ARTIFICIAL INTELLIGENCE

Vision-based grasp learning of an anthropomorphic hand-arm system in a synergy-based control framework

F. Ficuciello[1]*, A. Migliozzi[1], G. Laudante[2], P. Falco[3], B. Siciliano[1]

In this work, the problem of grasping novel objects with an anthropomorphic hand-arm robotic system is considered. In particular, an algorithm for learning stable grasps of unknown objects has been developed based on an object shape classification and on the extraction of some associated geometric features. Different concepts, coming from fields such as machine learning, computer vision, and robot control, have been integrated together in a modular framework to achieve a flexible solution suitable for different applications. The results presented in this work confirm that the combination of learning from demonstration and reinforcement learning can be an interesting solution for complex tasks, such as grasping with anthropomorphic hands. The imitation learning provides the robot with a good base to start the learning process that improves its abilities through trial and error. The learning process occurs in a reduced dimension subspace learned upstream from human observation during typical grasping tasks. Furthermore, the integration of a synergy-based control module allows reducing the number of trials owing to the synergistic approach.

INTRODUCTION

The ability to effectively manipulate different objects and to successfully use a variety of tools are two of the key skills that humans developed during their evolutionary history. For robots to approach the capabilities of humans to interact with their environment, the design, implementation, and control of dexterous, anthropomorphic hands appear to be pivotal points and have raised a lot of interest in the scientific community over the course of the years.

The human hand is a very complex, articulated biomechanical system, and the problem of replicating its structure and capability is very challenging in terms of not only mechanical design but also motion planning and control. Inspired by studies in neuroscience (*1*), the idea of considering coordinated joint motion patterns for robotic hands, realized both by means of mechanical transmissions and by developing control strategies in a subspace of reduced dimensionality, has been successfully applied in several works (*2*–*4*).

Many different approaches to grasp synthesis problem have been proposed, but most of them can be classified as either analytic or empirical (data driven) (*5*). Analytic grasp synthesis usually consists of formulating a constrained optimization problem to obtain grasps that satisfy one or more desired properties, such as stability and dexterity (*6*). Empirical approaches, instead, try to combine perceptual information acquired through sensors and previous knowledge coming from either experience or human demonstrations to compute grasps that optimize some quality metrics (*7*). Different classifications have been proposed for data-driven algorithms: For example, in (*5*), the authors distinguished between techniques based on human observation and those based on object features, whereas in (*7*), they were categorized on the basis of the amount of previous knowledge of the object (known, familiar, or unknown object). A key step in data-driven algorithms is grasp selection: Based on the available data, an appropriate grasp can be selected either by choosing one of the candidates in a database or by synthesizing it from scratch. In general, the features of the object are fundamental in selecting a good grasp; different solutions have been proposed to associate grasps to different objects. In some works, the object was modeled with basic shape primitives, which were used to reduce the number of candidate grasps (*8*, *9*); in (*10*), SVM (support vector machines) regression was used to match object shape, grasp parameters, and grasp quality; and in (*11*, *12*), grasping regions of the object were identified.

After a grasp has been selected, it is usually evaluated either in simulation or on the real robot, according to some metrics that can discriminate between good and bad grasps. Over the years, a variety of metrics for evaluating grasp performances have been discussed. An extensive overview of the different quality indices proposed in literature is provided in (*13*). Grasp quality measures can be divided in two broad categories: those associated with the positions of contact points on the object and those that depend on the hand configuration. The former can be further divided in three subgroups: measures based on the algebraic properties of the grasp matrix **G**, measures based on geometric considerations, and measures that keep into account limitations on the magnitude of the forces that can be exerted by the fingers.

Among reinforcement learning (RL) algorithms available in literature, there are two model-based RL algorithms that can be suitable for robotic manipulation: PILCO (probabilistic inference for learning control) and PI-REM (policy improvement with residual model). PILCO is a state-of-the-art model-based policy search algorithm introduced by Deisenroth and Rasmussen in 2011 (*14*). PI-REM is a recent model-based algorithm proposed by Saveriano *et al.* in 2017 (*15*). The key idea of PILCO is to learn a probabilistic forward model of the system dynamics to explicitly take uncertainties into account. The model is implemented as a nonparametric Gaussian process with previous mean function and squared exponential kernel. The policy improvement is gradient based, and the gradient is computed analytically. The basic idea of PI-REM is to develop initially an approximate model of the system, controlled with an approximate policy (learned with PILCO); in a second stage, a residual model (difference between the approximate and real model) is learned and used to improve the real policy that is applied directly on the robot. In this work, we considered a robotic system consisting of a five-fingered hand and a redundant anthropomorphic manipulator provided with many degrees of freedom (DoFs) (20 for the hand and 7 for the arm); therefore, we used dimensionality reduction techniques to define the learning algorithm in a subspace of reduced dimensionality (*16*). In particular, the concept of postural synergies, originally introduced in (*1*), was transferred to the robotic hand, as proposed in (*2*, *17*), by means of principal components analysis (PCA). A data-driven approach to grasp synthesis, combining learning from

[1]Prisma Lab, University of Naples Federico II (DIETI), Naples, Italy. [2]Industry AMS, Naples, Italy. [3]ABB Corporate Research, Västerås, Sweden.
*Corresponding author. Email: fanny.ficuciello@unina.it

demonstration based on neural networks (NN) and RL based on policy improvement with path integrals, was adopted to provide the robot with the ability to learn and improve grasps for previously unknown objects. To automatize the grasping process, we used a vision system to detect the object in the scene and to estimate its pose and geometric features. With respect to (*18*), this work contributes an upgraded framework that includes visual perception and its integration in the learning pipeline. In (*18*), the geometric parameters of the objects were assumed to be known, but in this work, they were computed from an RGB-D camera. Therefore, the level of autonomy of the algorithm has been increased. On the other hand, the accuracy and the resolution of the camera, as well as the efficiency of the algorithm adopted for geometry reconstruction and pose detection, can affect the learning process. The experiments run in this work demonstrate that the algorithm is stable and robust in case of uncertainties on perception of the environment.

RESULTS

Overview of the algorithm

A schematic overview of the proposed algorithm is provided in Fig. 1. First, information about the region of interest in the scene is acquired in the form of a point cloud through an RGB-D sensor. The acquired data are filtered and processed by the object recognition module, which is responsible for detecting the objects in the scene and estimating their shape, dimensions, and pose.

The extracted features are sent as an input to the NN module, where two sets of multiple NN, one for the hand and one for the arm, have been trained on data acquired from human demonstration through motion capture tools; the provided output is a vector of three synergy parameters for the hand and two coefficients for the arm. At this point, the parameters provided by the NN are used to initialize an RL loop, based on the policy search paradigm, to endow the robotic system with the ability to improve its performances over time, with reference to some appropriate quality metrics. On the basis of the initial policy parameters, some samples are extracted from Gaussian multivariate distributions for the coefficients of both the hand and the arm; each of these samples generates a trajectory for the robot, in Cartesian coordinates, during the planning phase. The planned trajectories are executed on the real system using a kinematic control strategy (*19*), where a closed-loop inverse kinematic algorithm is used to convert references from the operative space to the configuration space of the robot, serving as inputs to the lower-level controllers embedded in the system.

Each of the executed trajectories is assigned a cost that consists of two terms: a binary score that heavily penalizes trajectories leading to failed grasps and a cost function related to a quality index of the force-closure properties of the grasp (*20*). Last, based on the obtained costs, the policy is updated and a new set of parameters is extracted. This loop can be either repeated for a fixed number of times or run until the cost function becomes lower than a desired threshold.

