Merging physical and social interaction for effective human-robot collaboration

Preprint · October 2018

CITATIONS
0

READS
234

6 authors, including:

Phuong D. H. Nguyen
University of Hamburg
13 PUBLICATIONS **62** CITATIONS
SEE PROFILE

Fabrizio Bottarel
Istituto Italiano di Tecnologia
3 PUBLICATIONS **1** CITATION
SEE PROFILE

Matej Hoffmann
Czech Technical University in Prague
59 PUBLICATIONS **608** CITATIONS
SEE PROFILE

Lorenzo Natale
Istituto Italiano di Tecnologia
217 PUBLICATIONS **4,806** CITATIONS
SEE PROFILE

Preprint version; final version available at http://ieeexplore.ieee.org/
2018 IEEE-RAS 18th International Conference on Humanoid Robots, (Humanoids 2018)

Merging physical and social interaction for effective human-robot collaboration

Phuong D.H. Nguyen[1], Fabrizio Bottarel[1], Ugo Pattacini[1], Matej Hoffmann[2], Lorenzo Natale[1], Giorgio Metta[1]

***Abstract*— For robots to share the environment and cooperate with humans without barriers, we need to guarantee safety to the operator and, simultaneously, to maximize the robot's usability. Safety is typically guaranteed by controlling the robot movements while, possibly, taking into account physical contacts with the operator, objects or tools. If possible, also the safety of the robot must be guaranteed. Not less importantly, as the complexity of the robots and their skills increase, usability becomes a concern. Social interaction technologies can save the day by enabling natural human-robot collaboration.**

In this paper we show a possible integration of physical and social Human-Robot Interaction methods (pHRI and sHRI respectively). Our reference task is *object hand-over*. We test both the case of the robot initiating the action and, vice versa, the robot receiving an object from the operator. Finally, we discuss possible extension with higher-level planning systems for added flexibility and reasoning skills.

I. INTRODUCTION

In the near future, robots will certainly leave their safety cages to work alongside human workers. Removing the safety cages requires new capabilities that would eventually enable robots to analyze and assess both the physical (e.g. position and movement of humans, the presence of objects and useful tools) and the social properties of the environment (e.g. emotions and intentions of humans). These requirements translate into the development and, occasionally, further improvement of fundamental "skills", ranging from accurate perception (including human actions) up to the representation of the acquired information in the form of a "shareable knowledge" across different skills. The ultimate goal is to safely and effectively plan and execute generic tasks in the factory floor as well as to assist humans in their daily chores. More pragmatically, given the current level of technological development, we need to resort to a variety of computational techniques such as symbolic and sub-symbolic AI, machine learning, vision, planning and control. The task complexity that we address in this paper is still beyond reach of a single technique "end-to-end", i.e. a comprehensive approach that connects the raw sensory input down to the motor control output. Notable exceptions can be found in the work of Gu *et al.* [1] albeit deployed in simulated environments.

Phuong D.H. Nguyen, Fabrizio Bottarel, Ugo Pattacini, Lorenzo Natale, and Giorgio Metta are with iCub Facility, Istituto Italiano di Tecnologia, Genova, Italy {phuong.nguyen, fabrizio.bottarel, ugo.pattacini, lorenzo.natale,giorgio.metta}@iit.it

[2]Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague, Prague, Czech Republic matej.hoffmann@fel.cvut.cz

In this paper, we take the humbler but extremely practical approach of combining methods from the physical and social Human-Robot Interaction domains (pHRI and sHRI respectively). It must be noted that, indeed, technology is mature enough to deploy markerless perception of the human body in 3D, to recognize and model generic objects for grasping, seamlessly integrating visual perception with whole-body force/torque control, tactile sensing, and speech-based communication to name a few. In addition, machine learning provides the ability to teach the robot about new objects and tools, whose descriptions can be stored and organized – at least for the scope of these experiments – into a standard database. To set the stage for our work, we start by briefly reviewing some recent pHRI and sHRI architectures, their strengths and limitations.

