Advanced Grasping with the Pisa/IIT SoftHand

Manuel Bonilla[1], Cosimo Della Santina[2], Alessio Rocchi[1], Emanuele Luberto[2],
Gaspare Santaera[1], Edoardo Farnioli[1], Cristina Piazza[2], Fabio Bonomo[3],
Alberto Brando[3], Alessandro Raugi[3], Manuel G. Catalano[1],
Matteo Bianchi[2,4(✉)], Manolo Garabini[2], Giorgio Grioli[1], and Antonio Bicchi[1,2]

[1] Soft Robotics for Human Cooperation and Rehabilitation,
Istituto Italiano di Tecnologia, Genoa, Italy
[2] Centro E. Piaggio, Universitá di Pisa, Pisa, Italy
matteo.bianchi@centropiaggio.unipi.it
[3] QB Robotics, Pisa, Italy
[4] Department of Information Engineering, Universitá di Pisa, Pisa, Italy

Abstract. This chapter presents the hardware, software and overall strategy used by the team UNIPI-IIT-QB to participate to the Robotic Grasping and Manipulation Competition. It relies on the PISA/IIT Soft-Hand, which is underactuated soft robotic hand that can adapt to the grasped object shape and is compliant with the environment. It was used for the hand-in-hand and for the simulation tracks, where the team reached first and third places respectively.

Keywords: Grasping · Grasp simulation · Grasp planning

1 Introduction

Despite the continuous advancements in the field, the design and realization of dexterous robotic hands is still a big challenge in robotics. Over the years, several hand designs were proposed that try to match the level of dexterity of the human hand. These hands typically resort to very complex and articulated designs which cleverly integrate many actuators, sensors and joints trying to get close to the richness and complexity of the sensory and motor functions of the human hand, nevertheless this still remains a distant goal on the horizon (e.g. [19]).

An alternative and promising trend in robot hand design is simplification, indeed trying to encompass some of the limitations of overly complex mechanical system and removing some of its components can introduce more advantages than drawbacks, if done with the right criteria. One of the most interesting simplification criteria is that of embedding part of the control intelligence in the physical structure of the system itself, the main tool to achieve this goal is under-actuation [4]. Thanks to under-actuation, engineers can reduce the number of degrees of actuation (DOAs) of robotic hands and thus simplify their design.

While in a traditional robot hand a reduction of DOAs would imply less DOFs and, in turn, a considerable reduction of shape adaptation capabilities, the theory of under-actuation provides principles to design hands that retain a large number of DOFs and, by consequence, adaptability. To achieve this, under-actuation resorts to differential transmissions, realized in various forms, e.g. gears [26], tendons and pulleys [23].

Another idea, coming from the field of motor control [2], recently caused a great interest in the robotic hands field: postural synergies. Synergies can be defined [27] as a set of variables that show correlated changes in time. Postural synergies can be seen as a basis of the subspace of the effective human hand configurations among all those made possible by the kinematics of the body. Moreover, synergies can be ordered in terms of the statistical variance of the total amount of motion that they explain. This makes them a successful tool for artificial hand analysis, control and design simplification [3]. Notorious examples are [6,10].

The basic concept of postural synergies later evolved both on the human motor control and robotic sides. In particular the *Soft synergies* theory sprouted up [3], which assumes that synergies exist on an *ideal* domain, and define a virtual reference movement, toward which the physical system is elastically attracted. The manipulated environment, in turn, opposes to the ideal hand motion through its own compliance. The two actions concur to the generation of an equilibrium (e.g. [37,38]). Based upon the theory of soft synergies, *adaptive synergies* [20] integrated synergies with under-actuation, yielding a simple implementation of the former, with a series of considerable advantages, such as control and design simplification.

The Pisa/IIT SoftHand [9] is a recent outcome of this research (see Fig. 1): it implements one soft synergy, actuated with a transmission system that uses one tendon, pulleys, and one motor. The SoftHand demonstrated excellent grasping skills, in many different situations, combined with robustness and a simple control interface[1].

2   Organization

The rest of this chapter is organized as follows. Section 3 briefly introduces the Pisa/IIT SoftHand. In Sect. 4 we state the problem arising from the use of soft hands for grasping. Then, Sects. 5 and 6 present the simulation tool and grasping strategy used during the challenge. Section 7 presents the strategy of the PISA-IIT-QB team to compete in the grasping challenge in both hand-in-hand and simulation tracks. Finally Sect. 8 presents our results and conclusions.

3   The Pisa/IIT Softhand

The Pisa/IIT SoftHand was designed according to few specifications. On the functional side, requirements are to grasp as wide a variety of objects and tools

---

[1] Videos available at https://goo.gl/8zYDWs.

as possible, among those commonly used by humans in everyday tasks. The hand should be primarily able to effect a whole hand grasp of tools, properly and strongly enough to operate them under arm and wrist control, but also be able to achieve tip grasps. No in-hand dexterous manipulation is required for this prototype. For further evolutions in this sense please refer to [14].

The main nonfunctional requirements were resilience against force, overexertion and impacts, and safety in interactions with humans. The hand was designed to be lightweight and self-contained, to avoid encumbering the forearm and wrist with motors, batteries and cabling, along with cost effectiveness.

To meet the first functional requirement, the hand was designed anthropomorphically, with 19 DOFs arranged in four fingers and an opposable thumb, Fig. 1(a). To maximize simplicity and usability, however, the hand uses only one actuator. The hand assembly design is shown in Fig. 1(b). Each finger has four phalanges, while the thumb has three.
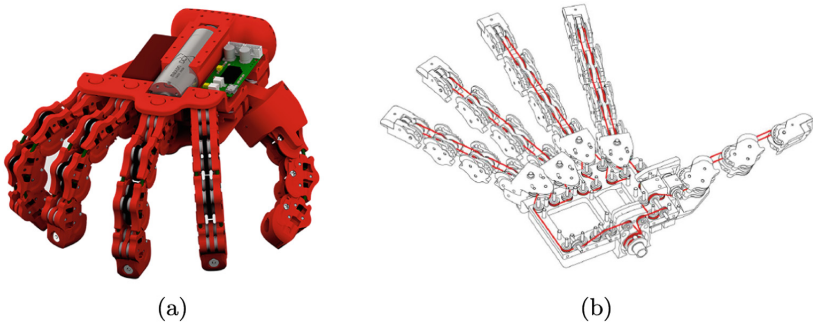

(a) (b)

Fig. 1. Pisa/IIT SostHand (a) is an anthropomorphic soft hand, with 19 DoFs actuated through a single tendon (b).