Dimensionality reduction

Robots are complex nonlinear systems, with many DoFs, that have to execute complex actions while potentially interacting with unstructured environments. Therefore, most of the machine learning techniques that have been successfully applied in other fields are difficult to apply to the huge amount of parameters involved in the context of robotics (*21*, *22*).

A potential solution to this problem is to adopt different techniques that are targeted at reducing the dimensionality of the system. Postural synergies, sometimes also called eigengrasps, were initially studied in the field of neuroscience: It had been observed that the movement of the human hand during grasping and manipulation is dominated by movements in a space of reduced dimensionality compared with the number of DoFs of the human hand (*1*).

This concept can be transferred to robots by means of data analysis techniques, such as PCA (*23*). By taking advantage of this idea, the anthropomorphic hand can be controlled directly in the postural synergies subspace, greatly reducing the complexity of planning, control, and

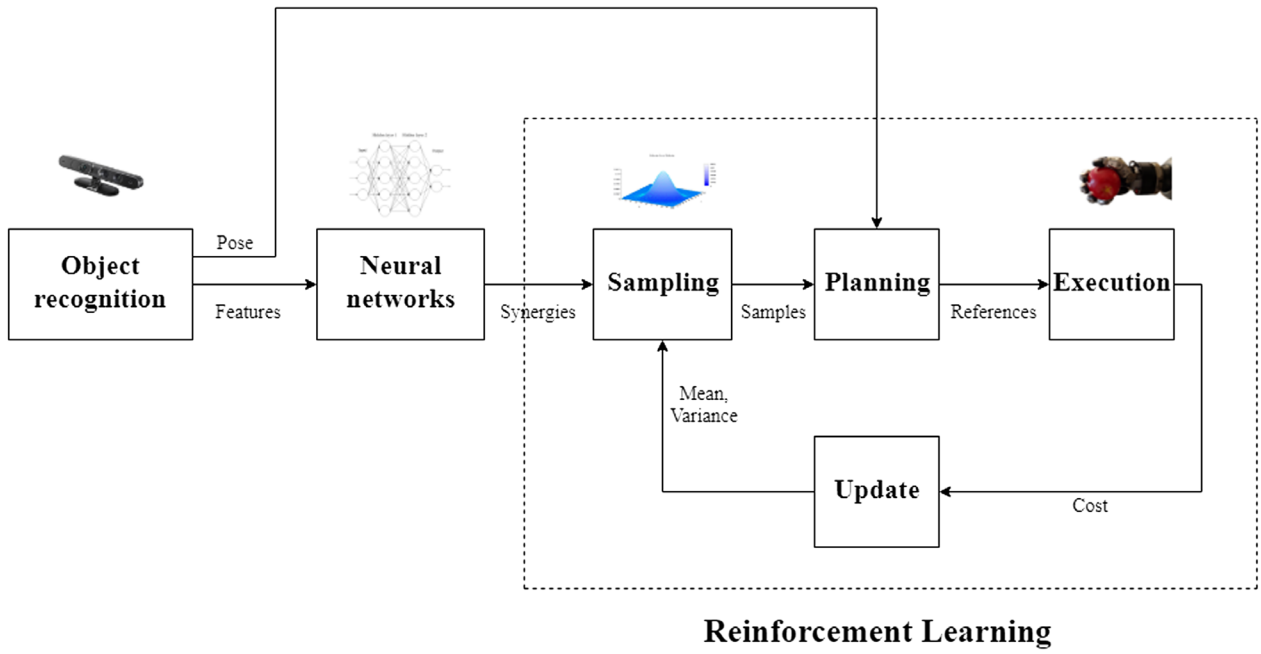

Fig. 1. Schematic overview of the proposed algorithm.

learning. The postural synergies of the SCHUNK S5FH (SVH) computed in (*17*) have been implemented in this work. The number of motors is notably lower than the number of joints; thus, joint motion couplings are regulated by means of mechanical synergies defined via mechanical transmissions. The differential kinematics between the mechanical synergies subspace and the Cartesian space is represented by $\dot{\mathbf{x}} = \mathbf{J}_{h_m}\dot{\mathbf{m}}$, where $\mathbf{J}_{h_m} \in \mathbb{R}^{n_x \times n_m}$ is the mechanical synergies Jacobian and is computed as $\mathbf{J}_{h_m} = \mathbf{J}_h\mathbf{S}_m$, such that

$$\dot{\mathbf{x}} = \mathbf{J}_h\mathbf{S}_m\,\dot{\mathbf{m}} = \mathbf{J}_h\dot{\mathbf{q}} \tag{1}$$

$\mathbf{x} \in \mathbb{R}^{n_x}$, with $n_x = 15$, is the position vector of the five fingertips and $\mathbf{J}_h \in \mathbb{R}^{n_x \times n_q}$ is the S5FH hand Jacobian. $\mathbf{S}_m \in \mathbb{R}^{n_q \times n_m}$ is the matrix of the mechanical synergies and maps motor velocities into joint velocities, $\mathbf{S}_m\dot{\mathbf{m}} = \dot{\mathbf{q}}$. In addition to the mechanical synergies of the hand, motion coordination patterns or motor synergies can be computed to further reduce the number of parameters needed to plan and control the grasping activities. Synergies subspace for hand control has been computed using human grasp data and a mapping algorithm developed in (*24*). Human grasp data are based on fingertip measurements that are used as desired references in a closed-loop inverse kinematic scheme that is in charge of reconstructing the hand configuration. The maps between the synergies subspace and the motor space, and between the synergies subspace and the joint space, are given by $\mathbf{m} = \mathbf{S}_s\boldsymbol{\sigma} + \bar{\mathbf{m}}$ and $\mathbf{q} = \mathbf{S}_m(\mathbf{S}_s\boldsymbol{\sigma} + \bar{\mathbf{m}}) + \mathbf{q}_0$, respectively.

The matrix $\mathbf{S}_s \in \mathbb{R}^{n_m \times n_s}$ represents the base of the synergies subspace, whose dimensions depend on the number of eigengrasps considered to approximate the grasp. Thus, the number n_s can vary from 1 to 9. In this work, the synergies subspace is three-dimensional, and thus, it is obtained by choosing $n_s = 3$. Thus, the columns are the first three eigenvectors (or eigengrasps) with higher variance computed using PCA on the grasp dataset mapped from the human hand to the robotic hand. Last, $\bar{\mathbf{m}} \in \mathcal{M} \subseteq \mathbb{R}^{n_m}$ is the zero offset of the motor synergies subspace.

Object recognition and pose estimation

In this work, a simple object-recognition and pose-estimation pipeline has been implemented. We used a semistructured environment involving a table on which different objects are placed. To simplify, we assumed that such objects could be recognized as sphere or cylinder. We used an Asus Xtion PRO LIVE camera for point cloud data acquisition.

We divide the analysis in two phases: syntactic phase, in which different clusters have been extracted from the scene, and semantic phase, in which all the clusters found are tested with two RANSAC models (sphere and cylinder) and the object is labeled as the best-fitting model.

Syntactic phase

After taking a point cloud of the scene, we follow a simplified processing pipeline. Because we have both a static scene and static position between the scene and the camera, we can assume that all the data out of certain limits can be cut out and filtered. As a second step, we find the largest planar component of the scene, representing the table, and extract it; we also need to divide all the other points in different clusters so that we obtain all the clusters that represent all the candidate objects that will be classified (see Fig. 2). For this purpose, we implement the Euclidean cluster segmentation criteria described as follows:

The point cloud P is represented in a k-d tree structure;
For each point p_i in P:
  Add p_i to the current queue Q;
  For each point p_i in Q do:
    add every neighbor of p_i in a radius r to Q unless it has already been processed.
  When the list of all points in Q has been processed, add Q to the list of clusters C, and reset Q to an empty list;

The algorithm terminates when all points $p_i \in P$ have been processed and are now part of the list of point clusters C.