Most pHRI frameworks focus on the "low-level" interaction (e.g., contact detection, avoidance and control) and consider humans merely as other objects in the workspace the robot needs to deal with. In their architecture, De Luca *et al.* [2] integrate residual-based collision detection and reaction (gathered from the proprioceptive layer) with collision avoidance (leveraged on depth information provided by a Kinect sensor). Since this work models the environment by considering only obstacle-to-robot distances, we argue that this approach is not flexible enough to scale up to more complex collaborative tasks. Haddadin *et al.* [3] outline a different solution where the robot simply switches between different functional modes (e.g. autonomous task execution with/without human, cooperation with human) based on the state of the human in the robot's workspace as detected through proximity sensors surrounding the robots. Although this approach combines different control techniques (e.g. impedance control, residual-based collision detection, and task relaxation reaction) as well as several sensory modalities (e.g. laser-based imaging, visual-based edge filtering) to offer safe HRI, it lacks a knowledge-based communication channel and thus it is not seamlessly extensible to other applications. It does a very good job though in implementing flexibility in the physical interaction layer.

On the other end of the spectrum, sHRI frameworks often overlook the physical aspects of the interaction (e.g safety and physical contacts) or simply resort to path planning methods to deal with static or slowly changing environments. For example, Lemaignan *et al.* [4] in their recent work present a cognitive architecture for service robots that supports human actions and decisions. However, it only utilizes path planning based methods [5], [6] to guarantee that the path of the robotic manipulator is collision free, which is