In rest position, with fingers stretched out and at a relative angle of about $15°$ in the dorsal plane, the hand spans approximately $230\,\text{mm}$ from thumb to little finger tip, is $235\,\text{mm}$ long from the wrist basis to the middle finger tip and has $40\,\text{mm}$ maximum thickness at the palm. The requirement on power grasp implies that the hand is able to generate a high enough grasping force, and to distribute it evenly through all contacts, be them at the fingertips, the inner phalanges, or the palm.

We adopted a non-conventional "soft robotics" design of the mechanics of the hand composed of rolling joints and elastic ligaments at a very low cost. Rolling contact articulations replaced standard revolute joints. Our design is inspired to a class of joints known as COmpliant Rolling-contact Elements (CORE) [8]. Among these, the Hillberry design of a rolling joint [22] was a source of inspiration to our design. A Hillberry joint consists of a pair of cylinders in rolling contact on each other, held together by metallic bands, which wrap around the cylinders on opposite sides as schematically shown in Fig. 2(b).
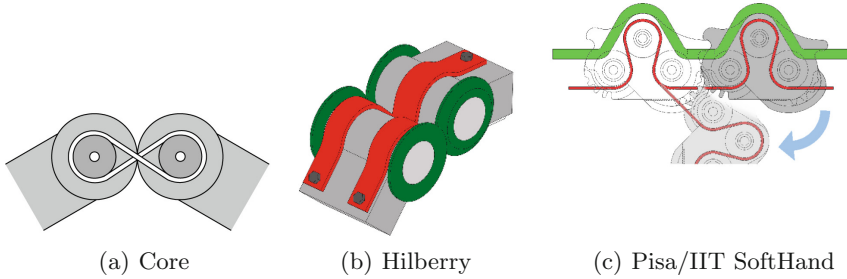
(a) Core   (b) Hilberry   (c) Pisa/IIT SoftHand

Fig. 2. Rolling contact articulations can be employed instead of standard revolute joints to obtain robust human-like articulations.

Figure 2(c) shows how we used CORE joints in the design of the Pisa/IIT SoftHand. In particular, we adopted CORE joints for all the interphalangeal, flexion/extension articulations. Conversely, conventional revolute joints was used for metacarpophalangeal, abduction/adduction articulations.

W.r.t. classic Hillberry joints, metallic bands were replaced with elastic ligaments fixed across the joint with an offset in the dorsal direction. Suitable pretensioning of the ligaments, together with a carefully designed profile of the two cylinders, introduces a desirable passive stability behavior, with an attractive equilibrium at the rest configuration (when fingers are stretched).

These features are particularly important for the system to behave softly and safely in contact, and to recover from force overexertion, due e.g. to impacts or jamming of the hand, making the hand automatically return to its correct assembly configuration. The joint can withstand severe disarticulations and violent impacts (Fig. 3).

Actuation of the hand is effected through a single Dyneema tendon routed through all joints using passive anti-derailment pulleys. The tendon action flexes and adducts fingers and thumb, counteracting the elastic force of ligaments, and implementing adaptive underactuation without the need for differential gears.

According to the design approach proposed in [20], the motor actuates the adaptive synergy as derived from a human postural database [36]. The mechanical implementation of the first soft synergy through shape-adaptive underactuation was obtained via numerical optimization of the pulley radii and joint stiffness (Fig. 4).

4   Problem Definition

Being actuated through just one motor, the Pisa/IIT SoftHand free closure movements is limited to just one direction. However, its overall compliance conveys the natural production of complex behaviors through interactions with the environment and the object to be grasped. This fact makes the Pisa/IIT SoftHand really intuitive to use by humans because they use it in the same way they would use their own hands. However, to make a robot use Soft Hands in the same way

that humans do is challenging. The main problem comes from the difficulty of giving robots the same level of intelligence and experience humans have.

It is the authors opinion that the reduced dimensionality of the actuation space, combined with the human inspiration, make this under-actuated and soft hand a valuable candidate not only for use on autonomous robotic manipulators (as e.g. [1]) but also for applications where a human user is active part of the planning and control loop, spanning from tele-operation, to prosthetics, to human grasp studies and rehabilitation robotics. Preliminary results in this sense with Pisa/IIT SoftHand are provided in [7, 18].

While direct control of the Pisa/IIT SoftHand is an easy problem, automatic motion planning is not, since it is difficult to model human-robot interactions. Classical motion planning algorithms are designed to place specific parts of the hand (mainly the fingertips) on specific parts on the object to grasp it [29]. This is not possible with Soft Hands since it is not possible to move each joint of the hand independently, introducing a paradigm shift in motion planning for robotic hands. Recently in [5] we introduced what we call "daring grasping", it consist in exploiting the environment to help in the grasping process. In such paper we discussed the problem of grasping with Soft Hands, we proposed the idea of taking advantages of the interactions of the robotic hands with the environment to shape the hand while it is grasping. In order to plan motion for Soft Hands we need to model environment interactions which is not an easy task.