In Algorithm 1, we describe in short how the syntactic phase work. The input to the syntactic phase algorithm is a point cloud S of the scene, and the output is the vector of point cloud *clusters*[*pointCloud*].

Algorithm 1 Syntactic phase
1: function (*clusters*[*pointCloud*]) = syntacticPhase(PointCLoud*S*)
2: (S_f) = regionFilter(S, *regionLimits*)
3: (S_f) = removeLargestPlanarComponent(S_f)
4: (*clusters*[*pointCloud*]) = EuclideanClusterExtraction(S_f)

Semantic phase

Now, we have to label the different objects found. Each object is processed both as a cylinder and as a sphere using RANSAC models. The criteria adopted are related to the number of inliers and outliers from which the RANSAC algorithm comes out. In this way, we can classify each object and establish how much of it is sphere or cylinder. Both spheres and cylinders are associated with a vector of parameters, $\boldsymbol{\vartheta}$. For spheres, it is $\boldsymbol{\vartheta} = (r, c)$, where r and c are the radius and the three-dimensional (3D) center of the mathematical model, respectively. For cylinders, the parameters are $\boldsymbol{\vartheta} = (r, c, d, h)$, where r is the radius, c is the 3D centroid of the cluster, d is the vector representing the axis of the cylinder, and h is the height computed as the distance between two points that have the maximum and minimum coordinate values on the maximum variance axis. Such criteria allow us to also use different objects that are quasi-sphere or quasi-cylinder (such as a cup of tea or a bottle). This approach emphasizes the robustness of the system to grasp or manipulate objects that have a similar shape to those already known. In the recognition phase, we assume that all of the objects can be sufficiently approximated with spheres or cylinders. Although this approximation can introduce errors in recognition and pose estimation, the learning process makes the system robust to perception errors. The

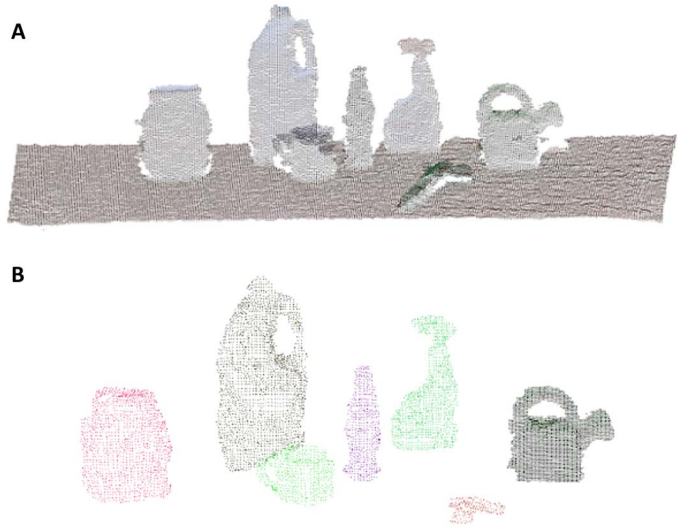

Fig. 2. Example of point cloud processing. (**A**) Original point cloud. (**B**) Processed point cloud. The main plane in the scene has been removed, and the segmentation of the remainder of the cloud has been executed.

pseudocode of the semantic phase is described in Algorithm 2. The input to the semantic phase algorithm is a point cloud O of a clustered object, and the outputs are the label l and the parameter vector $\boldsymbol{\vartheta}$ of the sphere or cylinder that best fits the object.

Pose estimation

RANSAC model has different parameters for each shape: center and radius for the sphere and center, radius, axis of the cylinder, and height for the cylinder. All coordinates are expressed in the camera frame; thus, a transformation in the robot frame is necessary to allow for manipulation. For this purpose, a marker has been placed at a fixed location on the base of the robot, and transformation matrices have been used to move from one coordinate frame to another.

Algorithm 2 Semantic phase

function (l, $\boldsymbol{\vartheta}$) = semanticPhase(PointCLoud$O$)
(i_s, $\boldsymbol{\vartheta}_s$) = RANSAC(O, *Sphere*)
(i_c, $\boldsymbol{\vartheta}_c$) = RANSAC(O, *Cylinder*)
if $i_c > i_s$ then
    l = *Cylinder*
    return (l, $\boldsymbol{\vartheta}_c$)
else
    l = *Sphere*
    return (l, $\boldsymbol{\vartheta}_s$)
end if

The following notations for the necessary transformation matrices can now be defined: $\mathbf{T}_o^c$, pose of the object in camera frame; $\mathbf{T}_c^m$, transformation between the camera and the marker; $\mathbf{T}_m^b$, constant transformation between the marker and the base frame; $\mathbf{T}_o^b$, transformation between the object and the base frame; $\mathbf{T}_o^c$ is provided by the object recognition script; $\mathbf{T}_c^b$ is acquired using a ROS wrapper for Alvar (*25*), an open source AR tag tracking library, whereas $\mathbf{T}_m^b$ is fixed and known.

For the composition property of transformation matrices (*19*), the desired matrix can be found as

$$\mathbf{T}_o^b = \mathbf{T}_m^b \mathbf{T}_c^m \mathbf{T}_o^c \tag{2}$$

Learning from demonstration

In a previous work (*26*), a dataset of grasps demonstrated by a human teacher was acquired through a motion capture suit. For each of the demonstrated grasp, the features of the object were defined as the input, whereas the values of both the synergy coefficients of the hand and the parameters of the arm were defined as the output. This dataset has been used to train two sets (one for the hand and one for the arm) of NN (*27*). Their architecture was experimentally chosen by trying different combinations of hidden layers and neurons and analyzing the corresponding performance in terms of mean squared error. We found that a feedforward structure with two hidden layers and 10 neurons for each is the most suitable choice for our particular application.

Reinforcement learning

The output of the NN module is then used to initialize the policy parameters of an RL loop. The idea is to explore the synergy subspace of the system locally to the initial values provided by imitation learning to improve the performances of the robot in a trial-and-error paradigm. The RL loop can be fundamentally divided in four main phases:

1) Sampling. N samples are extracted from two multivariate Gaussian distributions with means μ_{hand} and μ_{arm} given by the outputs of the NN and covariance matrices Σ_{hand} and Σ_{arm} chosen before the start of the loop. The choice of Σ_{hand} and Σ_{arm} determines the width of the exploration in the space of policy parameters. In this work, a simple exploration decay has been implemented to favor the exploitation of the acquired information over exploration in later stages of the learning process.

2) Planning. For each of the samples extracted in the previous phase, a trajectory in the Cartesian space is planned for both the hand and the arm. The equation mapping the synergy subspace to the Cartesian space for the hand is given by

$$\dot{\mathbf{x}} = \mathbf{J}_{hm}\dot{\mathbf{m}} = \mathbf{J}_h \mathbf{S}_s \dot{\boldsymbol{\sigma}} \tag{3}$$

where $\mathbf{J}_{hm}$ is the mechanical synergies Jacobian; $\dot{\mathbf{m}}$ and $\dot{\boldsymbol{\sigma}}$ are the vectors of motor and synergies velocities, respectively; $\mathbf{J}_h$ is the hand Jacobian; and $\mathbf{S}_s$ is the synergy matrix. The mapping between the reduced subspace for the arm and the Cartesian pose of the end effector is given by

$$\mathbf{p} = \bar{\mathbf{p}} + \mathbf{e}_p \alpha_p \tag{4}$$

$$\mathrm{e} = \bar{\mathrm{e}} + \mathbf{e}_{\mathrm{e}} \alpha_{\mathrm{e}} \tag{5}$$

where $\mathbf{p}$ is the position of the center of the wrist, $\boldsymbol{\epsilon}$ is the vector part of the quaternion, $\bar{\mathbf{p}}$ and $\bar{\mathbf{e}}$ are the mean values of $\mathbf{p}$ and $\boldsymbol{\epsilon}$ obtained from PCA, and $\mathbf{e}_p$ and $\mathbf{e}_{\boldsymbol{\epsilon}}$ are the first principal component for the position and the orientation, respectively. On the basis of these relationships, a rectilinear path from the home configuration of the wrist to the desired pose is planned. Once the desired pose has been reached, the fingers are closed according to the values of the synergy coefficients.

