd IEEE International Conference on Soft Robotics (RoboSoft)*. IEEE, 2019, pp. 322–329.

[15] C. Della Santina, C. Piazza, G. Santaera, G. Grioli, M. Catalano, and A. Bicchi, "Estimating contact forces from postural measures in a class of under-actuated robotic hands," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 2456–2463.

[16] G. Zöller, V. Wall, and O. Brock, "Acoustic sensing for soft pneumatic actuators," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 6986–6991.

[17] Y. She, S. Q. Liu, P. Yu, and E. Adelson, "Exoskeleton-covered soft finger with vision-based proprioception and exteroception," *arXiv preprint arXiv:1910.01287*, 2019.

[18] F. Ficuciello, A. Migliozzi, E. Coevoet, A. Petit, and C. Duriez, "Fem-based deformation control for dexterous manipulation of 3d soft objects," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 4007–4013.

[19] N. Elangovan, A. Dwivedi, L. Gerez, C.-M. Chang, and M. Liarokapis, "Employing imu and aruco marker based tracking to decode the contact forces exerted by adaptive hands," in *Humanoid Robots (Humanoids), 2019 IEEE-RAS 19th International Conference on*. IEEE, 2019.

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

[21] A. Sintov, A. S. Morgan, A. Kimmel, A. M. Dollar, K. E. Bekris, and A. Boularias, "Learning a state transition model of an underactuated adaptive hand," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1287–1294, 2019.

[22] C. Della Santina, C. Piazza, G. M. Gasparri, M. Bonilla, M. G. Catalano, G. Grioli, M. Garabini, and A. Bicchi, "The quest for natural machine motion: An open platform to fast-prototyping articulated soft robots," *IEEE Robotics & Automation Magazine*, vol. 24, no. 1, pp. 48–56, 2017.

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

[24] S. R. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," *IEEE transactions on systems, man, and cybernetics*, vol. 21, no. 3, pp. 660–674, 1991.

[25] S. R. Eddy, "Profile hidden markov models." *Bioinformatics (Oxford, England)*, vol. 14, no. 9, pp. 755–763, 1998.

[26] J. A. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural processing letters*, vol. 9, no. 3, pp. 293–300, 1999.

[27] Y. LeCun, Y. Bengio *et al.*, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, vol. 3361, no. 10, p. 1995, 1995.

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