In this chapter we use the PISA/IIT Soft Hand together with the same idea than in [5] to grasp objects but we use a more efficient simulator and grasping algorithm.
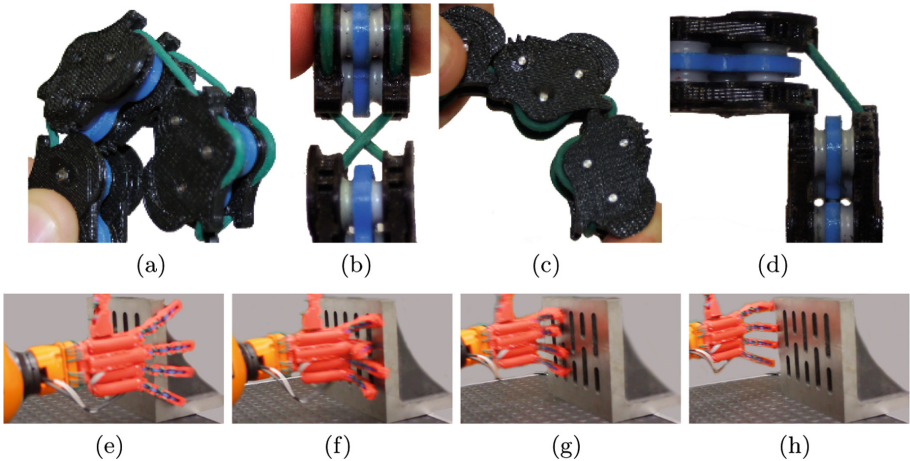

(a)   (b)   (c)   (d)

(e)   (f)   (g)   (h)

Fig. 3. The Pisa/IIT SoftHand joints can withstand severe force overexertion in all directions, automatically returning to the correct assembly configuration
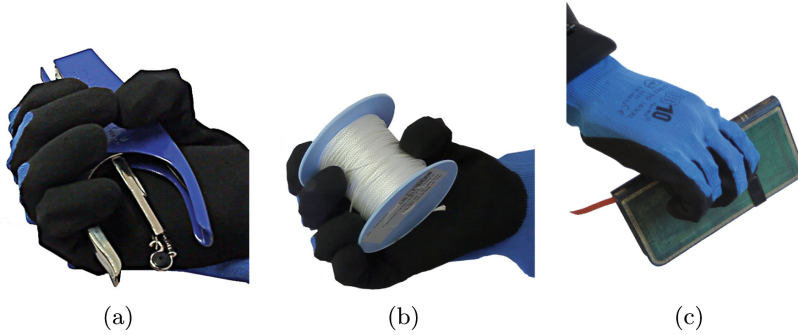
(a)   (b)   (c)

Fig. 4. Some experimental grasps performed with the Pisa/IIT hand, with the object placed in the hand by a human operator.

5   A Generic Simulator for Compliant Hands

Klampt generic compliant underactuated hand emulator (CUHE) has been used in order to simulate the Pisa/IIT SoftHand [9]. The emulator makes use of the adaptive synergy concept which can be generalizable to a wide array of under-actuated compliant hands, other than the SoftHand for which it has been firstly implemented. Example of compliant underactuated hands include the iHY [30], Reflex Hand [31], Pisa/IIT SoftHand, Robotiq 3-Fingered Gripper [32], RBO [12] and RBO2 [13] and Yale Hand [28].

Model-based predictions of motions and forces can offer powerful insights for grasp planning [24,25], and are an invaluable tool in testing high level grasp controllers before application to physical hardware. Simulation tools will help predict outcomes of controllers that make deliberate contact with the environment [15] as well as with the object. They may also be useful for computational exploration of gripper designs as well as control strategies, such as in reinforcement learning.

Motivated by the increasing interest in CUHs and recent advances in simulation software [34], and exploiting the framework that extends the concept of soft synergies into that of adaptive synergies [20] to model robotic hands with a known transmission distribution matrix and linear joint stiffness, a generic compliant underactuated hand emulator has been presented [33] that is built upon the open source Klamp't simulator (http://klampt.org) and provides a common tool for simulating a large set of CUHs.

Klamp't allows to simulate interaction with rigid objects taking into account the full dynamics of the hand, object and environment. It integrates models of compliant joints with recent contributions in robust mesh-mesh contact generation methods [21] including contact point clustering and uses adaptive time stepping capabilities to reduce penetration artifacts when simulating contact compliance.

While Gazebo plugins have implemented physics engines with the recent boundary-layer expanded meshes (BLEM) [21] technique for stable contact

generation [34], and there is existing work for simulating the Pisa/IIT SoftHand in Gazebo [35], there are no generic plugins which allow simulating different CUHs flexibly or quickly and seamlessly tuning a hand model.

On the other side, in [5] a dynamic simulation of the Pisa/IIT SoftHand is implemented in the multi-body dynamics simulator MSC Adams [11]. The simulator is used to validate the provided methods to generate pre-grasp palm configurations w.r.t. the object pose. The simulator demonstrated moderate fidelity to an experimental scenario, with some difficulties with hand-object penetrations and estimation of contact normals, as well as a level of performance which is orders of magnitudes slower than the presented method.

Recent developments of the Klamp't simulator provide a uniform interface for specifying emulators for custom sensors and actuators/transmission systems, by using a flexible API and the Python language. These features enable the development of a lean architecture for developing a generic compliant hand emulator and allows simulating a new CUH writing a minimal amount of code, providing fast simulations and allowing to quickly customize the simulation pipeline for grasp planning and learning purposes.

5.1   An Emulator for the Pisa/IIT SoftHand

An underactuated hand is modeled as a set of rigid links articulated by n joints, with n_a degrees of actuation and $n_a < n$. The state of the fingers is denoted as $q \in \mathbb{R}^n$. A control $u \in \mathbb{R}^{n_a}$ gives rise to a net torque on the joints $\tau \equiv \tau(q, u) \in \mathbb{R}^n$ which summarizes the sum of internal torques including gearing, stiffness, damping, joint stops, and friction.

Thus, the dynamics of the robot in contact are given by

$$B(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau(q, u) + \tau_c \tag{1}$$

where $B(q)$ is the robot's mass matrix, $C(q, \dot{q})$ is the Coriolis force matrix, $G(q)$ is the generalized gravity vector, and $\tau_c = J^T f_c$ are the joint torques resulting from external contact forces.

Given an initial state $(q_0, \dot{q}_0)$, a control trajectory $u(t)$, and a final time T a simulator will generate a trajectory of the robot $q(t) : [0, T] \to \mathbb{R}^n$ as well as the motions of other objects $O_1, \ldots, O_m$, taking the dynamics into account as from (1).