3) Execution. To execute the planned trajectories, the references have to be mapped in the configuration space of the system; in particular, we need to compute the target velocities of the joints that allow the robot to reproduce the desired motions. This has been done by applying a closed-loop inverse kinematic algorithm (*19*). In addition to the execution of the planned motions for the hand-arm system, an additional low-level reactive control has been implemented for the hand: The fingers close toward the centroid of the polygon with vertices at the center of the fingertips until the measured current reach a certain threshold.

4) Evaluation and reward. For each trial performed, the agent receives a reward function. As in (*28*), the scalar cost function $V(\boldsymbol{\sigma})$ is based on a force-closure quality index introduced in (*20*). Without going into details, an algorithm for optimal force distribution toward the improvement of force-closure property can be based on the minimization of a cost function $V(\delta\boldsymbol{\sigma})$ with respect to $\delta\boldsymbol{\sigma}$.

Let $\Omega_{i,j}^k \subset \mathbb{R}^h$ indicate the set of grasp variables $\boldsymbol{y}$ that satisfy the friction cone constraint with a (small, positive) margin k, where h is the dimension of the internal force subspace $\mathbf{E}$; let $\mathbf{w}$ indicate the object weight. The matrix $\mathbf{E}$ maps the controlled joint displacement into internal forces activated on the object. Underactuation reduce the subspace's dimension of controllable internal forces according to the mechanical and motor synergies; thus, according to the matrices, $\mathbf{E}_m$ equals $\mathbf{E}\mathbf{S}_m$ and $\mathbf{E}_s$ equals $\mathbf{E}\mathbf{S}_m\mathbf{S}_s$.

For the ith contact and the jth constraint, V is obtained as the summation of the terms $V(\mathbf{w}, \mathbf{y}) = \sum_i \sum_j V_{i,j}(\mathbf{w}, \mathbf{y})$ defined as

$$V_{i,j} = \begin{cases} (2\sigma_{i,j}^2(\mathbf{w}, \mathbf{y}))^{-1} & \mathbf{y} \in \Omega_{i,j}^k \\ a\sigma_{i,j}^2(\mathbf{w}, \mathbf{y}) + b\sigma_{i,j}(\mathbf{w}, \mathbf{y}) + c & \mathbf{y} \notin \Omega_{i,j}^k \end{cases} \tag{6}$$

where a, b, and c are constant positive parameters conditioned by properties imposed to V.

This cost function, detailed in (*20*), has a formulation suitable for practical implementation in manipulation planning. To be more specific, the function V has been adopted as a cost function indicating the quality of grasp, because the reciprocal of V reflects the distance of the grasp from violating the friction cone constraints and is obtained by formulating and solving the problem as a second-order cone programming.

In this work, the reward function $r(\boldsymbol{\sigma})$ has the following expression

$$r(\boldsymbol{\sigma}) = \beta V(\boldsymbol{\sigma}) + \phi \tag{7}$$

where $\beta = 10–6$ is a scaling factor and ϕ is

$$\phi = \begin{cases} 0 & \text{if grasp succeeds} \\ 10^4 & \text{if grasp fails} \end{cases} \tag{8}$$

The value $\phi = 10^4$ has been chosen higher than in the previous work to penalize more decisively the failed grasps and to avoid them in the following explorations. β and ϕ have been tuned experimentally with the aim that the cost function $V(\boldsymbol{\sigma})$ become meaningful when the grasp is successful.

Experimental results

We tested our algorithm on an experimental setup consisting of an SVH, a KUKA Lightweight Robot 4+, an Asus Xtion PRO LIVE sensor for point cloud acquisition, and a PC equipped with the ROS meta-operating system.

For each experiment, a different object is chosen and placed at a random position on a table in front of the robot. The sensor acquires a point cloud of the scene, and the features and pose of the object are estimated by the object recognition module, which publishes the information on a ROS topic that the NN node subscribes to. The initial values of the policy parameters are obtained by the NN and used to generate five samples for the RL loop; each of the corresponding trajectories is executed and a reward is computed on the basis of the force-closure cost function and on the result of a simple lifting test: Once the fingers are closed, the robot tries to lift the object and carry it at the home configuration. If the test fails, a high penalty is assigned to the trial, whereas no penalty is assigned for a successful test. The rewards associated to each trajectory are used to update the policy of the algorithm, assigning low probabilities to trajectories that performed poorly.

Three different objects were considered for these experiments: a tennis ball, a plastic strawberry, and a plastic bottle; the results of the experiments are shown in Figs. 3, 4, and 5. For each experiment, we report

1) A color-coded table, where the ith row is the ith update of the RL loop, and the jth column is the jth trial. Each entry is red if the corresponding grasp failed, and green if it was successful. Thus, five trials for each update led to a total number of 25 trials. For each tried grasp, the matching with the corresponding values on the graphs [images (B), (C), and (D)] can be found by reading the table from left to right and from the top to the bottom.

2) A plot of the evolution of the force-closure cost function.

3) A plot of the evolution of the hand synergies and arm scores.

We can see how the number of successful grasps significantly increases during the final updates in all the experiments, while the cost associated with the force-closure index converges to lower values. In particular, in the experiment with the tennis ball, we can see how the number of successful attempts is higher in the beginning of the experiment compared with the other objects. Because the ball can be accurately represented with a simple spherical shape, the values of the parameters obtained by imitation learning already provide a good approximation for synthesizing a stable grasp. When trying to execute harder grasps, the information acquired from human demonstration is not good enough to obtain stable grasps, and the RL loop is necessary to explore the parameter space and improve the initial grasps by learning from the interaction of the robot with the environment (Fig. 6).

The role of force-closure cost function

To appreciate the influence of the two terms of the reward function in the convergence of the algorithm, we ran experiments (using the value $\phi = 103$) for (i) the whole hand-arm system and (ii) only the hand. The conclusion is that, for the hand-arm system, we have to increase ϕ, as done in this work ($\phi = 104$), to have a faster convergence (almost halving the trials). By increasing ϕ, the pose of the arm leading to a correct grasp is learned faster. The experiments confirm that the learning capabilities and the convergence of the algorithm are significantly affected by the parameter ϕ, whereas the force-closure cost function affects mainly the hand configuration. To appreciate the influence of the force-closure cost function, we ran the algorithm with and without the adoption of this cost only for the hand. The conclusion is that the force-closure cost function does not influence the success of the grasp but the stability of the grasp. The number of trials to get the convergence of the algorithm does not change, but the force-closure cost changes significantly. In Table 1, a quantitative comparison of final grasps, obtained for different objects with and without the adoption of the quality index to generate the action, is reported.

