# POLITECNICO DI TORINO
## Repository ISTITUZIONALE

Learning to Prevent Grasp Failure with Soft Hands: From Online Prediction to DualArm Grasp Recovery

*Terms of use:*

*Publisher copyright*

(Article begins on next page)

06 May 2024

RESEARCH ARTICLE

ADVANCED
INTELLIGENT
SYSTEMS
Open Access

www.advintellsyst.com

# Learning to Prevent Grasp Failure with Soft Hands: From Online Prediction to Dual-Arm Grasp Recovery

*Giuseppe Averta,\* Federica Barontini, Irene Valdambrini, Paolo Cheli, Davide Bacciu, and Matteo Bianchi*

Soft hands allow to simplify the grasp planning to achieve a successful grasp, thanks to their intrinsic adaptability. At the same time, their usage poses new challenges, related to the adoption of classical sensing techniques originally developed for rigid end defectors, which provide fundamental information, such as to detect object slippage. Under this regard, model-based approaches for the processing of the gathered information are hard to use, due to the difficulties in modeling hand–object interaction when softness is involved. To overcome these limitations, in this article, we proposed to combine distributed tactile sensing and machine learning (recurrent neural network) to detect sliding conditions for a soft robotic hand mounted on a robotic manipulator, targeting the prediction of the grasp failure event and the direction of sliding. The outcomes of these predictions allow for an online triggering of a compensatory action performed with a second robotic arm–hand system, to prevent the failure. Despite the fact that the network is trained only with spherical and cylindrical objects, we demonstrate high generalization capabilities of our framework, achieving a correct prediction of the failure direction in 75% of cases, and a 85% of successful regrasps, for a selection of 12 objects of common use.

G. Averta,[+] F. Barontini, I. Valdambrini, P. Cheli, M. Bianchi
Centro di Ricerca "Enrico Piaggio"
Universita' di Pisa
Largo Lucio Lazzarino 1, Pisa 56126, Italy
E-mail: giuseppe.averta@polito.it, g.averta3@gmail.com

G. Averta, F. Barontini, I. Valdambrini, P. Cheli, M. Bianchi
Dipartimento di Ingegneria dell'Informazione
Universita' di Pisa
Largo Lucio Lazzarino 1, Pisa 56126, Italy

D. Bacciu
Dipartimento di Informatica
Universita di Pisa
Pisa 56127, Italy

The ORCID identification number(s) for the author(s) of this article can be found under https://doi.org/10.1002/aisy.202100146.

[+]Present address: Politecnico di Torino, 10129, Italy

## 1. Introduction

In recent years, the introduction of soft elements in robotic hands demonstrated to be an asset to easily provide capabilities never seen with rigid components.[1,2] The intelligence, directly embedded into the mechanics, enables to fold the fingers around the object in a natural fashion, and to gently adapt the shape of the hand when interacting with the environment. This characteristic comes with the additional benefit that potential uncertainties in local relative placement between the end-effector and the object are compensated by the compliance of the hand, thus relaxing constraints in robot planning.[3–6]

However, this increased dexterity is also responsible for a reduced amount of information that the regulator may feedback to close a control loop. Indeed, because of the difficulties in defining accurate models of hands,[7] of the hand–object interaction when softness is involved,[8] and to the intrinsic uncertainties that elastic components produce in the measurements,[9] it is in general not straightforward to use sensing techniques originally developed for rigid end effectors (e.g., rigid force sensors at the fingertips, encoder[10]), and to implement model-based feedback solutions that can react to unexpected situations.[11] Indeed, although the increased performances in terms of grasp success that characterize the usage of soft grippers, objects picking and grasping may still fail in many cases. In those events, it is important to have a system able to predict, within a reasonable time horizon, when a grasped object is going to slide w.r.t. the hand, and would probably fall, and eventually trigger a suitable corrective action (**Figure 1**).

To solve this problem, one of the most common approaches relies on the direct measure of contact forces, usually relying on force/torque sensors at the joints or at the fingertip level.[12–16] However, these approaches are typically hardly feasible in practice, given the large cost of the hardware and the complexity of the sensing setup—which introduces significant computational effort—and are not suitable in general for continuum soft hands, where the shape and the mechanical response of the fingertip may be significantly different than rigid or articulated soft hands.

**ADVANCED**
**SCIENCE NEWS**
www.advancedsciencenews.com

**ADVANCED**
**INTELLIGENT**
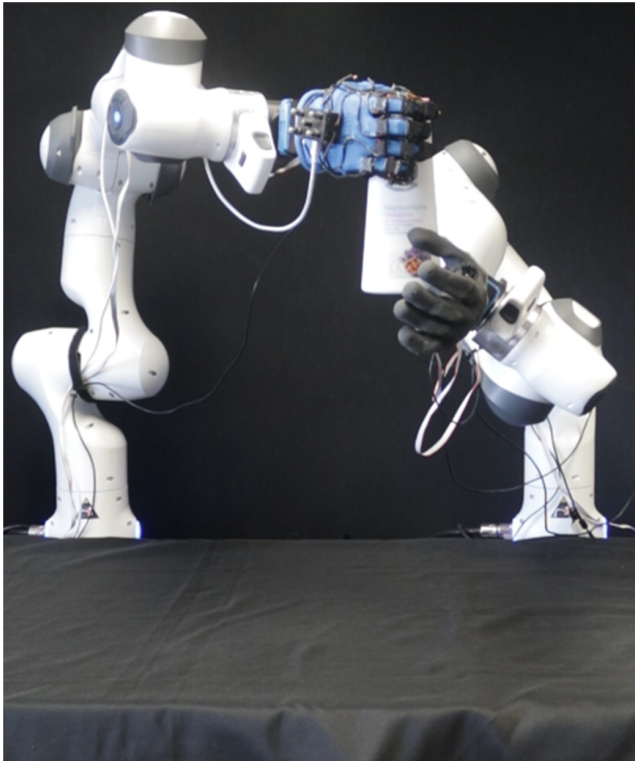**SYSTEMS**
Open Access
www.advintellsyst.com

**Figure 1.** A Franka Emika Panda robotic arm integrated with a soft hand equipped with IMU sensors is used to reach and grasp a generic object. A sliding event is detected by processing the IMU information with a RNN, triggering a reactive regrasping primitive with a second arm–hand system, to firmly hold the object.