A fundamental part of simulating grippers, and in particular CUHs lies in implementing a model of underactuation, to emulate the underactuated transmission system regardless of the particular model of hand and its kinematic and dynamic properties. The proposed emulator has the objective of providing such functionalities, and as described in Sect. 5.2, it is loaded by Klamp't, together with the hand model providing its dynamic and kinematics properties, in order to dynamically simulate a CUH.

Modeling Underactuation and Compliance. Underactuated and compliant hands are linked with transmission mechanisms, e.g., tendons or mechanical

linkages, that distribute actuator effort across multiple joints. They also include spring mechanisms that restore the hand to a consistent rest state once gripping effort is removed, that is, restore deterministic behavior to the hand in spite of underactuation (given a certain value for the actuated variables, an infinite number of possible configurations exist for the underactuated joints). Simulations must allow for actuators to drive multiple links forward, but also to allow for forces on one link to affect the distribution of effort across other links. We model these effects with a formulation based on the adaptive synergies framework [20].

First, we use a general constraint model that relates actuator displacements s to configuration displacements q

$$s = Rq, \tag{2}$$

where the reference configuration is chosen so the zero actuator and joint correspond. The $n_A \times n$ transmission matrix R determines how joint movements pull on each actuator.

We assume the drive mechanism generates torques on each joints in order to maintain these constraints. Denote the tensional force at the tendons generated by the actuators as $f \in \mathbb{R}^n_a$, and the torques generated by the drive mechanism be denoted $\tau_d \in \mathbb{R}^n$. By the principle of virtual work, we have $\tau_d = R^T f$. Let us also define the joint torques produced by spring mechanisms as $\tau_s = -Eq$ where E is a $n \times n$ joint stiffness matrix (which is usually diagonal). Neglecting friction effects, the resultant vector of joint torques is

$$\tilde{\tau} = R^T f - Eq. \tag{3}$$

where $\tilde{\tau} = \tau_c$ at the equilibrium. We then solve for f and q that satisfy the constraints (2) and (3) by solving a system of linear equations, which has a closed form solution in terms of s, R, E, and τ [20] for a constant transmission distribution matrix R and joint stiffness matrix E :

$$f = AJ^T f_c + Bs \tag{4}$$

$$q = CJ^T f_c + Ds \tag{5}$$

with A, B, C, D properly defined matrices, as in [20]. Given the solved f from Eq. (4) and substituting the result in (3), we obtain joint torques to actuate on every joint given a position command s on actuators. Given (1) and imposing quasi-static equilibrium conditions, we then compute the gravity compensation terms and define the actuation torque as

$$\tau = \tilde{\tau} + G(q). \tag{6}$$

This procedure needs to iterate over multiple time steps to achieve equilibrium between mechanism torques and contact forces, which may cause chattering if contact forces are nonsmooth. At rest, the driven hand will converge to the configuration given by Eq. (5), when the underactuated joints are modeled without friction. For hands with configuration dependent transmission distribution

matrix R, we can still use the formula assuming small displacements in q imply little variations in R, and thus setting a number of simulation substeps $\gg 1$. Moreover, while Eqs. (2)–(5) assume proper offsets in the actuator configuration and a zero rest position for the joint elastic elements, it will not be necessarily the case for the hand model, so that when emulating the underactuated transmission, we will resort to commanding a position command $\sigma \in [0,1]$ such that

$$\sigma = \sigma_{\text{scaling}}(s - \sigma_{\text{offset}}) \tag{7}$$

and Eq. (3) becomes

$$\tilde{\tau} = R^T f - E(q - q_{\text{rest}}) \tag{8}$$

Also, not necessarily all joints in the hand will be part of the underactuated transmission, so that all previous formulas do not apply to all joints, but to the subset of underactuated joints n_u. More details on this aspect will be given in the following section.

5.2   Compliant Hand Emulator

Compliant underactuated hands that can be modeled with adaptive synergies are implemented by subclassing the `CompliantHandEmulator` class. Each outer simulation step may be composed of several substeps, during which the simulated world state evolves and sensors and actuators emulators are updated; the controller is updated once per every outer step while forces and dynamics are evolved at a higher rate. To set parameters describing the structure of the hand, the subclass must implement the `loadHandParameters()` function. These parameters define the set of underactuated joints, the synergy joints (i.e. the joints for which a non-trivial transmission exists that relates it to the underactuated joints of the gripper), and the actuated joints with a standard transmission (i.e. joints with `Normal` or `Affine` transmission, which are the default transmission types implemented at the moment in Klamp't) (Fig. 5).

Contact point and force information are obtained thanks to the BLEM [21] algorithm allowing for a stable emulation of the adaptive synergy [34]; thanks to adaptive time-stepping, mesh interpenetration are avoided even in the case of large contact forces where finite contact stiffness can cause the contact to happen below the mesh boundary layer. The relative smoothness of the BLEM contact force predictions are crucial to synergy emulation, since the external forces component f_c in (4) directly influence actuated torques.

The `CompliantHandEmulator` accepts commands both directly from the user (man-in-the-loop) or from a high level controller. It simulates simple joint transmission constraints (present in the SoftHand emulator due to the presence of roll-articular joints [9]) using a simple PID control with a setpoint updated to the reference (eventually underactuated) joint position at every simulation substep, while all the underactuated joints are commanded using Eq. (4).

The simulator obtains a real time factor of ~ 0.5 with simple contact scenarios and ~ 0.24 in a typical grasp simulation with an i7-6500U @ 2.50 GHz processor, 10 substeps (10 ms per time step, 1 ms every physics substep), with the adaptive

timestep scheme impacting performances for more complex contact scenarios or less stiff objects.
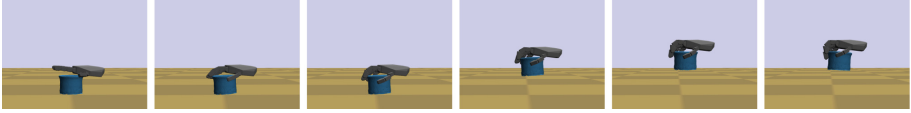

Fig. 5. A sequence showing CUHE being used to simulate the SoftHand grasping and lifting object `play_go_rainbow_stakin_cups_9_red` from the *YCB* dataset by using `simple_controller.py`, provided as example base controller in the simulation framework package for the Robot Grasping and Manipulation Competition