Evaluation of the algorithm in different initial conditions

To validate the algorithm in different initial conditions, we compared the results using force-closure cost function after 25 trials. We tested the algorithm in three different conditions with reference to the parameter initialization: (i) The initialization of the policy is provided by comparing the object to be grasped with examples contained in a reference table (including a limited number of object/grasp pairs) and by choosing parameters related to the closest one; (ii) the shape and dimension of the object are known and given as input to the NN, and the initialization of the policy is the output of the NN; and (iii) the object is unknown, and the vision system is in charge of extracting the information to be given as input to the NN. In case (i), the algorithm fails and does not converge. In (ii) and (iii), the convergence is ensured but the quality of the grasp is influenced by the uncertainty introduced by the perception. The results are better for (ii) in terms of quality of the grasp. For (iii), the cost function is greater of an order of magnitude, regardless of object and grasp type. Therefore, we can state that the algorithm is robust to uncertainties on perception to a certain extent [see cases (ii) and (iii)], but if the initialization is too far from the correct value, the algorithm does not converge [see case (i)]. Another consideration is that, if the system is constituted only by the hand, in case (i) the algorithm converges. This means that the robustness of the algorithm with respect to the initial conditions decreases as the complexity of the system increases.

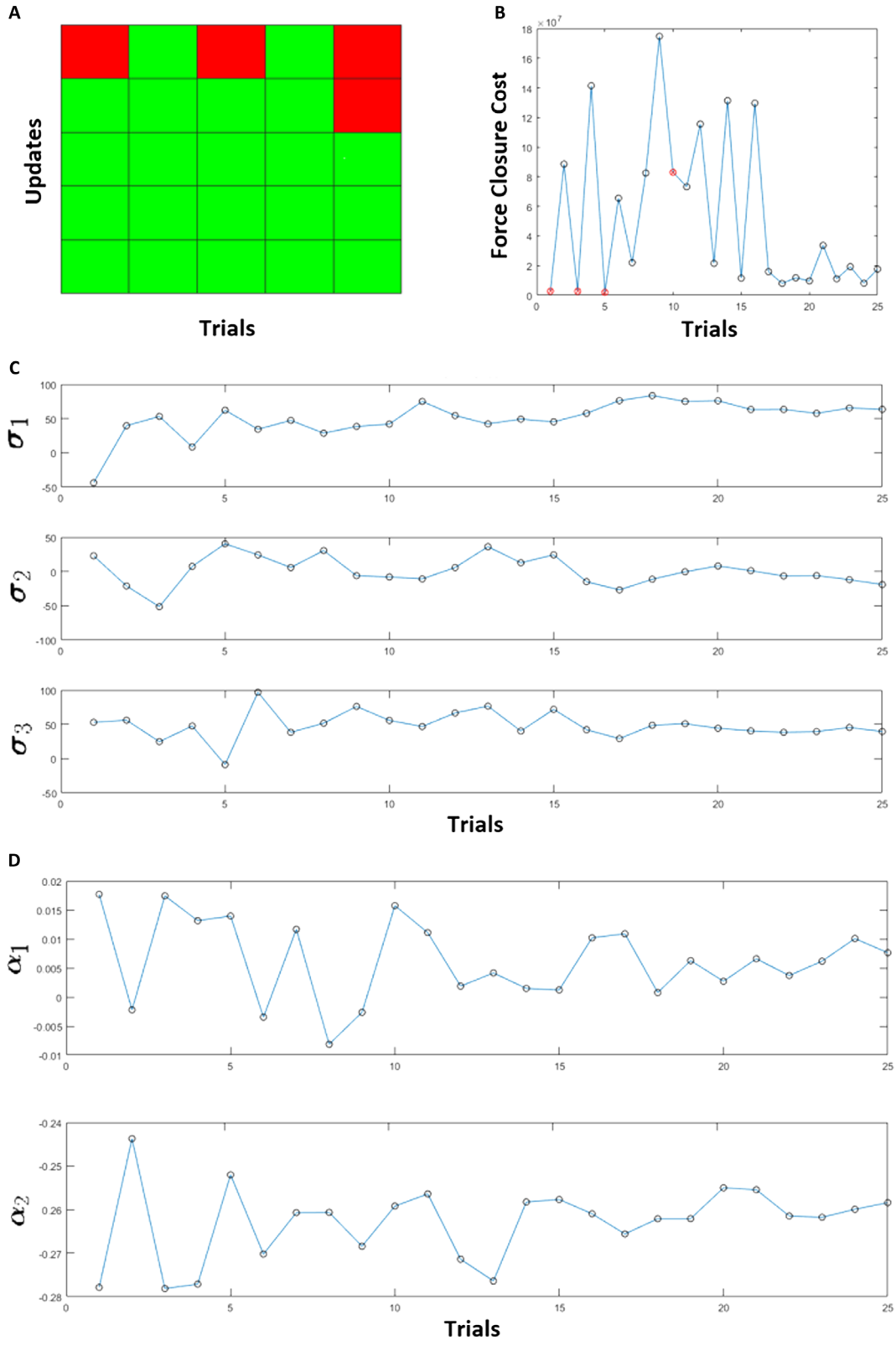

Fig. 3. Experiment on a tennis ball. (**A**) Grasp success table. (**B**) Force-closure cost function. (**C**) Synergy coefficients. (**D**) Arm scores.