As an alternative to force sensing, the community is recently exploring the usage of other sensory sources, such as audio signals,[17] inertial sensing,[18] video streams,[19] and tactile sensors,[20,21] and infer contact forces through algorithms. However, little has been done so far to exploit such sensory information to predict when a grasp is going to fail, and to trigger reactive recovery primitives.

Recently, we proposed to exploit inertial sensing (accelerations and angular velocities) to feed a deep neural network which was able to accurately classify offline if the stream of data were associated to a grasp failure, and even predict online its occurrence.[22] More specifically, in the study by Arapi et al.,[22] we demonstrated that inertial measurement units (IMUs)—placed at the fingers level—are capable to record the vibrations caused by the sliding of grasped objects, and a deep architecture, trained to detect the occurrence of these conditions, can be used to predict when a grasp is going to fail.

In this article, we build upon our preliminary work and further extend our deep learning framework for grasp failure prediction. More specifically, however, in previous experiments, failures were caused by a rope which mechanically constrained the maximum distance between the grasped object and the table, resulting in an abrupt and nonecological failure condition, in this work, we completely redesigned the experimental part, generating failures as a consequence of a variable weight added to the

object. Furthermore, a robotic arm was used to execute the reach-and-grasp task, in both success and failure cases, thus removing potential artifacts introduced by the manual handling of the robotic hand as done in the study by Arapi et al.[22] Another significant contribution of this work with respect to the study by Arapi et al.[22] is that we now target not only the prediction of the failure event, but also the identification of the specific direction of slippage. This will enable the triggering of reactive regrasp primitives carried out by a second manipulator that can exploit the information of the direction of slippage to firmly secure the grasp.

In this work, we collected a grand total of 1800 independent trials. Of these, 56% was used to train the neural architecture, 24% for its validation and the remaining 20% for testing. Extensive research was carried out to identify the optimal recurrent neural architecture to use, aiming at maximizing the prediction accuracy over a dataset of testing trials, whereas minimizing the footprint of the network. With respect to the study by Arapi et al.,[22] where a convolutional neural network (CNN) was combined with a long short-term memory (two layers of 128 neurons) to perform the prediction, we removed the CNN for feature extraction and implemented a recurrent neural network (RNN) architecture based on gated recurrent units (GRUs)[23] (one layer of 128 neurons). Finally, we also developed a completely new online feedback system, which takes as input the inference of the proposed RNN (in terms of prediction of the sliding event as well as its direction, i.e., top and lateral) and selects a reactive regrasping primitive, performed by a second robotic arm–hand system (see **Figure 2**), that ultimately manages to firmly secure the grasp. Despite the fact that the network was trained only with spherical and cylindrical objects, we demonstrate high generalization capabilities of our framework, achieving a correct prediction of the failure direction in 75% of cases ($\approx$2 s in advance), and a 85% of successful regrasps, for a selection of 12 objects of common use.

## 2. Experimental Section

As introduced in the previous section, the goal of this work is to develop a closed-loop framework able to predict online if an object, grasped by a soft robotic hand, is sliding (and along which direction) and will likely drop. Such information is used to feed a reactive controller that triggers a regrasping primitive which, in turn, firmly stabilizes the grasp. As a test bench, we used two Franka Emika Panda manipulators[24], both endowed with two Pisa/IIT soft hands[25] as end effectors. To collect data for RNN training, we used a 3D-printed object, composed by a interchangeable handle and a support where one or more masses were placed to modify the weight of the object. We considered two different shapes of handles, a sphere and a cylinder, which forced the shape of the hand in two different configurations. These were presented to the robot (i.e., one robotic arm and hand system) with two roughness level, one smooth and one covered with sandpaper (400 Grit). The handle was grasped following two main approaches: top grasp, i.e., with the palm parallel to the horizontal plane, and lateral grasp, i.e., with the palm parallel to a vertical plane. For each of these grasp approaches, we further considered two potential failures types: central slippage, i.e.,
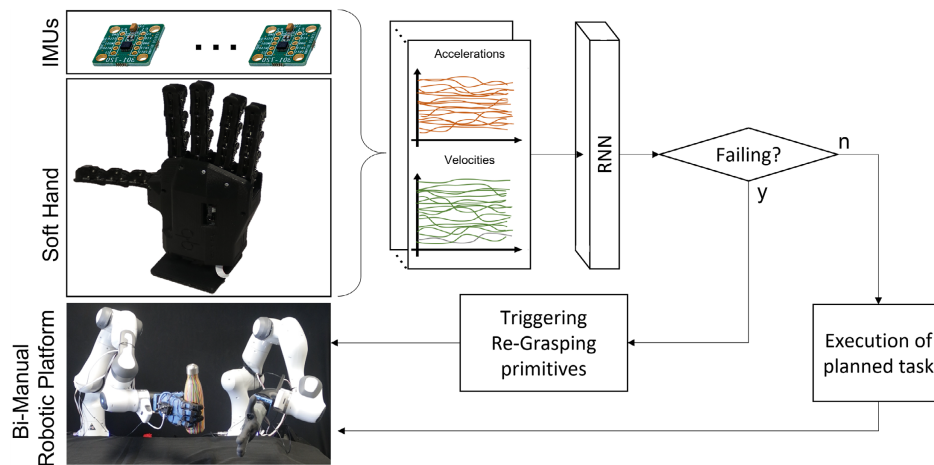
**Figure 2.** General scheme of the proposed closed loop framework. A bimanual robotic system, composed by two Franka Emika Panda Robots, endowed with two Pisa/IIT soft hands, is used as test bench. IMUs are fastened on the back of the fingers of the hand used to reach and grasp the object. A RNN architecture is used to continuously detect, on-line and with an inference rate of 5 Hz, the occurrence of a sliding event, and the direction of the relative motion. This information is then fed back to the controller to trigger a reactive regrasping primitive, which depends on the output of the RNN and on the current status of the first manipulator.

when the object slips along the long fingers, and lateral slippage, i.e., when failure is caused by a relative motion perpendicular to the long fingers (examples are shown in **Figure 3**a,b). Of note, to avoid that the network could identify failures only along the direction of the gravity, we included slippage data where the object was pulled off the soft hand along a direction perpendicular to the gravity itself, by a second robotic arm–hand system (i.e., Franka Emika Panda equipped with a soft hand) (see e.g., Figure 3a right, b left, 4b).