The emulator for the SoftHand is implemented by extending the `CompliantHandEmulator` class, in order to define the underactuated joints, the joint stiffness and the transmission distribution matrix. Pisa/IIT SoftHand was loaded using the *URDF* (Unified Robot Description Format) format from the hand, available in [35]. In particular a `a SoftHandLoader` class has been created which is able to automatically load transmission and stiffness parameters form the URDF. The loader is thus usable as a reference for hands whose parameters are encoded in the model's *URDF*. Virtual joints are issued to implement the synergistic actuators, in particular the joint driving `invisible_wire_link` for the SoftHand *URDF* model. This is a design choice which is enforced in the emulator code and should be kept in new hand models whose transmission is simulated through the emulator. For the five-fingered SoftHand, $n_a = 1$, $n_u = 19$, with one abduction/adduction joint for each finger, plus 3 DoFs except for the thumb (which has 2) which are rolling-contact joints [9]. Hence to the 19 underactuated joints correspond 14 additional joints, since each rolling-contact (Hillberry) joint is modeled using one underactuated joint and one matched joint constrained to follow its parent joint movements [5].

6   Grasp Planning: Minimum Volume Bounding Box Approach

In this section, we present a method to plan grasps for soft hands. Considering that soft hands can easily conform to the shape of an object, with preference to certain types of basic geometries and dimensions, we decompose the object into one type of these geometries, particularly into Minimal Volume Bounding Boxes (MVBBs), which are proven to be efficiently graspable by the hand we use. A set of candidate hand poses are then generated using geometric information extracted from such MVBBs. All the candidates are validated by means of dynamic simulation by using a Soft Hand CUHE to build a database of grasps for each object.

6.1 Bounding Box Decomposition

Algorithm 1 was presented in [17]. The idea is to decompose the object in MVBBs minimizing the volume of the boxes which fit partial point clouds. The algorithm takes a point cloud of an object ($points^{3D}$) and approximates it with MVBBs. This is performed by first projecting a point onto three planes which are the non-opposite faces of the box. Then, using Algorithm 2, the points are split in two sets $p1$ and $p2$. The split is performed for each of the projected points ($faces$) and for each of the two projection axes. After that, the points are approximated with a box and their areas $a1$ and $a2$ are computed. At the end, the algorithm returns the point and the split direction minimizing the mentioned area. The split is then performed for the set of all 3D points and it results in two boxes p and q with a set of 3D points. The reduction rate of the volume of the two new boxes compared with the original is then compared with a user-given parameter t to judge if the split was useful or not. If it is useful, the split is performed and the points in each of the boxes are considered as separate new point clouds to repeat the procedure, otherwise the algorithm is stopped.

Algorithm 1. Approximate the object in MVBB

1: **procedure** BoxApproximate($faces$, $points^{3D}$)
2:      $box \leftarrow FindBoundingBox(points^{3D})$
3:      $faces \leftarrow nonOppositeFaces(box)$
4:      $(p, q) \leftarrow split(FindBestSplit(faces, points^{3D}))$
5:      **if** ($percentualVolume(p+q, box) < t$ **then**   ▷ t is a stop criteria
6:          $BoxApproximate(p)$
7:          $BoxApproximate(q)$
8:      **end if**
9: **end procedure**

Figure 6 shows a comparison of different values of t. Depending on the task that we want to perform, this parameter can assume different values. For example, if we want to grasp objects from handles, like in cups or pots, they can be
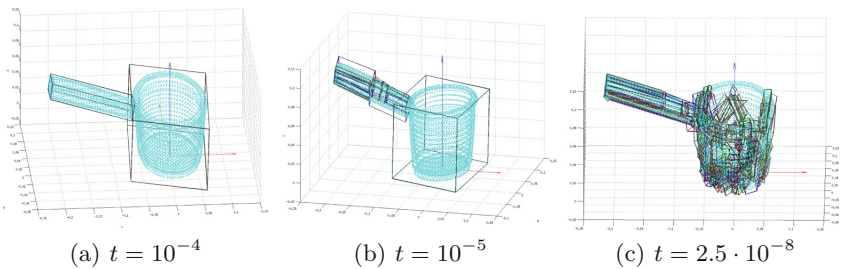

(a) $t = 10^{-4}$ (b) $t = 10^{-5}$ (c) $t = 2.5 \cdot 10^{-8}$

Fig. 6. Comparison of the MVBB generated by the Algorithm 1 using different values of t.

Algorithm 2. Split the point cloud

```
 1: procedure FINDBESTSPLIT(faces, points³ᴰ)
 2:     for i ← 1 to 3 do
 3:         p²ᴰ ← projects(points³ᴰ, faces[i])
 4:         for x ← 1 to width(faces[i]) do
 5:             (p1, p2) ← verticalSplit(p²ᴰ, x)
 6:             a1 ← boundArea²ᴰ(p1)
 7:             a2 ← boundArea²ᴰ(p2)
 8:             if (a1 + a2 < minArea) then
 9:                 minArea ← (a1 + a2)
10:                 bestSplit ← (i, x)
11:             end if
12:         end for
13:         for x ← 1 to height(faces[i]) do
14:             (p1, p2) ← verticalSplit(p²ᴰ, y)
15:             a1 ← boundArea²ᴰ(p1)
16:             a2 ← boundArea²ᴰ(p2)
17:             if (a1 + a2 < minArea) then
18:                 minArea ← (a1 + a2)
19:                 bestSplit ← (i, y)
20:             end if
21:         end for
22:     end for
23: end procedure
```

---

isolated using a large value of t, see Fig. 6(a). Similar values can be used to grasp a cup from above for example. On the other hand, if we want to explore more deeply the geometry of the object (e.g. to grasp edges in pinch grasp configurations), the parameter t must be decreased, see Fig. 6b and (c).