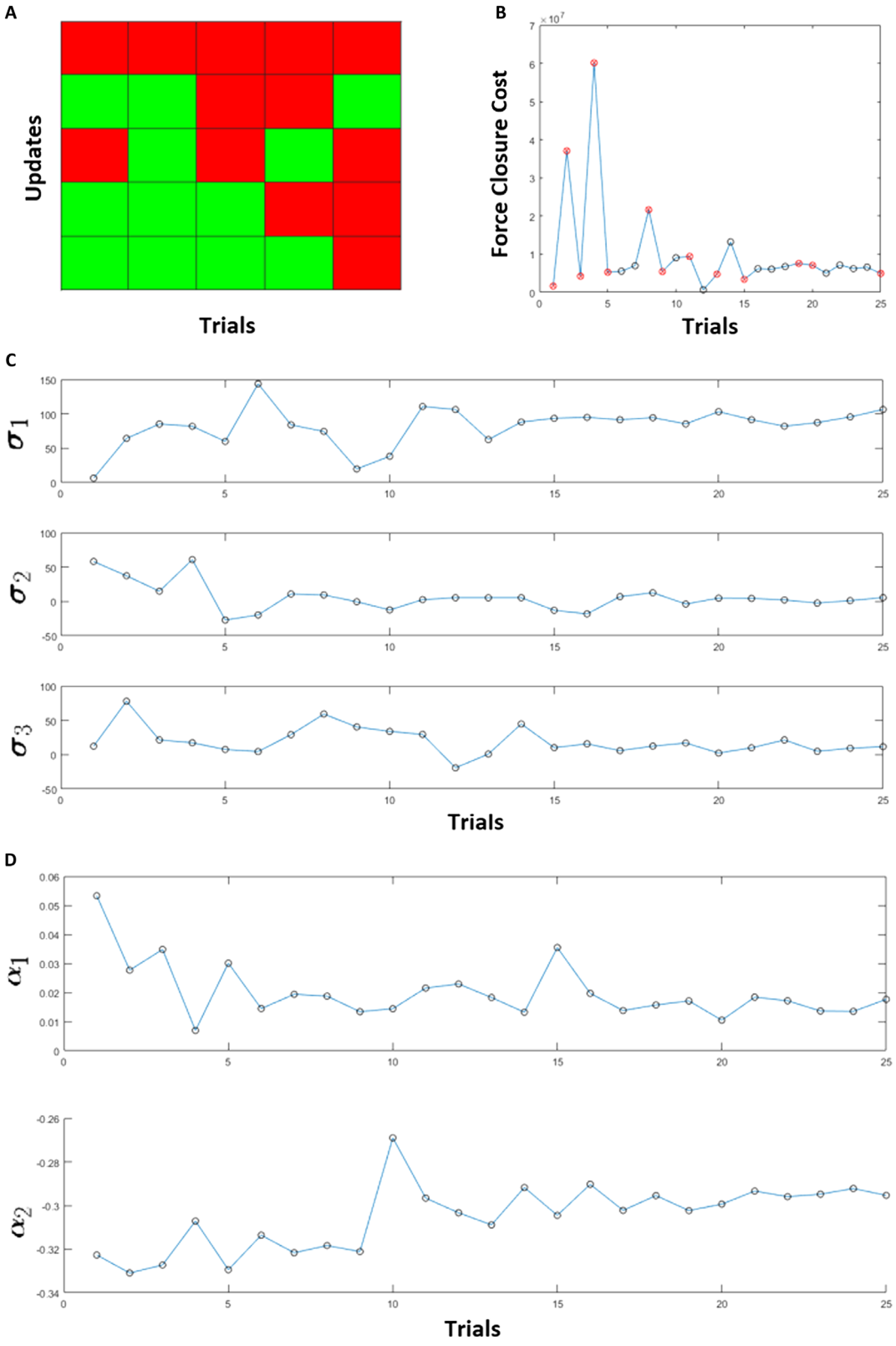

Fig. 4. Experiment on a plastic strawberry. (**A**) Grasp success table. (**B**) Force-closure cost function. (**C**) Synergy coefficients. (**D**) Arm scores.

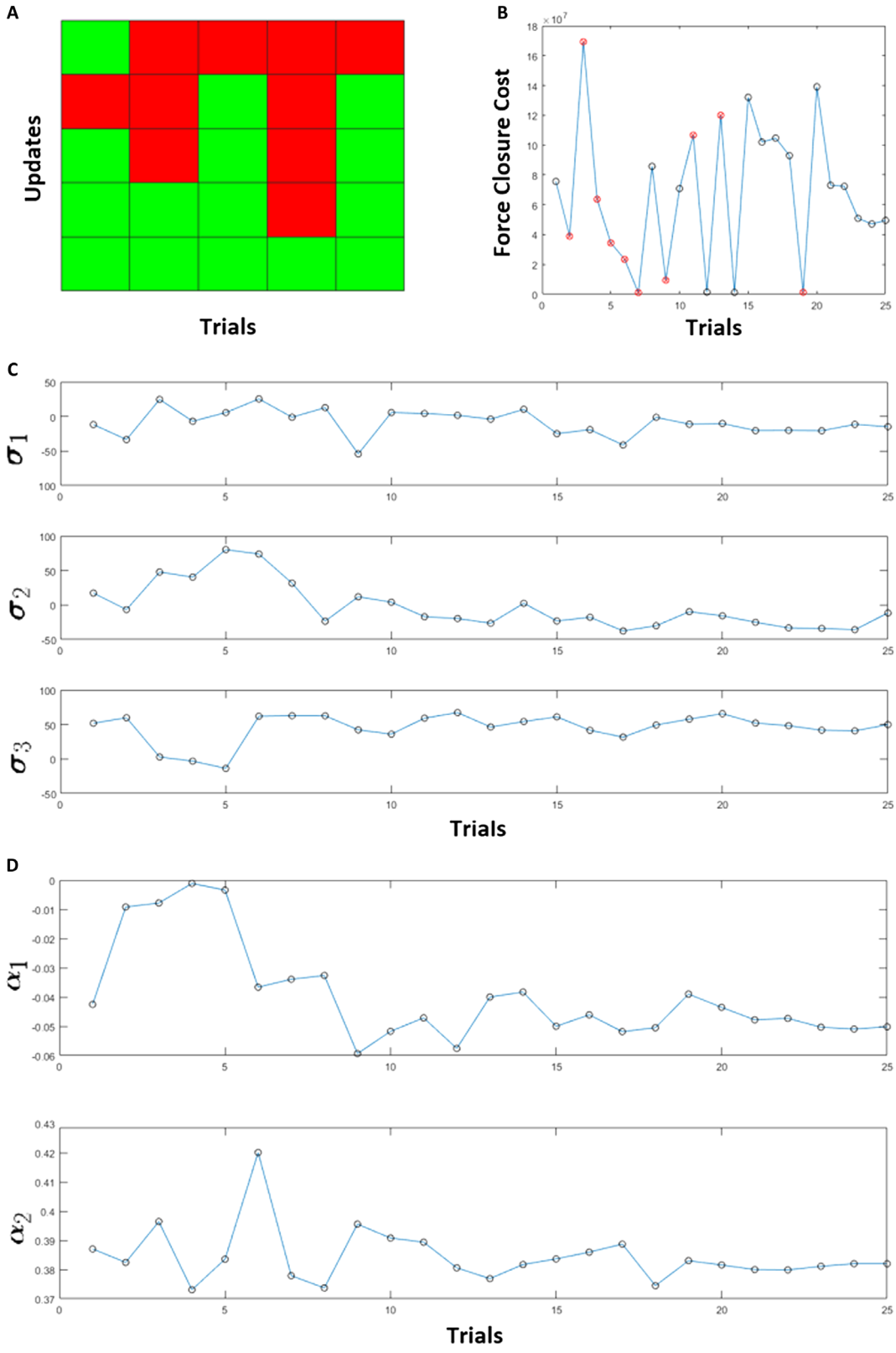

Fig. 5. Experiment on a plastic bottle. (**A**) Grasp success table. (**B**) Force-closure cost function. (**C**) Synergy coefficients. (**D**) Arm scores.

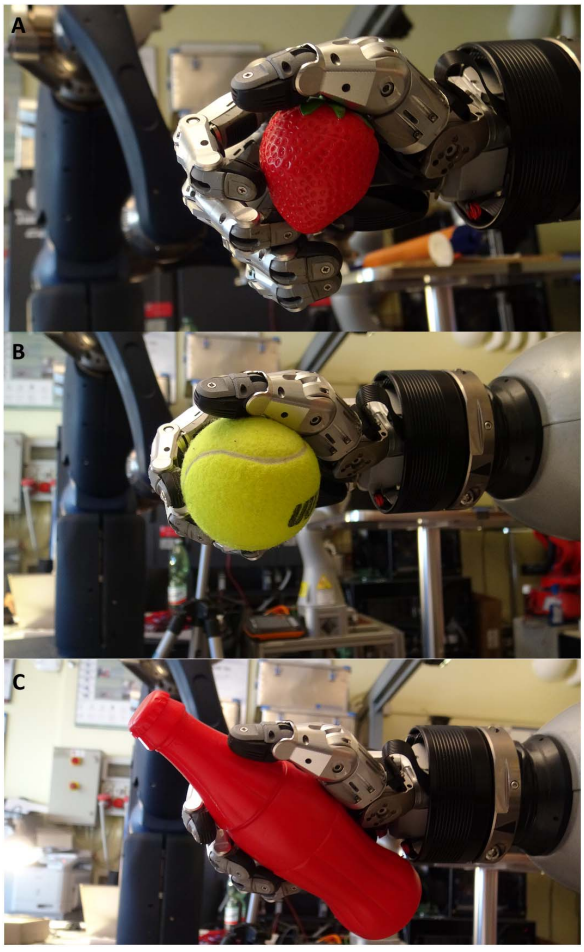

Fig. 6. Successful grasps of the tennis ball, strawberry, and plastic bottle at the end of the learning process.

Table 1. Comparison of grasps obtained with and without force-closure cost function.