The position and orientation of hand and object during experiments were continuously tracked through a 3D motion tracking system (Optitrack Flex 13, NaturalPoint Inc., Corvallis, Oregon, USA, refresh rate 120 Hz). The robotic hand that performed the reach to grasp task was endowed with a soft glove, on which we mounted 17 IMUs, one for each phalanx, fastened on the back of the hand as in the study by Arapi et al.[22] Four IMUs were attached to the thumb, and three to each long finger. One additional sensor was placed on the hand dorsum, close to the wrist, for reference (see Figure 3c). Considering all the combinations discussed earlier, we performed a grand total of 1800 independent acquisitions, of which one-third was composed of successful grasps, one-third of central slippage, and one-third of lateral slippage. For each of these classes, we randomized the shape of the handle, the roughness level and the type of grasping approach (top versus lateral), making sure that the different parameters were represented in a balanced manner. A random weight, ranging between 200 and 700 g, was added to the object. For each trial, we recorded synchronously the stream of IMUs readings, the position of optical markers attached to hand and object, the encoder of the hand (which measures the degree of closure) and the robot joint positions, all with a refresh rate of 70 Hz.

For each trial, then, we segmented the portion of data we intended to use as input for the neural architecture. More specifically, we identified as initial frame of the sequence the instant in which the arm start lifting the object. The final

frame, instead, is identified as the one in which the distance between hand and object increases by 5 mm w.r.t. the previous values (i.e., the object is dropped). Finally, zero-padding was added at the beginning of each sample, to homogenize the trials length.

Once the dataset was built, to teach the network to recognize the event in advance, we removed from the dataset the final block of the signal, corresponding to a time slot immediately before the object drop. This has the twofold purpose of 1) removing high peaks in the signal stream caused by the drop of the object and 2) learning to recognize small oscillations that are characteristics of failure events in the first frames of sliding, rather than larger oscillations evident in the final portion of the signal (see **Figure 5**). We tested three different levels of anticipation, corresponding to 1, 2, and 3 s before the actual drop, shown in Figure 5 with a blue, green, and red dashed lines, respectively. Hereinafter, we will refer to the parameter quantifying this anticipation as $\Delta$.

Data were then randomized and splitted in three groups: 20% was devoted to testing and the remaining was further divided in 30% for validation and 70% for training.

The neural architecture we selected is based on gated recurrent units (GRU),[23] which are neurons with a feedback channel, which enables to store, and learn from, the history of a time series. Training was carried out using ADAM optimizer and Cross Entropy as loss function. Early stopping and dropout were also used to prevent overfitting. We tested different combination of hyperparameters, resulting in three different architectures that demonstrated the highest validation accuracy and the minimum footprint of the network (to minimize inference time), one for each $\Delta$ value considered. This is motivated by the fact that the larger is the model the larger is the time to perform inference. Among these, we selected for the implementation the network trained with $\Delta = 2$ s, because this provided a time horizon sufficiently large to eventually plan a recovery action, while keeping high accuracy values over validation data.
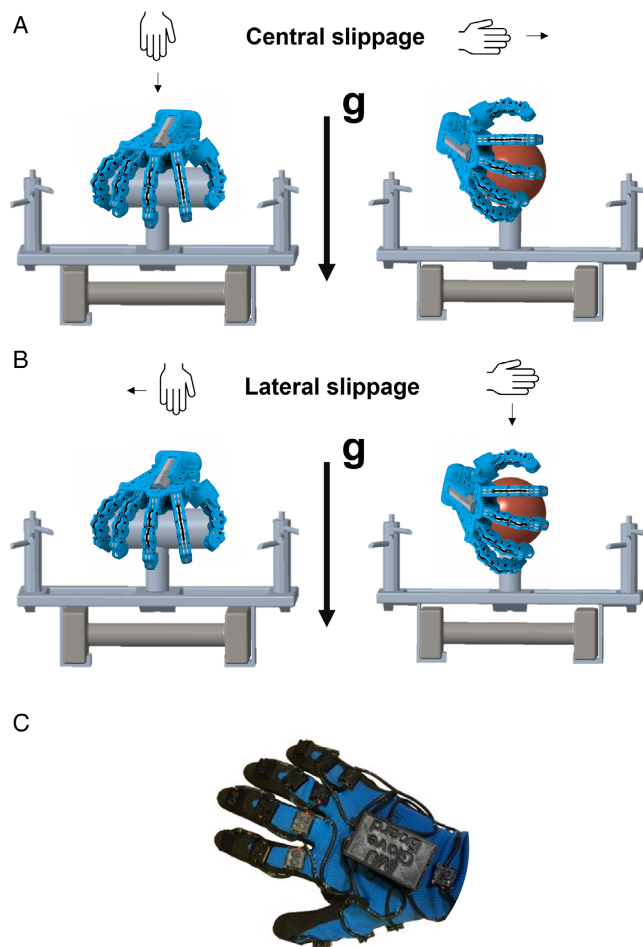
**ADVANCED
SCIENCE NEWS**

www.advancedsciencenews.com

**ADVANCED
INTELLIGENT
SYSTEMS**
Open Access

www.advintellsyst.com

**Figure 3.** Pictures of the experimental framework used for data collection. Two different handles (light gray in the figures) were used: a sphere (as in the right of panels A and B) and a cylinder (left of panel A and B). Handles were used with and without a sandpaper coverage to modify the roughness. Panel A shows the two hand configurations that we implemented to replicate central slippage. Panel B shows the two hand configurations for lateral slippage. In both panels, also the gravity vector is reported, to show that not all failures occur along the gravity direction. The object was endowed with two supports for the markers of the 3D motion tracking system (one on the right and one on the left) designed with the shape of a star to always ensure visibility of at least four markers as in the study by Verta et al.[26] The variable weight (in the range 200–700 g) was attached to the handle (dark gray in the figures). Panel C reports a picture of the IMU glove we mounted on the soft hand to continuously collect inertial measurements from each hand phalanx.