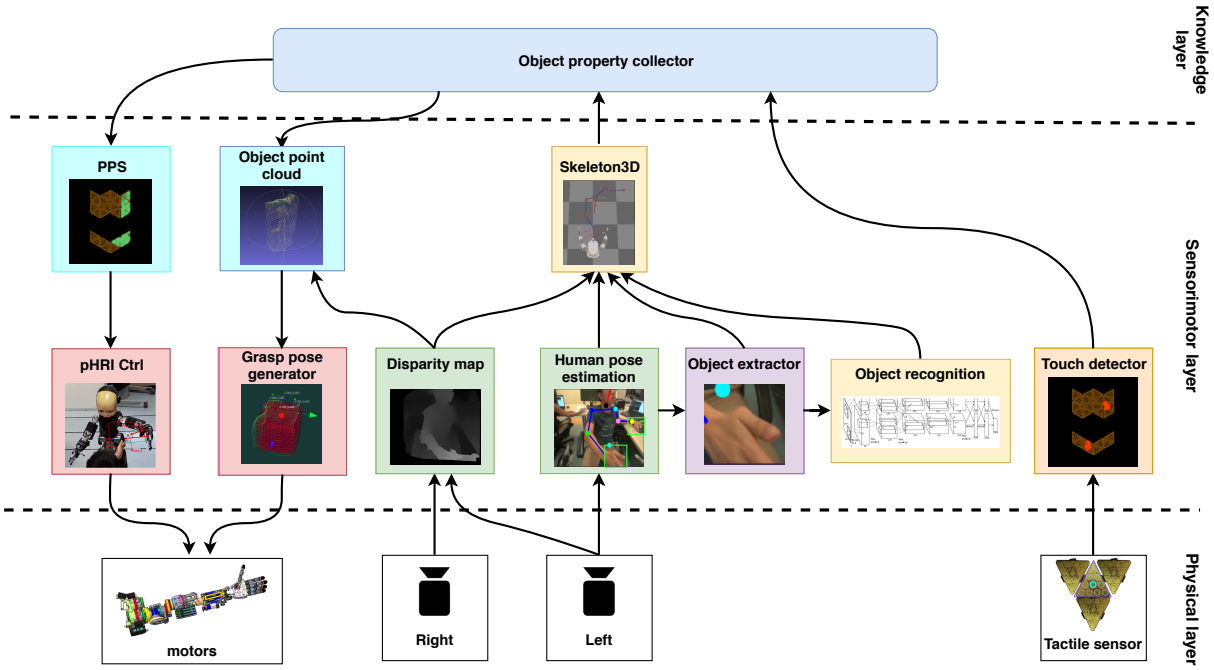


Fig. 1. Overview of the overall system comprising perception (right side) and action (left side) pathways. At the physical level, perception includes vision and touch. The robot's visual system allows for stereoscopic vision. Low-level motor control allows specifying the position trajectories of the joints exploiting as feedback also a combination of pressure (from the tactile sensors) and force information (from a number of 6-axis force-torque sensors located on the robot's structure). The sensorimotor layer transforms raw sensory data into symbolic tokens (e.g. object identities, posture, 3D shape, human body posture, etc.) that can be easily stored into the "object property collector" database. This symbolic knowledge is used to control action, as for example to avoid contacts rather than to grasp objects, through reasoning modules (i.e. *PPS, Object point cloud, pHRI Controller, and Grasp pose generator*).

difficult to satisfy in highly dynamic environments, and in particular when interacting with a human partner.

Moulin-Frier *et al.* [7] and Fischer *et al.* [8] developed the so called DAC-h3 cognitive architecture. One of DAC-h3's main strengths is its implementation, which is an ensemble of functional modules. Functional modules can be mapped one-to-one to software modules of typical middleware systems. The authors validate DAC-h3 by experimenting with human-robot and robot-object interaction to acquire and express procedural knowledge. Although the robot can execute a wide repertoire of actions such as waving, pointing, pulling, pushing objects in a table-top setting, DAC-h3 employs exclusively predefined motor primitives and does not address the problem of safety (e.g. avoiding human and moving objects). It is a very good reference implementation for sHRI. Along the same line, in Moulin-Frier *et al.* [9], the authors integrate diverse AI techniques into a single cognitive architecture that combines symbolic reasoning with embodied behaviors, yet they do not consider the "low-level" details of physical interaction. Notably, all the aforementioned systems employ fiducial markers and/or exteroceptive sensors to enhance the robot perception. Nonetheless, certain elements of DAC-h3 are readily combined into our system. In Section IV, we speculate about a possible integration with DAC-h3.

To summarize, the main contribution of this paper is the design of a complete control system that merges elements of pHRI and sHRI, namely:

- A compact human-centered visual perception system for humanoid robots;
- A visuo-tactile reactive controller that allows the robot to safely react in both *pre-* and *post*-collision phases;
- A simple symbolic "storage" of information about humans, objects, and tools supporting social interaction.

Some parts of the system presented in this paper were developed and analyzed in our recent publication [10], namely the human keypoints estimation, the reactive controller and the Peripersonal Space representation. They are by and large reused "verbatim" in this work. Here we demonstrate that our approach can handle effectively different types of interaction. We develop two main experiments, where the robot is given an object by the human partner and it is subsequently asked to grasp another object from a table top to perform a handover task.

The remainder of the paper is organized as follows: we present the method in details in Section II, and analyze performance showing experiments and results in Section III. Finally, in Section IV, we analyze the possibility of integrating this work into existing cognitive architectures.

II. METHODOLOGY

A. General architecture

The underlying architecture of our framework is shown in Fig. 1, where *functional modules* are classified into three different layers described in the following:

- The **physical layer** consists of the low-level systems of the iCub humanoid [11]: the *stereo-vision*, the *artificial skin* covering the robot body, and the *joint actuators*. This layer allows the robot to perceive the surroundings as well as act on the environment.

- The **sensorimotor layer** encompasses those modules responsible for processing the raw signals produced by the physical layer in order to yield meaningful internal representations: the *touch detector*, the *disparity map*, the *human pose estimation*, the *object extractor*, the *object recognition* and the *skeleton3D* for visual input. Components responsible for the computation of control signals are also listed here, such as the *Peripersonal Space*, the *pHRI controller*, the *Object point cloud* and the *Grasp pose generator*.
- The **knowledge layer** contains the *Object properties collector* module (discussed in Section II-C), whose task is to store and manage the properties of the entities perceived from the environment.