	Force-closure cost value	
Grasp	**Cost not used**	**Cost used**
Bipodal	1.9×10^7	8.5×10^5
Tripodal	2.5×10^7	3.1×10^6
Sphere five finger	2.7×10^8	4.6×10^7
Cylinder five finger	1.9×10^8	1.4×10^7

DISCUSSION

In this work, the problem of grasping novel objects with a robotic hand-arm system has been considered; in particular, an algorithm for learning stable grasps of unknown objects has been developed based on a simple shape classification and on the extraction of some of the associated features.

Building upon previous researches on the topic, the results presented in this work confirmed that the combination of learning from demonstration and RL can be an interesting solution for complex tasks, such as grasping with anthropomorphic hands, providing the robot with a good base to start the learning process, and allowing it to improve its abilities through trial and error. The experiments carried out on our setup provided encouraging results, showing that this framework is suitable for synthesizing stable grasps of different objects and how it can potentially be improved over time, with reference to some quality metrics.

The algorithm has been tested in three different conditions with reference to parameter initialization. The experimental results demonstrate that the system is robust to uncertainties on perception of the environment to a certain extent. If the initialization is "wrong"—namely, too far from a good value—the algorithm does not converge. The initialization is obtained using an NN trained by human grasping samples. We have verified that, if NN are not used to initialize the policy, the algorithm does not converge. In other words, a complex system with high DoFs would never converge without proper initialization of the policy parameters. In this context, synergies will help a smart choice of the policy, whereas a synergy-based supervised learning algorithm allows a suitable initialization of the policy parameters. One of the main limitations of the proposed algorithm is in the very simple object recognition algorithm, which could be improved by considering additional shape templates, as well as decomposing complex objects into simpler parts that can be associated with basic shapes in a satisfying way. Furthermore, an automatic way for testing the success of a grasp, without relying on a supervisor, could be implemented: For example, this could be done by tracking the object position with the camera. The use of different cost functions in the RL loop could also be investigated, for example, developing some metric for the quality of a grasp derived from visual information or considering costs that depend on the particular task that we want to accomplish.

The proposed algorithm could also be integrated in a wider framework, where different actions are selected on an higher hierarchical level in a task-oriented fashion, guiding the learning process toward grasps that better suit a particular application. Future comparisons with different learning algorithms are referred in particular to RL policy search methods like PILCO (*14*) and PI-REM (*15*). It is not easy to integrate these algorithms in such a complex framework, so the comparison with other methods is not trivial. In effect, different aspects need to be considered and carefully evaluated.

MATERIALS AND METHODS

Hardware

Schunk five-finger hand

The SVH is a very compact, compliant under-actuated hand, which closely resembles the structure and the appearance of the human hand. One of the key features of this hand is that all controllers, regulators, and power electronics are fully integrated in the wrist. The hand has a total of nine drives that move the 20 DoFs of the hand, owing to the mechanical coupling between joints inspired by the way human fingers usually move together. Technical details and specifics for this device can be found in (*29*). To communicate with the hand, a ROS wrapper for the SVH driver, available at (*30*), was used.

KUKA Lightweight Robot 4+

The KUKA Lightweight Robot 4+ is a very efficient and portable robot weighing only 16 kg, with a payload of 7 kg; motors, gears, and sensors are accommodated inside an aluminum housing, as well as the necessary control and power electronics. The redundant DoF provides the arm with additional flexibility and can be used in different ways based on the task at hand: It can be exploited to avoid obstacle, to increase the manipulability,

or in general to obtain more favorable motions for the desired task. Technical details and specifics for this device can be found in (*31*).

The control of the KUKA arm was implemented by taking advantage of the FRI library, available at (*32*), which provides a simple user interface for communicating with the robot. The library runs on a remote PC that is connected with the KRC (KUKA Robot Controller) via Ethernet.

Asus Xtion PRO LIVE

Asus Xtion PRO LIVE (*33*) is an RGB-D sensor consisting of an RGB camera, an infrared emitter, and a CMOS (complementary metal-oxide semiconductor) sensor. The Asus Xtion PRO LIVE is a compact, plug-and-play device, with a resolution of 640 pixels by 480 pixels for the depth stream and 1280 pixels by 1024 pixels for the color stream, with a depth range of approximately 0.6 to 3.5 m and a field of view of 58° H, 45° V, 70° D (horizontal, vertical, diagonal).

Software

ROS

ROS is a meta-operating system providing many functionalities, such as low-level device control, hardware abstraction, and package management. ROS is based on a peer-to-peer network of processes, called nodes, which communicates using the ROS communication infrastructure. The ROS framework is based on an asynchronous publish/subscribe system, and it provides many different functionalities and is easily accessible either by GUI or by command line (*34*).

PCL library

The PCL library provides implementations of many state-of-the-art algorithms for point cloud processing, such as filtering, segmentation, and model fitting (*35*).

Methods

Neural networks

The NN used in this work were built and trained using the Neural Networks Toolbox available in MATLAB. All of the networks have the same architecture, chosen experimentally and consisting of two hidden layers with five neurons each. The available dataset was randomly divided into training, validation, and test subsets comprising 70, 15, and 15%, respectively. The weights were randomly initialized using the Nguyen-Widrow rule, and the Levenberg-Marquadt algorithm was used for training.

Policy improvement with path integrals (PI2)

The implemented solution for the RL algorithm loop is based on the episodic PI2 formulation originally proposed in (*36*). Each trajectory was assigned a probability in the following way

$$P(\tau_i) = \frac{e^{-\frac{1}{\lambda}S_i}}{\sum_{i=1}^{n} e^{-\frac{1}{\lambda}S_i}} \tag{9}$$

where $S_i = S(\tau_i)$ is the cost of trajectory τ_i.

These probabilities are used to update the mean value of the parameters for the next iteration

$$\mu_{\theta k} = \mu_{\theta k-1} + \sum_{i=1}^{n} P(\tau_i)(\mu_{\theta i-1} - \theta_i) \tag{10}$$

where $\mu_{\theta 0}$ is the initial mean value of the policy parameters, $\Sigma_{\theta 0}$ is the initial covariance matrix, and $\theta_i \sim N(\mu_{\theta 0}, \Sigma_{\theta 0})$ is the ith policy parameter sample.

Because we are implementing an exploration decay, Σ_θ is also updated

$$\boldsymbol{\Sigma}_{\theta k} = \gamma \boldsymbol{\Sigma}_{\theta k-1} \tag{11}$$

The values of the parameters used for the experiments are $k = 5$, number of updates; $n = 5$, number of trials per update; $\lambda_{\text{hand}} = 1000$, hand synergies variance; $\lambda_{\text{arm}} = 0.0002$, arm coefficients variance; and $\gamma = 0.7$, exploration decay.