To develop an online implementation of reactive primitives triggered by the output of our grasp failure predictor, we built a first-in-first-out (FIFO) pile structure with a fixed size equal to the one used at training time, and containing fresh data coming from inertial sensors. During the execution of the online framework, the pile will always contain the last $N$ readings coming from IMUs, where $N$ is the number of time frames of acquisitions used during training (after prepadding). At startup, the pile is initialized as a zero matrix. Because with the architecture, we selected for implementation ($\Delta = 2$ s) we observed an average inference time of $\approx 0.15$ s, we implemented two ROS nodes,

the first, running at 70 Hz, where data were read from the IMU glove and collected into a dynamic array, and a second one, running at 5 Hz, where the block of data collected by the first node was inserted in the FIFO pile, removing the exceeding samples from the top of the pile (i.e., the oldest samples). Data contained in the pile were then provided as input to the RNN. When the predicted value is constant for at least five consecutive inference rounds, and the classified entry is a lateral or a central slippage, then this signal is used to trigger the reactive behavior of the second manipulator. Note that five represents a trade-off between promptness of response for the controller and number of false positives and was manually and heuristically tuned.

To test our methods, we implemented two parametric reactive primitives for the secondary robot, one appropriate for the central slippage and one for the lateral slippage (i.e., the two failure classes considered in this work). More specifically, we programmed the first primitive (i.e., for central slippage) as a linear interpolation between the initial robot configuration and the Cartesian position of the first end effector. The orientation of the second hand is imposed to be with the palm upward (see **Figure 6**a). The estimation of Cartesian forces provided by the second manipulator is continuously read and fed back to the controller, in such a way that when the module of the readings overcome a certain threshold (2 N) we assume that the robot is in contact with the object and we stop the execution of the primitive. In case of lateral slippage, the second robot is programmed to reach via a linear interpolation the position of the first end effector. The orientation, instead, is rotated along the direction of the long fingers in such a way that the angle between the horizontal plane and the plane of the palm is 45° (see Figure 6b). Also in this case, we command as reference the position of the first end effector, and exploit the estimation of contact forces to identify the contact.