In practice, the selection of the parameter t is related to the translation of high level task specifications to low level grasp actions. Let us consider an example in which a robot has to pick up a pot to pour the content into a glass. In this case, a convenient choice is to grasp the pot from the handle, as shown in Fig. 7(a). Therefore, in this case a very fine object decomposition is not necessary. On the other hand, if we consider the task of passing an object from one hand to another in a bi-manual manipulation setting, the selected box to be grasped could be one in the border of the cooker body, see Fig. 7(b), such that the second arm has more options to decide where to grasp the object without colliding with the first hand. Thus, in this case, a finer object decomposition is beneficial.

6.2   Proposing Grasp Poses

The aim of this section is to explain how we align the hand with respect to the object in order to grasp it. We consider the orientation of each of the MVBBs and the orientation of the object itself. The orientation of the boxes comes from the principal axis determined through Principal Component Analysis (PCA)
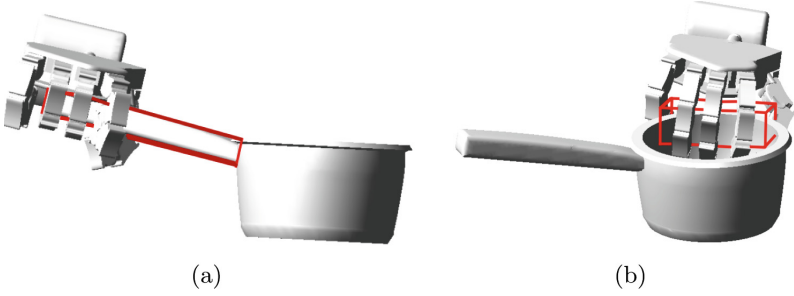
(a)   (b)

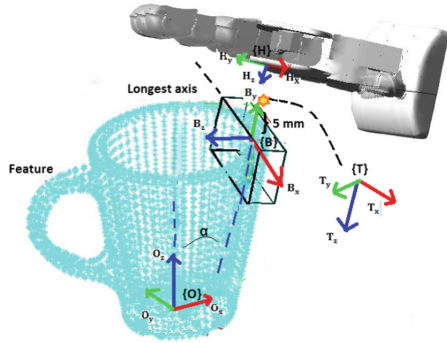**Fig. 7.** The selection of the box to grasp depends on the high level task specification.

Fig. 8. Graphical explanation of the procedure performed to align the hand with each bounding box.

performed in the $FindBoundingBox$ function. The inclusion of the PCA is one of the differences with respect to the original algorithm in [17], and makes the algorithm invariant to the reference frame of the point cloud.

Once the object is decomposed into MVBBs, the next step is to select a box to grasp. There are many criteria to do this, the most promising and useful depending on the task that the robot has to perform once the object is grasped. A possibility is to start generating hand poses from the outermost box. This choice is driven by our first priority of just grasping the object in a robust successful manner — most probably in a power grasp configuration, as the hand is just closed to a certain extend — for, e.g., clearing a table. Once a MVBB is selected, the procedure followed to find the transformation T_O^H describing the pose of the hand with respect to the object is the following:

1. Align the x axis of the hand parallel the longest side of the MVBB.
2. Align the z axis of the hand with the axis of the box which has the smallest angle with respect to the z axis of the hand.
3. Compute the orientation of the y axis to form a right-handed frame.

From this procedure, we can generate the rotation matrix R_O^H defining the orientation of the hand frame H with respect to the object frame O. The frame H is placed 5 mm out of the MVBB, in the negative direction of the z axis defined previously. The procedure is explained graphically in Fig. 8.

From the previous steps we can see that, from steps (1) and (2), the selected axis (longest axis of the box and the one with smallest angle between the axis of the box and the z axis of the hand) can be the same. In this case, we still align the x axis with the longest side of the MVBB, but the z axis is aligned with the axis of the box which has the smallest angle with respect to the vector connecting the centroid of the MVBB with the object centroid. If those axis are parallel so we pick a different axis randomly from the point cloud.

6.3   Pose Variations

The previous procedure generates just a single hand configuration. However, once a MVBB is generated, there is a large number of possibilities to grasp it.

In order to generate more variations for a box, we first set the range of motion in which we can move the hand, translating a distance x_t and rotating by an angle α_t, both along the longest axis of the box, while still not colliding with the object. Figure 9 shows the random variations created for the cup. Variables x_t and α_t generate a 2D space, with high probability of being collision free, from where we pick a random point, with uniform distribution, and then check for collisions with the hand. If this configuration is collision free, then it is a candidate pose to grasp the object. In this work, we generate 40 random configurations for each box and considered the first 5 boxes on the object, thus for each object there are 200 candidate poses.

This procedure constitutes one of the differences with respect to the method proposed in [5], where the authors did not consider collisions in the procedure previous to the simulations. As they explained in the paper, they had a high percentage of failures just because from the very beginning of the simulations, there were many collisions with the object. In this work we performed hand-object collision checking before any simulation and discarded those in collision.

7   The Competition

7.1   Hand-in-Hand Track

The hand-in-hand competition was divided in two sessions: the pick and place and the manipulation sessions. The tasks regarding the pick and place session were the simplest; the attendant has to pick, ten different objects from a basket and place them into four different virtual boxes. In particular the bottle of water, the scissors and the hammer had to be placed in their boxes with a defined orientation. For this kind of tasks the Pisa/IIT SoftHand showed its total simplicity of use. Combined with a custom-made handle it allowed the user to use his own wrist in order to move the hand inside the basket and successfully pick and place all objects (Fig. 10).
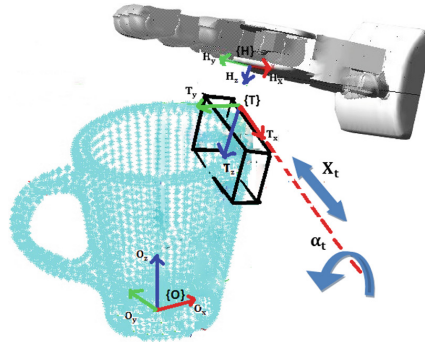
Fig. 9. In order to generate more poses to grasp each box, the hand is rotated and translated along x axis of the box.