B. Environment acquisition and perception:

1) Human detection and tracking: In [10], we proposed a real-time framework to estimate the 3D pose of humans from the 6 DoF stereo vision system mounted in the iCub head. The framework is composed of 2 steps: (1) a 2D human pose detection given as a set of keypoint pixels $[u_i, v_i]$ extracted from the raw images using the *DeeperCut* model [12]; (2) a 3D human pose reconstruction from 2D information and depth map. The latter is performed by averaging the spatial projection of each 2D keypoint along with its neighbors through the depth map. The output set of 3D coordinates $[x_i, y_i, z_i]$ is then refined by applying median filtering.

2) Context-aware object detection and tracking: To provide the robot with a context-aware ability during the collaboration with the human partner, we extend the above human tracking framework to incorporate object recognition. To this end, we adopt the method developed by Pasquale *et al.* [13], which turns to be simple yet efficient in our setting, where the system must detect and recognize objects held by the human. The proposed image recognition system utilizes a *CaffeNet* [14] deep neural network (DNN) pre-trained on the ImageNet dataset to extract features from the input images, and a *Regularized Least Squares* method for the classification stage. The algorithm also allows partners to train the robot with novel object via verbal annotation.

Unlike [13] employing a heuristic motion detector to acquire cropped images for the DNN, we propose an accurate and flexible solution specifically designed for the HRI context. By resorting to the keypoint pixels obtained from the 2D human pose computation, we precisely estimate in real-time the bounding boxes containing the human hands (with or without objects) that in turn are passed on to the image recognition system for labeling purposes. The size of the bounding boxes are constantly adapted based on the distance of the human hands as retrieved from the depth map.

3) Physical collision detection through artificial skin: The body of iCub is covered by a layer of artificial skin composed of capacitive tactile sensors [15], termed *skin taxels*. The poses of the skin taxels are calibrated with respect to the kinematic model of the robot , and are kept updated during robot movements. Thus, physical contacts with the iCub skin can be sensed and localized accurately. Notably, this approach differs from other recent methods (e.g. [16]) that typically rely on proprioceptive inputs instead. To reduce spiking effects, we aggregate multiple adjacent tactile contacts firing concurrently over a preset threshold into one representative *super contact* whose activation a^t_{PPS} corresponds to the highest pressure value measured at the relative taxels. The *super contact* is also parameterized in terms of its location $\mathbf{P}^t_C$ and normal vector $\mathbf{n}^t_C$, which also encodes, to a first approximation, the collision direction (Fig. 2).

C. Centralized knowledge representation through Object Properties Collector (OPC):

In order to effectively cooperate with humans, robots do not only need to perceive the surroundings through their sensors, but they are asked to convert these representations into a "common knowledge" that can be shared with the partners to support reasoning and task planning. For this purpose, we adopt an ontology based framework [17] for knowledge representation. We consider these representations as the centralized working memory of the robot during the interaction with the environment. Thereby, we partially solve the grounding problem of pure symbolic cognitive systems, where environment stimuli are firstly transformed into lower dimensional representations with machine learning methods, and then are mapped into symbols (given *a priori*) in a database.

In this regard, the working memory makes use of the *Entity* and the *Relation* to identify basic concepts and the connections among different entities, respectively. Thus, the perceived objects are denoted as *Objects*, an abstract type of *Entity*, composed of some physical properties such as position, dimension, valence (further properties like objects affordances can be added easily). On the contrary, objects that have self-motion abilities are deemed as *Agents*: humans and robots fall under this class. A structured hierarchy of classes can be constructed along the same line, comprising e.g. *Bodies*, *Emotions*, *Beliefs* etc., as described in [17].

Leveraging on this knowledge management, the human partner can be represented within the OPC memory as an *Agent* whose *Body* parts are localized in 3D through vision (see Section II-B.1). A similar process applies to the visually recognized objects along with their properties (e.g. location, color, valence) that are relevant to the task at hand.