We tested our online framework with two additional experiments. First, we replicated the failures with the same setup used for data collection. Twenty trials were performed for each of the three classes, considering randomly one of the handles of Figure 3. We then considered a selection of 12 objects of common use, of which ten are extracted from the Yale-CMU-Berkeley (YCB) dataset[27] and two are l-shaped objects with smooth and rough surfaces (see **Figure 7**). This selection was made with the purpose of forcing different types of power grasps, such as power circular, power prismatic, palm circular, and palm prismatic (for terminology, we refer to the study by Arapi et al.[28]). We made sure that the objects' weight was in the range between 200 and 700 g, by adding external weight when necessary. For each of these objects, we used the grasping strategy afforded by the object. Indeed, as hypothesized in the study by Gibson,[29] the geometry of an object suggests one (or more) preferable grasping approaches, which we attempted to respect. For this reasons, tall objects, such as standing bottles, were grasped using a lateral grasp, whereas short ones were grasped using a top grasp. Of note, the bottle was presented in both the standing and lying down configuration. We repeated the grasp of each object ten times, forcing its ecological failure by regulating the strength of the hand closure,[25] achieving a grand total of 130 samples.
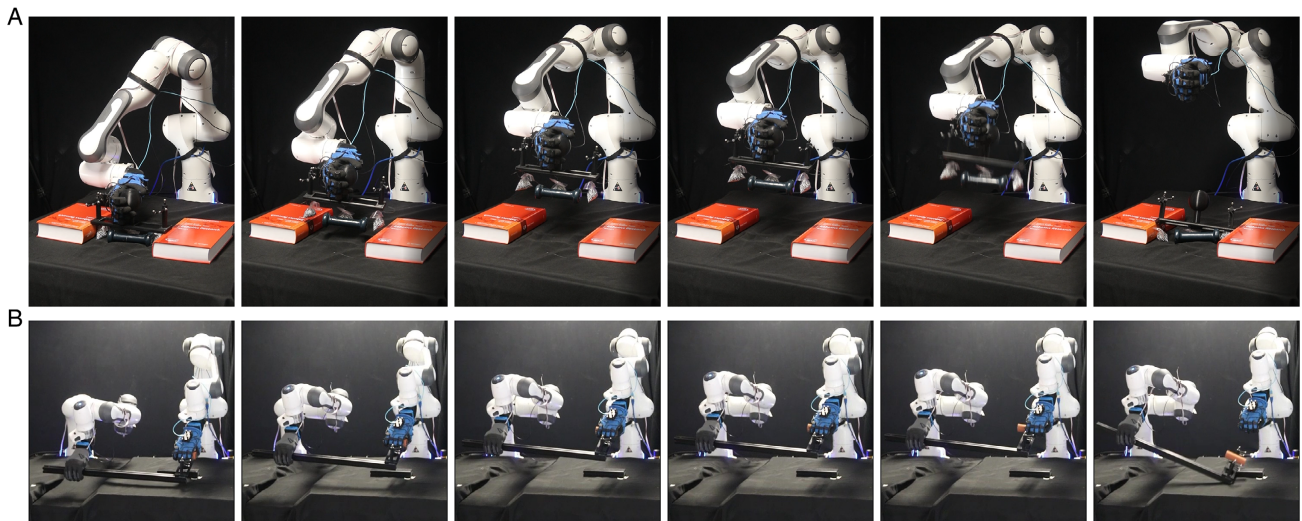
**Figure 4.** Two examples of failures events executed during data collection. a) Lateral slippage with a spherical smooth handle. b) Lateral slippage with a rough cylindrical handle, caused by the second robot.
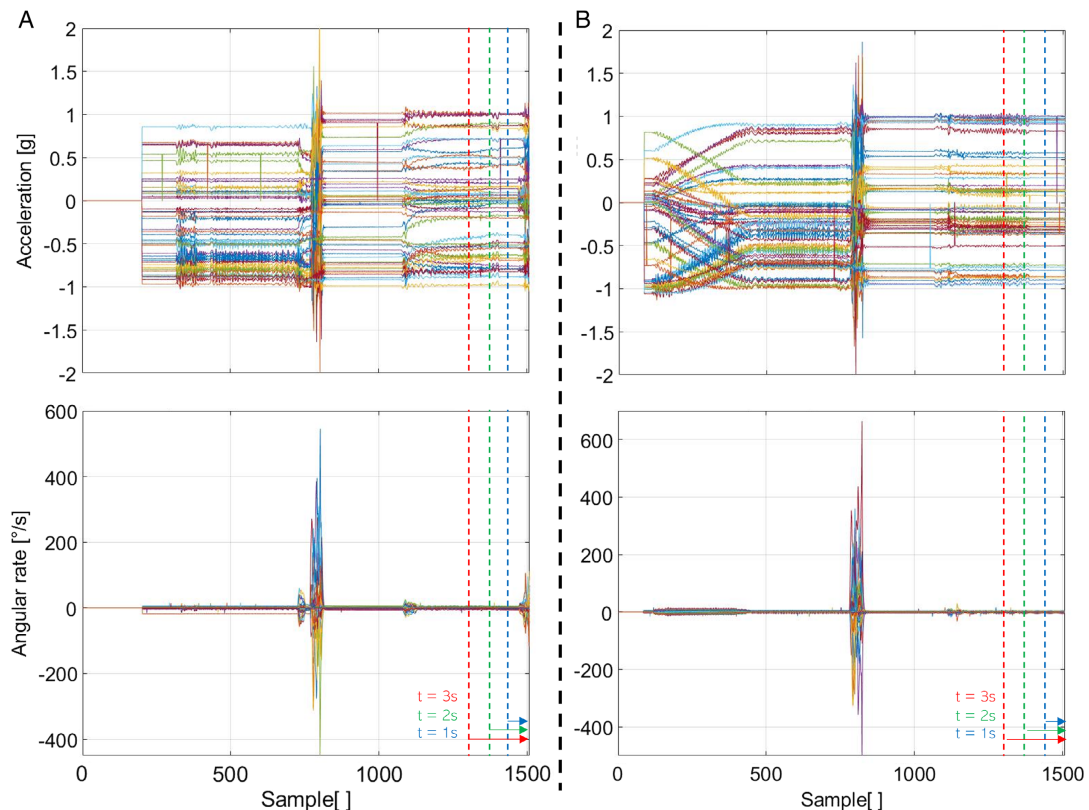


**Figure 5.** Stream of raw accelerometers (top row) and gyroscopes (bottom row) during a) a failed grasp and b) a successful grasp. Colored dashed lines identify the initial frame of the block of signal that is removed from data before training, corresponding to 1 s (blue), 2 s (green), and 3 s (red). We refer to this quantity as $\Delta$.

## 3. Results

As already mentioned in the previous section, we tested our framework in two different ways. First, we validated the network by assessing the prediction accuracy over a pool of test data not used during training and validation, consisting of 360 independent samples of three classes: successful grasp, central slippage, and lateral slippage. Then, we implemented our network in an
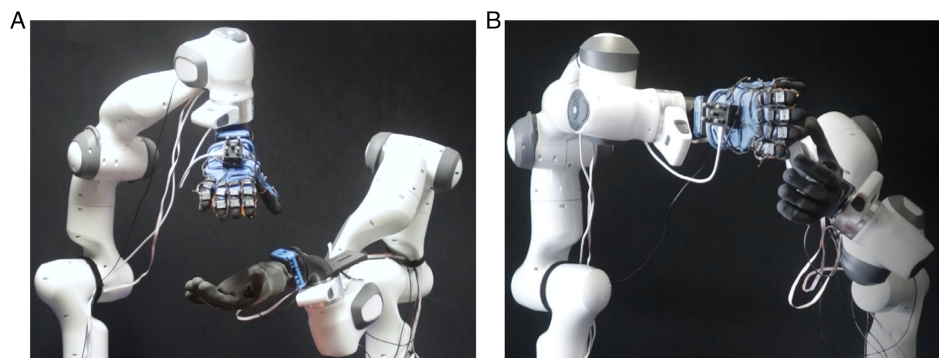
**Figure 6.** Regrasp primitives considered for a) top grasps and b) lateral grasps.