SUPPLEMENTARY MATERIALS

robotics.sciencemag.org/cgi/content/full/4/26/eaao4900/DC1
MATLAB code (.zip)

REFERENCES AND NOTES

1. M. Santello, M. Flanders, J. F. Soechting, Postural hand synergies for tool use. *J. Neurosci.* **18**, 10105–10115 (1998).
2. M. T. Ciocarlie, P. K. Allen, Hand posture subspaces for dexterous robotic grasping. *Int. J. Rob. Res.* **28**, 851–867 (2009).
3. M. Ciocarlie, C. Goldfeder, P. Allen, Dimensionality reduction for hand-independent dexterous robotic grasping, in *IEEE/RSJ International Conference on Intelligent Robots and Systems* (IEEE, 2007), pp. 3270–3275.
4. M. Gabiccini, A. Bicchi, D. Prattichizzo, M. Malvezzi, On the role of hand synergies in the optimal choice of grasping forces. *Auton. Robots* **31**, 235 (2011).
5. A. Sahbani, S. El-Khoury, P. Bidaud, An overview of 3d object grasp synthesis algorithms. *Rob. Auton. Syst.* **60**, 326–336 (2012).
6. K. B. Shimoga, A. A. Goldenberg, Soft robotic fingertips: Part I: A comparison of construction materials. *Int. J. Rob. Res.* **15**, 320–334 (1996).
7. J. Bohg, A. Morales, T. Asfour, D. Kragic, Data-driven grasp synthesis—A survey. *IEEE Trans. Robot.* **30**, 289–309 (2014).
8. A. T. Miller, S. Knoop, H. I. Christensen, P. K. Allen, Automatic grasp planning using shape primitives, in *IEEE International Conference on Robotics and Automation* (IEEE, 2003), pp. 1824–1829.
9. K. Huebner, D. Kragic, Grasping by parts: Robot grasp generation from 3d box primitives, in *Proceedings of the 4th International Conference on Cognitive Systems* (CogSys, 2010).
10. R. Pelossof, A. Miller, P. Allen, T. Jebara, An SVM learning approach to robotic grasping, in *IEEE International Conference on Robotics and Automation* (IEEE, 2004), pp. 3512–3518.
11. A. Saxena, J. Driemeyer, A. Y. Ng, Robotic grasping of novel objects using vision. *Int. J. Rob. Res.* **27**, 157–173 (2008).
12. M. Stark, P. Lies, M. Zillich, J. Wyatt, B. Schiele, Functional object class detection based on learned affordance cues, in *International Conference on Computer Vision Systems* (Springer, 2008), pp. 435–444.
13. M. A. Roa, R. Suárez, Grasp quality measures: Review and performance. *Auton. Robots* **38**, 65–88 (2015).
14. M. Deisenroth, C. E. Rasmussen, PILCO: A model-based and data efficient approach to policy search, in *Proceedings of the 28th International Conference on Machine Learning* (ACM, 2011), pp. 465–472.
15. M. Saveriano, Y. Yin, P. Falco, D. Lee, Data-efficient control policy search using residual dynamics learning, in *IEEE/RSJ International Conference on Intelligent Robots and* Systems (IEEE, 2017), pp. 4709–4715.
16. F. Ficuciello, P. Falco, S. Calinon, A brief survey on the role of dimensionality reduction in manipulation learning and control. *IEEE Robot. Autom. Lett.* **3**, 2608–2615 (2018).
17. F. Ficuciello, A. Federico, V. Lippiello, B. Siciliano, Synergies evaluation of the SCHUNK s5fh for grasping control, in *Advances in Robot Kinematics 2016*, J. Lenarčič, J. P. Merlet, Eds. (Springer, 2018), pp. 225–233.
18. F. Ficuciello, Hand-arm autonomous grasping: Synergistic motions to enhance the learning process. *Intell. Serv. Rob.* **2018**, 1–9 (2018).
19. B. Siciliano, L. Sciavicco, L. Villani, G. Oriolo, Robotics: Modelling, planning and control, in *Advanced Textbooks in Control and Signal Processing Series*, M. J. Grimble, L. Bushnell, Eds. (Springer, 2009).
20. A. Bicchi, On the closure properties of robotic grasping. *Int. J. Rob. Res.* **14**, 319–334 (1995).
21. J. Peters, S. M. Vijayakumar, S. Schaal, Reinforcement learning for humanoid robotics, in *IEEE-RAS International Conference on Humanoid Robots* (IEEE, 2003), pp. 1–20.
22. M. P. Deisenroth, G. Neumann, J. Peters, A survey on policy search for robotics. *Found. Trends Rob.* **2**, 1–142 (2013).

23. F. Ficuciello, G. Palli, C. Melchiorri, B. Siciliano, Postural synergies of the UB hand IV for human-like grasping. *Rob. Auton. Syst.* **62**, 515–527 (2014).
24. G. Palli, C. Melchiorri, G. Vassura, L. Moriello, G. Berselli, A. Cavallo, G. De Maria, C. Natale, S. Pirozzi, C. May, F. Ficuciello, B. Siciliano, The DEXMART hand: Mechatronic design and experimental evaluation of synergy-based control for human-like grasping. *Int. J. Rob. Res.* **33**, 799–824 (2014).
25. ROS, ROS wrapper for the Alvar AR tag tracking library; http://wiki.ros.org/ar_track_alvar.
26. F. Ficuciello, D. Zaccara, B. Siciliano, Learning grasps in a synergy-based framework, in *International Symposium on Experimental Robotics*, D. Kulić, Y. Nakamura, O. Khatib, G. Venture, Eds. (Springer, 2016), pp. 125–135.
27. M. P. Perrone, L. N. Cooper, *How We Learn; How We Remember: Toward an Understanding of Brain and Neural Systems: Selected Papers of Leon N Cooper* (World Scientific, 1995), pp. 342–358.
28. F. Ficuciello, D. Zaccara, B. Siciliano, Synergy-based policy improvement with path integrals for anthropomorphic hands, in *IEEE/RSJ International Conference on Intelligent Robots and Systems* (IEEE, 2016), pp. 1940–1945.
29. Schunk, SVH documentation; https://schunk.com/it_en/ gripping-systems/highlights/svh/.
30. Schunk, SVH driver ROS wrapper; http://wiki.ros.org/schunk_svh_driver.
31. KUKA, Lightweight Robot 4+ documentation; https://kukakore.com/wp-content/uploads/2012/07/KUKA_LBR4plus_ENLISCH.pdf.
32. FRILibrary, Fast research interface library; https://cs.stanford.edu/people/ tkr/fri/html/.
33. Asus, Xtion PRO LIVE; www.asus.com/3D-Sensor/Xtion_PRO_ LIVE/.
34. ROS, Robot operating system; www.ros.org/.
35. R. B. Rusu, S. Cousins, 3D is here: Point Cloud Library (PCL), in *IEEE International Conference on Robotics and Automation* (IEEE, 2011), pp. 1–4.
36. F. Stulp, O. Sigaud, Path integral policy improvement with covariance matrix adaptation. arXiv:1206.4621 [cs.LG] (18 June 2012).

Funding: The research leading to these results has been partially supported by the RoDyMan project (FP7/2007-2013) under ERC AdG-320992 and partially by MUSHA National Italian project. This research was carried out in the frame of Programme STAR, financially supported by UNINA and Compagnia di San Paolo. **Author contributions:** F.F., P.F., A.M., and B.S. conceptualized the project and developed the methodology. All authors contributed to experimental validation, analyses, data visualization, and writing and revising the manuscript. **Competing interests:** The authors declare that they have no competing interests. **Data and materials availability:** All data and code needed to evaluate the conclusions are available in the paper or the Supplementary Materials.

Submitted 4 September 2018
Accepted 11 January 2019
Published 30 January 2019
10.1126/scirobotics.aao4900

Citation: F. Ficuciello, A. Migliozzi, G. Laudante, P. Falco, B. Siciliano, Vision-based grasp learning of an anthropomorphic hand-arm system in a synergy-based control framework. *Sci. Robot.* **4**, eaao4900 (2019).

Science Robotics

Vision-based grasp learning of an anthropomorphic hand-arm system in a synergy-based control framework

F. Ficuciello, A. Migliozzi, G. Laudante, P. Falco, and B. Siciliano

Sci. Robot. **4** (26), eaao4900. DOI: 10.1126/scirobotics.aao4900

Science Robotics (ISSN 2470-9476) is published by the American Association for the Advancement of Science. 1200 New York Avenue NW, Washington, DC 20005. The title *Science Robotics* is a registered trademark of AAAS.