Fig. 10. Pisa/IIT SoftHand before and after the pick and place session

The manipulation session was composed of ten different tasks, divided in four levels of difficulty, so during the challenge the user had to accomplish four tasks getting 10, 20, 30 and 40 points depending on the level at which the task belonged. Most tasks inside the first level consisted of grasping simple objects, as spoon or a salt shaker, and then use the object to move a third object from a bowl to another one. In the second level the manipulation was still simple but the objects to be grasped were more difficult because of their geometry (e.g. a plug or a straw) and physical features (e.g. the hammer). In the third level also the manipulation becomes difficult, for example there was a task in which the user had to grasp the plunger of a fully closed syringe and open the syringe without removing it. Finally the last level focused on the repeatability of the manipulation tasks, e.g. there was a task where the user had to grasp a pair of scissors (a difficult object that many gripper were not able to grasp) and then

cut a paper following a line, repeating several cut motions always in the same manner, following the line.

Our strategy for this session was to show videos to the user to let him know that the Pisa/IIT SoftHand can be used in a simple and natural way, just like he uses his own hands (Fig. 11).

Fig. 11. Pisa/IIT SoftHand while cutting a sheet with the scissors

7.2 Simulation Track

A strategy for the second task, **Picking irregular objects from a cluttered shelf** has been devised that makes use of the elements already presented in the architecture explained in Fig. 12. In an offline phase, we proceed to:
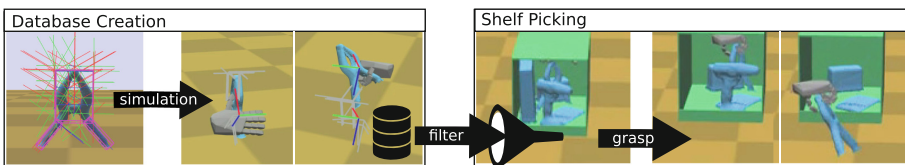

Fig. 12. For each mesh in the set a mvbb decomposition is created, and for each obtained pose a simulation is run. Both successful and unsuccessful simulations are then stored in a database which is filtered once the grasp scenario is known to leave only grasps which do not cause collisions with the environment. In the figure, an example grasping large black spring clamp from the YCB object set

– extract MVBBs from every object of the YCB and Amazon datasets. For objects whose meshes had too many vertices, an automatic simplification using quadric-error edge-collapse simplification [16] has been used. Generate candidates grasp from each MVBB as explained in Sect. 6.2.

- programatically simulate each grasp candidate to flag the successful and unsuccessful grasps and store them in a database. The objects are placed on a plane and subject to gravity, and a scoring procedure is established which consists on closing the hand, lifting for one second by 0.2 m and checking the (world aligned) z-coordinate of the object CoM w.r.t. the CoM at rest against a predefined threshold value in order to detect successful grasps.

During the run (online phase) we then:

- check which of the successful grasps from step (ii) of the offline phase are collision free. Only collisions between hand and environment are considered, but not between hand and objects, nor the collisions during the approach phase.
- rotate the hand to orient it according to the goal grasp pose, align it with the pose, raised slightly, then perform a simple linear approach phase
- lower the hand, grasp, lift, and move the hand on top of the box

The state machine responsible for the second phase would attempt the procedure automatically for all objects on the shelf.

8   Results and Conclusions

For the hand-in-hand track, the ten tasks were completed by the user in 69 s, obtaining the maximum score, showing not only the simplicity of using the handle to control the SoftHand but also the idea behind the design of the hand. In fact, the SoftHand was the only five fingered hand design to participate to the competition, while the other teams used simpler grippers composed of two or three fingers which mechanically have different geometry than a human hand. During the competition we realized how important is the design of the handle to drive the Pisa/IIT SoftHand. The one we used was designed to be simple to use and lightweight, however a deeper analysis on its design is suggested in order to make it more intuitive. Using a glove for the human to teleoperate the robotic hand and the combination with EMG signals is envisioned as future research.

The Pisa/IIT SoftHand was the fastest device to complete both sessions. In the authors opinion it was probably because, for a generic user, it is more intuitive and natural to move, manipulate and grasp objects with a robotic hand which has a similar shape and function of the human hand. In particular we suspect that when users tries to grasp or manipulate an object, they tend to leverage their own knowledge of the successful wrist movements for a given manipulation task.

Also for the manipulation session the SoftHand showed the power of the idea behind its design. In fact, the user was able to use it in a simple and natural way, completing all the tasks in almost 22 min while the second runner spent about 32 min. As for the pick and place session, also in the manipulation, the fact of using a grasping tool similar to their own hands allows the user to leverage his knowledge to place the wrist properly.

In the case of the simulation track, the competition proved a good testing grounds for the grasp algorithms and emulator already presented in the previous sections. Unfortunately, for the first task, the balls resulted to be too big for the hand to be grasped, as the task was more tailored to gripper than to human-like hands. On the second task, though, we have been surprised by the efficiency of the devised algorithms to successfully pick the objects from the shelf. A simple state machine has been developed in order to perform the task, with the addition of a "shaking" procedure at the end of the placing phase, together with the hand opening sequence, in order to make sure that objects would fall in the box without remaining stuck to the hand.

The competition has been an opportunity to tune the hand model in the simulator in order to better match the design of the latest physical models. In particular, by tweaking the transmission matrix R the closing behavior of the thumb has been tuned, thus qualitatively obtaining a similar behavior to the physical hand, both kinematically and with respect to grasp performance. While the *CUHE* allows to tweak a `synergy_scaling` parameter (σ_{scaling} in 7), normalizing R row-wise ensured a consistent value for `synergy_scaling` during tweaking of the hand parameters.