**Figure 7.** Objects of common use selected for the experiments: a sauce bottle, an apple, a tennis ball, a squeeze tube, a mug, a box, a saucepan, a water bottle, a food box, a shampoo bottle, a rough l-shape, a smooth l-shape. We increased the weight of some of these objects by adding external weight (as done during experiments for dataset collection) to match the range between 200 and 700 g.

on-line integrated framework of failure prediction and reactive regrasp. We tested this implementation over a selection of 12 objects of common use extracted from the YCB dataset.[27]

### 3.1. Validation of the Neural Architecture

Considering an anticipation time $\Delta$ of 1, 2, and 3 s, we converged to three optimized architectures, all based on GRU neurons. The optimal architecture with $\Delta = 1$ is composed by two layers of 64 neurons and was trained with a dropout of 0.3. This network demonstrated a validation accuracy of $0.93 \pm 0.003$ over ten different rounds of training (all starting from a random seed). For $\Delta = 2$, the optimal selection converged to a single layer of 128 neurons trained with a dropout of 0.5, achieving a validation accuracy of $0.91 \pm 0.01$ over ten different rounds of training (all starting from a random seed). Finally, with $\Delta = 3$ s, the model consisted of two layers of 64 neurons, trained with a dropout of 0.3, yielding a validation accuracy of $0.87 \pm 0.02$ over ten different rounds of training (all starting from a random seed). After validation, we quantified the accuracy of prediction also

over fresh data, not used during the training phase. This new dataset consisted of 360 samples, 120 for each class. Confusion matrices of the classification for different values of $\Delta$ are shown in **Figure 8**. We obtained an overall test accuracy of 87%, 84%, and 76% for $\Delta = 1$ s, 2 s, and 3 s respectively.

### 3.2. Validation of the Online-Integrated Framework

We decided to consider for the online implementation the architecture trained with $\Delta = 2$ s, because this represents an appropriate trade-off between satisfactory prediction performances and capabilities of detecting small oscillations that are present in the early stages of sliding (minimizing the network footprint). As introduced in Experimental Section, we performed two different experiments to test the capabilities of our framework in the online predict-and-regrasp task. The first one consisted in replicating the experimental setup already used to collect the training data. In this case, with a pile that continuously (with a refresh rate of 5 Hz) updates the stream of data given as input to the neural architecture, we obtained a correct classification in ≈78% of
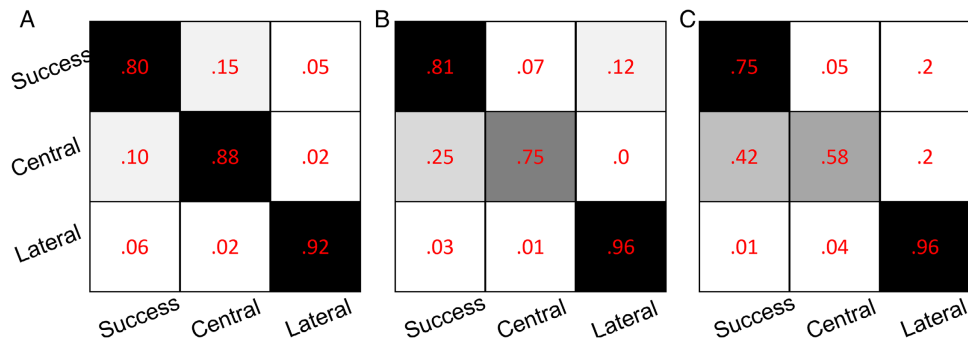
**Figure 8.** Confusion matrices referred to failure classification over test data for a) Δ=1 s, b) Δ=2 s, and c) Δ=3 s. Values are in percentage versus the total number of entries of each class. On the rows the real class, on the columns the predicted class. Cells are color-coded: black stands for 100%, white stands for 0%.
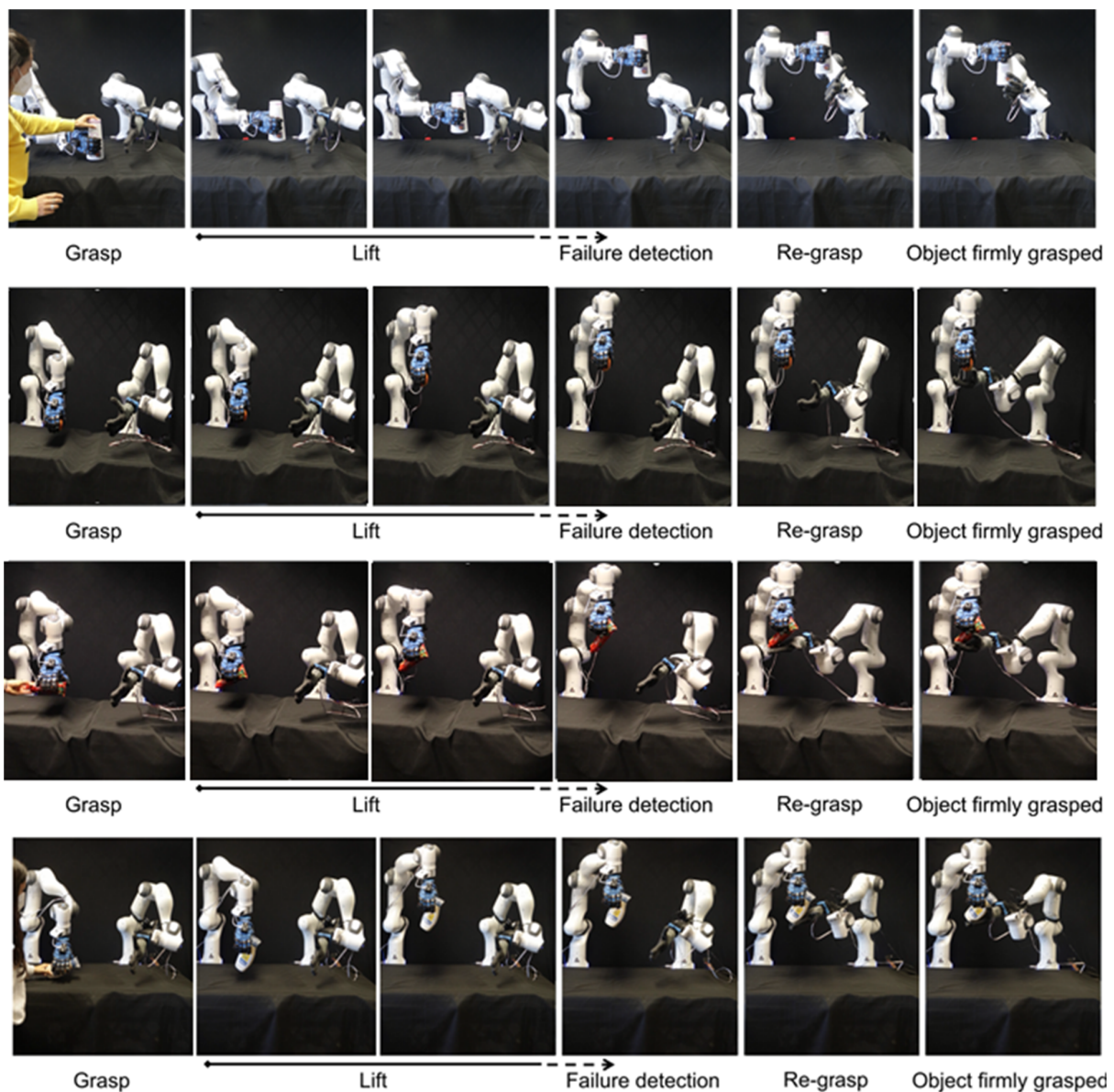