D. Learned Peripersonal Space (PPS) as an adaptive embodied perception layer

In Roncone *et al.* [18], the authors propose a distributed multisensory representation model, named *Peripersonal Space* (PPS), to associate the visual and tactile inputs prior to and at collision time. This model is then trained by allowing approaching objects to contact physically with the robot skin taxels. In deployment, this representation serves as a mapping of visual stimuli to body parts of the robot, parametrized in terms of location $\mathbf{P}^v_C$ and normal vector $\mathbf{n}^v_C$ of magnitude a^v_{PPS}.

An interesting application of this concept can be found in [10], where Nguyen *et al.* consider the PPS as a distributed safety zone around the robot body. They build on the model in order to modulate (expand or shrink) the spatial extension of such a zone depending on the involved part, resembling what is observed in humans (e.g. smaller zone for the hands, bigger for the head). Formally, the modulated PPS signal $a_{m,i}(t)$ occurring at the *i*-th taxel w.r.t. the valence–threatening value $\theta_k(t)$ of the *k*-th object at time instant *t* is given by:

$$a_{m,i}(t) = a_i(t)[1 + \theta_k(t)], \tag{1}$$

where $a_i(t)$ represents the original PPS activation.

In this work, we further exploit this adaptable PPS representation for different collaboration contexts. Human body parts that are meant to be contactable during the cooperation (right hand holding an object) would entail lower activations than other parts (left hand). This allows the robot to approach the right hand for a close interaction while handing over the object and to avoid collisions with other parts (i.e. left hand, head). This contextual modulating mechanism can be synthesized as follows:

$$a_{m,i}(t) = \begin{cases} \min\Big(a_i(t),\, a_i(t)[1 + \theta_k(t)]\Big) & k \text{ contactable} \\ \max\Big(a_i(t),\, a_i(t)[1 + \theta_k(t)]\Big) & \text{otherwise} \end{cases} \tag{2}$$

E. Controllers:

1) Bio-inspired reactive controller for safe physical interaction: Most robot movements in the interaction scenarios can be formalized as reaching with obstacle avoidance. To this aim, we proposed earlier in [10] a reactive controller tasked with solving the following nonlinear constrained optimization problem:

$$\dot{\mathbf{q}}^* = \arg\min_{\dot{\mathbf{q}} \in \mathbb{R}^n} \left\| \bar{\mathbf{x}}_{EEd} - \left(\bar{\mathbf{x}}_{EE} + T_S \cdot \mathbf{J}(\bar{\mathbf{q}})\dot{\mathbf{q}}\right)\right\|^2. \tag{3}$$

The controller aims to find the optimal joint velocities $\dot{\mathbf{q}}^*$ at each time instant by minimizing the distance between the desired end-effector pose $\bar{\mathbf{x}}_{EEd}$ and the one-step prediction of the robot current pose $\bar{\mathbf{x}}_{EE}$, complying with the constraints of the feasible joint range $[\mathbf{q}_L, \mathbf{q}_U]$ and the joint velocity limits $[\dot{\mathbf{q}}_L, \dot{\mathbf{q}}_U]$. $\mathbf{J}(\bar{\mathbf{q}})$ and T_S are the Jacobian of the instantaneous configuration $\bar{\mathbf{q}}$ and the sampling period, respectively.

In this setting, visually perceived objects elicit PPS activations, thus reshaping the movements of the robot's parts through the PPS representation (see Section II-D) by adapting in real-time the joint velocity limits (refer to [10] for more details). Remarkably, the controller can respond in a similar manner to tactile stimuli, hence dealing with *post*-collision scenarios. Fig. 2 depicts the occurrence of a physical contact, eliciting the activation of a skin taxel on a robot body part. As a result, the bounding values of the velocities of the corresponding joints (e.g. mainly the elbow joint as visible in Fig. 2) are adapted accordingly. In formal terms, the joint velocity constraints of Eq. 3 relative to both visual and tactile inputs are expressed by:

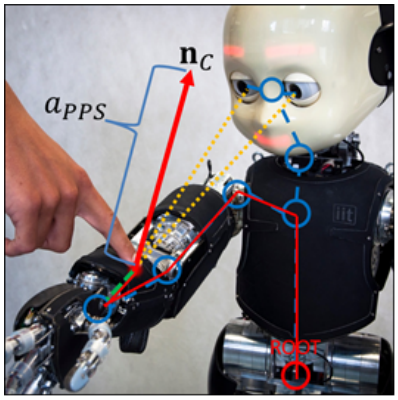


Fig. 2. React-control dealing with a tactile stimulus. The diagram shows the quantities involved when a control point is elicited upon the detection of a real contact (physical contact triggered by tactile sensors).

$$\begin{aligned} \mathbf{s} &= -\mathbf{J}_C^T \cdot \mathbf{n}_C \cdot V_C \cdot K \cdot a_{PPS} \\ \dot{\mathbf{q}}_{L,j} &= \max\left\{V_{L,j},\, s_j\right\}, \quad s_j \geq 0 \\ \dot{\mathbf{q}}_{U,j} &= \min\left\{V_{U,j},\, s_j\right\}, \quad s_j < 0 \end{aligned} \quad , \tag{4}$$

where C is a control point attached to a generic robot link, represented by either a *mapped PPS locus* or a *super contact*, depending on the nature of the input signal (i.e. visual or tactile, respectively); $\mathbf{J}_C$ is the Jacobian associated to C; V_C is a gain factor used for avoidance; V_L, V_U are a predefined set of bounding values of joint velocities; a_{PPS} is either the visual (a^v_{PPS}) or the tactile (a^t_{PPS}) activation; K is a tunable gain. In particular, the gain K is set to be higher for tactile events than visual events (i.e. three times in this implementation), reflecting the notion that a physical collision detected by the skin system is more critical than a collision predicted from the visual input.

2) Superquadric-based object grasping controller: Given the desired object, the robot must be able to compute an optimal grasping pose in terms of position and orientation. The *one-shot* approaches adopted in [19] [20] suggest that modeling objects with analytical shapes (i.e. superquadrics) is a good assumption for autonomous grasping of partially occluded objects. Drawing inspiration from such work we propose our own grasp planner, relying on superquadric shapes to model objects and employing geometric, analytical and kinematic considerations for pose generation and evaluation. The pose synthesis pipeline is integrated with the HRI framework as outlined in Fig. 1 and consists of the 5 steps described hereinafter.

a) Point cloud acquisition: The algorithm prompts the OPC module to acquire the location of the object to grasp in the field of view. Once found and segmented from the background, the location of single object pixels in the 3D space are retrieved from the *disparity map* and the information is stored as a point cloud. Possible outliers are removed by applying density-based spatial clustering [21].

b) Superquadric modeling: At this stage, we retrieve the superquadric that best fits the point set $P \in \mathbb{R}^3$. The mathematical representation of the superquadric has the following implicit form:

$$\left(\left|\frac{x-x_c}{s_x}\right|^{\frac{2}{\epsilon_2}} + \left|\frac{y-y_c}{s_x}\right|^{\frac{2}{\epsilon_2}}\right)^{\frac{\epsilon_2}{\epsilon_1}} + \left|\frac{z-c_z}{s_z}\right|^{\frac{2}{\epsilon_1}} = 1, \quad (5)$$

where $p_i = \{x_i, y_i, z_i\} \in P$. The vertical axis of the superquadric is constrained to be orthogonal to the table surface, therefore the superquadric rotation is defined by the angle ϕ around the z axis. This is a reasonable assumption since the objects are modeled with a single superquadric, and constitutes the first branching point of our approach from [19]. To find $\lambda = \{x_c, y_c, z_c, s_x, s_y, s_z, \epsilon_1, \epsilon_2, \phi\} \in \mathbb{R}^9$ we minimize the distance between each $p \in P$ and the superquadric surface. The problem can be cast as the least squares minimization:

$$\lambda = \arg\min \sum_{i=1}^{|P|} \left(\sqrt{s_x s_y s_z}(F(p_i, \lambda) - 1)\right)^2, \quad (6)$$

where $F(p_i, \lambda)$ is the left-hand side of (5). We constrain some of the parameters in order to obtain a convex shape and not to extend under the table surface; therefore, such optimization problem is constrained and nonlinear. Another difference between this approach and the one described in [19] is that we use the analytical gradient of (6) during optimization, instead of finite differences.

c) Pose generation: Feasible hand poses must be generated in order to perform *top* and *side power grasps*. The robot can use any hand it is commanded to. With respect to [19], our approach seeks for solutions in a narrowed search space for position and orientation:

- position is constrained to the cardinal points (intersections between axes and surface) of the superquadric, so that the palm touches the surface;
- hand orientation (shown in Fig. 3(a)) is constrained so that each pose axis is parallel to one superquadric axis;
- side grasps are constrained to having the thumb always point upwards.

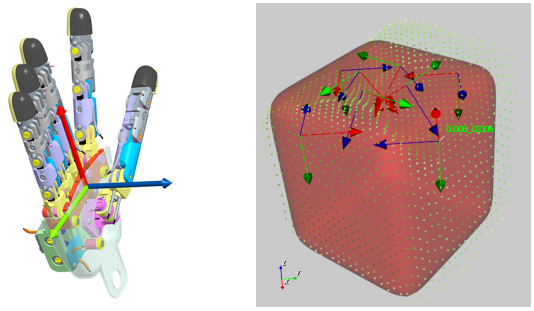
Grasp poses are generated as detailed in Algorithm 1, and are represented as homogeneous transformations $g_i = \begin{pmatrix} R_i & T_i \\ 0 & 1 \end{pmatrix}$ linking the robot palm to the root frame. In Operation 14, the object is graspable if its cross section fits in the hand of the robot. In Operation 11, the pose is rotated around the wrist pitch to avoid collision between the thumb and the object during approach.

d) Pose ranking: The candidate poses are then ranked according to the square norm of the position error, calculated with the inverse kinematics solver for iCub [22]. The poses that reach at least a degree of positioning accuracy (e.g. error < 1 cm) are further ranked according to the cost function J_i in (7). The parameter w weighs the two components of J_i, where $J_{i,1}$ accounts for the orientation accuracy and $J_{i,2}$ favors grasps around the smallest side of the superquadric:

Algorithm 1 Grasp generation

Input:
Center s_c, axes unit vectors $\{\vec{a_x}, \vec{a_y}, \vec{a_z}\}$, size $\{s_x, s_y, s_z\}$ of the superquadric
Output:
Grasp candidate set S_g
Grasp pose $g_i = \{R_i, T_i\} \in S_g$

1: **procedure** GENERATEGRASPS(λ)
2: $S_g = \emptyset$
3: $S_{g_x} \leftarrow \{\vec{a_x}, \vec{a_y}, -\vec{a_x}, -\vec{a_y}\}$ ▷ g_x, g_y search spaces
4: $S_{g_y} \leftarrow \{\vec{a_x}, \vec{a_y}, \vec{a_z}, -\vec{a_x}, -\vec{a_y}, -\vec{a_z}\}$
5: **for** $\vec{g_{i,x}} \in S_{g_x}$ **do**
6: **for** $\vec{g_{i,y}} \in S_{g_y}$ **do**
7: $\vec{g_{i,z}} \leftarrow \vec{g_{i,x}} \times \vec{g_{i,y}}$
8: $T_i \leftarrow s_c - s_z \vec{g_{i,z}}$
9: $R_i \leftarrow [g_{i,x}\ g_{i,y}\ g_{i,z}]$
10: **if** ISGRASPFEASIBLE(R_i) **then**
11: $S_g \leftarrow$OFFSET(R_i, T_i)
12: **return** S_g
13: **procedure** ISGRASPFEASIBLE(R)
14: **if** OBJECTISGRASPABLE(s_x, s_y, s_z) **then**
15: **if** ORIENTATIONACCEPTABLE(R) **then**
16: **return** true
17: **return** false