References

1. Ajoudani, A., et al.: A manipulation framework for compliant humanoid COMAN: application to a valve turning task. In: IEEE-RAS International Conference on Humanoid Robots (Humanoids), pp. 664–670. IEEE (2014)
2. Bernstein, N.A.: The Co-ordination and Regulation of Movements, 1st edn. Pergamon Press Ltd., New York (1967)
3. Bicchi, A., Gabiccini, M., Santello, M.: Modelling natural and artificial hands with synergies. Philos. Trans. R. Soc. B: Biol. Sci. **366**(1581), 3153–3161 (2011)
4. Birglen, L., Gosselin, C.M., Laliberté, T.: Underactuated Robotic Hands, vol. 40. Springer, Heidelberg (2008)
5. Bonilla, M., et al.: Grasping with soft hands. In: IEEE-RAS International Conference on Humanoid Robots (Humanoids), Madrid, Spain, 18–20 November 2014
6. Brown, C.Y., Asada, H.H.: Inter-finger coordination and postural synergies in robot hands via mechanical implementation of principal components analysis. In: IEEE International Conference on Intelligent Robots and Systems (IROS), pp. 2877–2882. IEEE (2007)
7. Brygo, A., et al.: Synergy-based interface for bilateral tele-manipulations of a master-slave system with large asymmetries. In: International Conference on Robotics and Automation (2016)
8. Cannon, J.R., Howell, L.L.: A compliant contact-aided revolute joint. Mech. Mach. Theory **40**(11), 1273–1293 (2005)
9. Catalano, M.G., et al.: Adaptive synergies for the design and control of the Pisa/IIT SoftHand. Int. J. Robot. Res. (IJRR) **33**, 768–782 (2014). https://doi.org/10.1177/0278364913518998

10. Ciocarlie, M., Goldfeder, C., Allen, P.: Dexterous grasping via eigengrasps: a low-dimensional approach to a high-complexity problem. In: Robotics: Science and Systems Manipulation Workshop-Sensing and Adapting to the Real World. Citeseer (2007)
11. MSC Software Corp. Adams. http://www.mscsoftware.com/product/adams. Accessed 26 Aug 2015
12. Deimel, R., Brock, O.: A compliant hand based on a novel pneumatic actuator. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 2047–2053 (2013). https://doi.org/10.1109/ICRA.2013.6630851
13. Deimel, R., Brock, O.: A novel type of compliant, underactuated robotic hand for dexterous grasping. In: Robotics: Science and Systems, Berkeley, CA, pp. 1687–1692 (2014)
14. Santina, C.D., et al.: Dexterity augmentation on a synergistic hand: the Pisa/IIT SoftHand+. In: IEEE-RAS International Conference on Humanoid Robots (Humanoids), pp. 497–503. IEEE (2015)
15. Eppner, C., Brock, O.: Planning grasp strategies that exploit environmental constraints. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 4947–4952 (2015). https://doi.org/10.1109/ICRA.2015.7139886
16. Garland, M., Heckbert, P.S.: Surface simplification using quadric error metrics. In: Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1997, pp. 209–216. ACM Press/Addison-Wesley Publishing Co., New York (1997)
17. Geidenstam, S., et al.: Learning of 2D grasping strategies from box-based 3D object approximations. In: Robotics: Science and Systems (RSS), Seattle, USA (2009)
18. Godfrey, S.B., et al.: A synergy-driven approach to a myoelectric hand. In: 2013 IEEE International Conference on Rehabilitation Robotics (ICORR), pp. 1–6. IEEE (2013)
19. Grebenstein, M., et al.: The DLR hand arm system. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 3175–3182. IEEE (2011)
20. Grioli, G., et al.: Adaptive synergies: an approach to the design of under-actuated robotic hands. In: IEEE International Conference on Intelligent Robots and Systems (IROS), pp. 1251–1256. IEEE (2012)
21. Hauser, K.: Robust contact generation for robot simulation with unstructured meshes. In: International Symposium on Robotics Research, Singapore (2013)
22. Hillberry, B.M., Hall Jr., A.S.: Rolling contact joint. US Patent 3,932,045 (1976)
23. Hirose, S.: Connected differential mechanism and its applications. In: Proceedings of 2nd ICAR, pp. 319–326 (1985)
24. Kappler, D., Bohg, J., Schaal, S.: Leveraging big data for grasp planning. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 4304–4311. IEEE (2015)
25. Kim, J., et al.: Physically based grasp quality evaluation under pose uncertainty. IEEE Trans. Robot. **29**(6), 1424–1439 (2013). ISSN: 1552–3098, https://doi.org/10.1109/TRO.2013.2273846
26. Laliberté, T., Birglen, L., Gosselin, C.: Underactuation in robotic grasping hands. Mach. Intell. Robot. Control **4**(3), 1–11 (2002)
27. Latash, M.L.: Fundamentals of Motor Control. Academic Press, New York (2012)
28. Ma, R.R., Odhner, L.U., Dollar, A.M.: A modular, open-source 3D printed under-actuated hand. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 2737–2743 (2013). https://doi.org/10.1109/ICRA.2013.6630954

29. Miller, A.T., Allen, P.K.: Examples of 3D grasp quality computations. In: IEEE International Conference on Robotics and Automation, pp. 1240–1246. IEEE (1999)

30. Odhner, L.U., et al.: A compliant, underactuated hand for robust manipulation. Int. J. Robot. Res. (IJRR) **33**(5), 736–752 (2014). https://doi.org/10.1177/0278364913514466

31. RightHand Robotics. Reflex SF Spec Sheet. http://www.righthandrobotics.com/main:reflex. Accessed 26 Aug 2015

32. Robotiq. 3-finger adaptive robot gripper spec sheet. http://robotiq.com/products/industrial-robot-hand/. Accessed 26 Aug 2015

33. Rocchi, A., Hauser, K.: A generic simulator for underactuated compliant hands. In: 2016 IEEE International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR) (2016)

34. Rocchi, A., et al.: Stable simulation of underactuated compliant hands. In: IEEE International Conference on Robotics and Automation (ICRA) (2016)

35. Rosales, C.J.: Pisa/IIT Soft Hand. https://github.com/CentroEPiaggio/pisa-iit-soft-hand. Accessed 26 Aug 2015

36. Santello, M., Flanders, M., Soechting, J.F.: Postural hand synergies for tool use. J. Neurosci. **18**(23), 10105–10115 (1998)

37. Wimboeck, T., Ott, C., Hirzinger, G.: Passivity-based object-level impedance control for a multifingered hand. In: IEEE International Conference on Intelligent Robots and Systems (IROS), pp. 4621–4627. IEEE (2006)

38. Kai, X., et al.: Design of an underactuated anthropomorphic hand with mechanically implemented postural synergies. Adv. Robot. **28**(21), 1459–1474 (2014)