**Figure 9.** Four examples of our framework while the occurrence and the direction of a failure was correctly predicted and a reactive primitive successfully prevented the drop of the object. From top to bottom, a shampoo bottle, an apple, a squeeze tube, and a sauce bottle.

cases, of which ≈87% resulted in a successful robot regrasp. However, a correct prediction of the failure does not necessarily match with a successful regrasp, because after triggering the reactive primitive the robot could spend a certain amount of time to plan and execute the trajectory (≈1.5 s). For this reason, in certain cases, especially with very smooth objects, the total success rate of the regrasp could be lower, and it is more appropriate to report on the number of successful regrasps over the ones correctly predicted. Indeed, in this first experiment we had that for occurrences of central slippage we were able to successfully prevent the failure in 80% of cases, whereas the performances increased to 94% for lateral slippage. This is caused by the fact that in the first case the time of sliding is shorter, on average, than the second one. We further validated our framework by performing a second experiment with 12 objects, of which ten are extracted from the YCB dataset[27] (see Figure 7, snapshot of the experiments are reported in **Figure 9**). Also in this case, we verified the prediction and classification accuracy and then quantified the success rate of the failure prevention over the cases in which we were able to successfully predict the failure. Over a grand total of 130 experiments, we achieved a classification accuracy of 75% and, for the cases in which the type of failure was predicted correctly, we successfully prevented the failure with our reactive primitive in 85% of cases. Of note, we observed marked differences across objects. More specifically, very smooth and spherical objects, such as the wooden apple in our pool of objects, although easily classified by our neural architecture (with a correct classification in 80% of cases), was successfully regrasped in the 25% of cases of correct classification only. Others, instead, such as the mug, the saucepan, the tennis ball, the squeeze tube and the launch box, were successfully regrasped in all the cases in which the neural architecture was able to correctly predict the failure.

## 4. Discussions and Conclusions

With this article, we demonstrated the feasibility, and provided an implementation of a neural architecture that can predict—up to 87% with test data—the occurrence and the direction of grasp failure, considering accelerations and angular velocities collected from a soft robotic hand (mounted on a robotic manipulator and equipped with an IMU glove) that autonomously grasp an object. We implemented our framework relying on a GRU architecture, a widely popular and consolidated RNN. Our implementation enables the triggering of reactive primitives, performed by a second robotic arm–hand system, achieving a correct prediction (≈2 s in advance) of the failure occurrence and its directions in 75% of cases, and—when correctly classified—firmly secure the grasp with a recovery action, with a success rate of 85%. These results refer to an experiment conducted with a pool of objects of common use, which were never used during the training phase, whereas the performances using the same experimental setup of the training phase were 78% and 87%, respectively. Note that these results come from a combination of different factors, such as the smoothness of the object and the upper bound of the velocity of the manipulator. Of note, our implementation is completely online, from inference to motion execution, with an average inference time of 0.15 s and an average time required to complete the reactive behavior of 1.5 s. We noticed that certain objects, such as the wooden apple in our second experiments, were particularly harsh for our framework. Indeed, while the sliding was correctly predicted and classified in the 80% of cases, the failure was extremely quick and resulted in a successful regrasp for the 25% of cases only. This is mainly related to the control of the hand itself and we believe that the extension of our reactive behavior to other regrasping primitives may improve these performances.

To improve the prediction accuracy, on one side, and the regrasp success on the other side, our future work will focus, on one side, on the investigation of the usage of other tactile sensors, such as the Tactip,[20] to gather a larger amount of information during the grasp, which could improve the accuracy of our prediction. On the other side, we will consider the use of neural architecture search (NAS) techniques to optimize the design of the neural network. Furthermore, we will also consider the usage of supplementary sensing sources, as for example cameras. In this way, sensor fusion could be exploited to feed with a more complete source of information the neural architecture and to improve the overall failure prediction accuracy. However, it is also worth mentioning that such an improvement would come with a significant increase in the dimensionality of raw data, resulting in higher complexity of the mechatronic system and in a larger footprint of the neural architecture. For this reason, we believe that a trade-off must be reached, depending on the resources available for a given application. For example, for fully autonomous robots, which should process the whole information with on-board electronics (possibly on the edge), one could use a minimalistic tactile sensing as the one used in this article to minimize the footprint of the network, whereas for industrial scenarios, it may be feasible to have more complex systems. At the same time, we are planning to further expand the pool of reactive primitives considered, including also single-arm actions, such as end effector reorientation and hand squeezing force regulation. To this aim, additional sources of information, as for example a vision layer, will be evaluated, which will help in discriminating which strategy could be the more appropriate for the specific case.

Ultimately, we believe that our work may represent a valuable contribution toward the development of intelligent manipulators, capable of identifying online whether a task is performed correctly, eventually triggering reactive behaviors to adapt the execution of the action to the expected goal.[18] This will help in developing grasp planning that minimize the force exerted during the object grasps, and demanding to the prediction–regrasping component the correction of a possible failure. This, together with the intrinsic adaptability of the soft hands, could offer a viable solutions for the grasping of fragile and delicate objects.

## Conflict of Interest

The authors declare no conflict of interest.

## Data Availability Statement

Research data are not shared.

## Keywords

[1] C. Piazza, G. Grioli, M. Catalano, A. Bicchi, *Annu. Rev. Control Rob. Auton. Syst.* **2019**, *2*, 1.

[2] J. Hughes, U. Culha, F. Giardina, F. Guenther, A. Rosendo, F. Iida, *Front. Rob. AI* **2016**, *3*, 69.

[3] R. Deimel, O. Brock, *Soft Robotics*, Springer, New York, NY **2015**, pp. 211–221.

[4] G. Averta, F. Angelini, M. Bonilla, M. Bianchi, A. Bicchi, *IEEE Rob. Autom. Lett.* **2018**, *3*, 2686.

[5] C. Choi, W. Schwarting, J. Delpreto, D. Rus, *IEEE Rob. Autom. Lett.* **2018**, *3*, 2370.

[6] M. Pozzi, S. Marullo, G. Salvietti, J. Bimbo, M. Malvezzi, D. Prattichizzo, *Int. J. Rob. Res.* **2020**, *39*, 1706.

[7] L. Schramm, A. Sintov, A. Boularias, in *2020 IEEE Inter. Conf. on Robotics and Automation (ICRA)*, IEEE, Piscataway **2020**, pp. 4579–4585.

[8] F. Ficuciello, A. Migliozzi, E. Coevoet, A. Petit, C. Duriez, in *2018 IEEE/RSJ Inter. Conf. on Intelligent Robots and Systems (IROS)*, IEEE, Piscataway **2018**, pp. 4007–4013.

[9] Y. Haibin, K. Cheng, L. Junfeng, Y. Guilin, *Mech. Mach. Theory* **2018**, *128*, 254.

[10] G. Santaera, E. Luberto, A. Serio, M. Gabiccini, A. Bicchi, in *2015 IEEE Inter. Conf. on Robotics and Automation (ICRA)*, IEEE, Piscataway **2015**, pp. 2728–2735.

[11] S. Kitagawa, K. Wada, K. Okada, M. Inaba, in *Inter. Conf. on Intelligent Autonomous Systems*, Springer, New York, NY **2018**, pp. 428–439.

[12] R. Krug, A. J. Lilienthal, D. Kragic, Y. Bekiroglu, in *2016 IEEE Inter. Conf. on Robotics and Automation (ICRA)*, IEEE, Piscataway **2016**, pp. 165–171.

[13] A. Ajoudani, E. Hocaoglu, A. Altobelli, M. Rossi, E. Battaglia, N. Tsagarakis, A. Bicchi, in *2016 IEEE Inter. Conf. on Robotics and Automation (ICRA)*, IEEE, Piscataway **2016**, pp. 1972–1979.

[14] Z. Kappassov, J.-A. Corrales, V. Perdereau, *Robot. Auton. Syst.* **2015**, *74*, 195.

[15] L. Pinto, A. Gupta, in *2016 IEEE Int. Conf. on Robotics and Automation (ICRA)*, IEEE, Piscataway **2016**, pp. 3406–3413.

[16] X. Song, H. Liu, K. Althoefer, T. Nanayakkara, L. D. Seneviratne, *IEEE Trans. Rob.* **2014**, *30*, 203.

[17] G. Zöller, V. Wall, O. Brock, in *2018 IEEE/RSJ Inter. Conf. on Intelligent Robots and Systems (IROS)*, IEEE, Piscataway **2018**, pp. 6986–6991.

[18] M. Bianchi, G. Averta, E. Battaglia, C. Rosales, M. Bonilla, A. Tondo, M. Poggiani, G. Santaera, S. Ciotti, M. G. Catalano, A. Bicchi, in *2018 IEEE Int. Conf. on Robotics and Automation (ICRA)*, IEEE, Piscataway, NJ **2018**, pp. 7794–7801.

[19] Y. She, S. Q. Liu, P. Yu, E. Adelson, arXiv preprint arXiv:1910.01287 **2019**.

[20] B. Ward-Cherrier, N. Pestell, L. Cramphorn, B. Winstone, M. E. Giannaccini, J. Rossiter, N. F. Lepora, *Soft Rob.* **2018**, *5*, 216.

[21] E. Donlon, S. Dong, M. Liu, J. Li, E. Adelson, A. Rodriguez, in *2018 IEEE/RSJ Inter. Conf. on Intelligent Robots and Systems (IROS)*, IEEE, Piscataway **2018**, pp. 1927–1934.

[22] V. Arapi, Y. Zhang, G. Averta, M. G. Catalano, D. Rus, C. Della Santina, M. Bianchi, in *2020 3rd IEEE Inter. Conf. on Soft Robotics (RoboSoft)*, IEEE, Piscataway **2020**, pp. 653–660.

[23] R. Dey, F. M. Salem, in *2017 IEEE 60th Int. Midwest Symposium on Circuits and Systems (MWSCAS)*, IEEE, Piscataway **2017**, pp. 1597–1600.

[24] F. Emika, https://www.franka.de/ (accessed: August 2021).

[25] M.G. Catalano, G. Grioli, E. Farnioli, A. Serio, C. Piazza, A. Bicchi, *Int. J. Rob. Res.* **2014**, *33*, 768.

[26] G. Averta, C. Della Santina, E. Battaglia, F. Felici, M. Bianchi, A. Bicchi, *Front. Rob. AI* **2017**, *4*, 37.

[27] B. Calli, A. Singh, J. Bruce, A. Walsman, K. Konolige, S. Srinivasa, P. Abbeel, A. M. Dollar, *Int. J. Rob. Res.* **2017**, *36*, 261.

[28] V. Arapi, C. Della Santina, G. Averta, A. Bicchi, M. Bianchi, *IEEE Rob. Autom. Lett.* **2021**.

[29] J. J. Gibson, *Perceiving, Acting, and Knowing*, Lawrence Erlbaum Associates, Hilldale, NJ **1977**, Vol. 1, p. 67.