(a) Hand reference frame (b) Computed grasp poses

Fig. 3. Grasp poses computation. In (a) the g_x axis is represented in red, g_y in green and g_z in blue. In (b), the acquired point cloud is displayed together with the modeled superquadric and computed grasp poses expressed in the root reference frame. The poses are represented according to the hand axes colors in (a) and the green captioned pose is the optimal given the superquadric.

$$\begin{cases} J_{i,1} = ||\tilde{o_i} \sin\tilde{\theta}|| \\ J_{i,2} = 1 - \frac{s_{finger}}{s_{max}} \\ J_i = wJ_{i,1} + (1-w)J_{i,2}, w \in [0,1] \end{cases}. \quad (7)$$

In (7), $\{\tilde{o_i}, \tilde{\theta}\}$ is the axis angle representation of $R_i \hat{R}_i^T$, and $\hat{R}$ is the hand orientation that the robot can actually reach (according to the inverse kinematics solver) with respect to the root reference frame. s_{max} is the size of the longest axis of the object superquadric, while s_{finger} is the size of the superquadric axis that lies in the direction of the fingers (hand x axis).

e) Reaching and grasping: The best pose according to (7) is selected, and the robot reaches for it with one hand

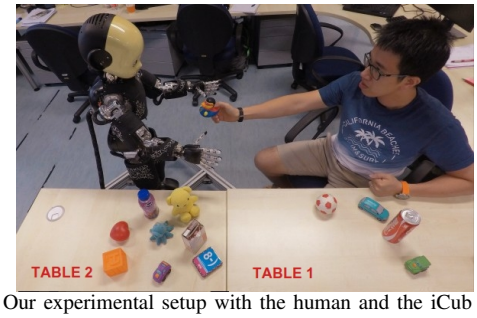


Fig. 4. Our experimental setup with the human and the iCub sharing the workspace. The human is sitting next to *Table 1* while the iCub is located near *Table 2*.

to perform a five fingers power grasp. The grasp is executed by actuating the proximal and distal finger joints until the load torque exceeds a contact force threshold. The grasping pipeline is freely available online[1].

III. EXPERIMENTS & RESULTS

In this section, we evaluate our method in two different hand-over experiments.

- *Safe human-robot hand-over task*, where the robot receives the object from the human.
- *Safe robot-human hand-over task*, where the robot has to pick up the object from a table to then perform hand-over.

Our experimental setup is presented in the Fig. 4, where there are two tables, denoted as *Table 1* and *Table 2*. The human partner is next to *Table 1* and cannot reach objects located on *Table 2*, whilst the robot can reach and grasp objects lying on *Table 2*. This setup is designed to simulate the situation where the robot and the human need to cooperate to complete a shared task, such as moving an object from *Table 1* to *Table 2* or *vice versa*. Intentionally, we set the hand-over phase of robot's action long enough (at least 15 s) to enable any possible physical interaction with the user.

A. Safe human-robot hand-over task

In this experiment, when engaged by the human partner, the robot has to look for the requested object the human holds in his hand, to then take it over and place it on the table. During this interaction, the robot movements can be interfered by the human so that the robot needs to anticipate possible collisions in order to guarantee a safe cooperation. The dialogue between the human and the robot can be scripted as below:

PARTNER : Hi iCub, help me put the DUCK in the basket!
ICUB : I don't have the DUCK. You have the DUCK. Please give it to me!
(PARTNER *shows the* DUCK *to* ICUB, *and* ICUB *moves the hand to receive the* DUCK *from* PARTNER)

In details, iCub locates the human hand and the object with its visual system, then moves its hand to approach the object. As it is evident in the Fig. 5 and Fig. 6, the valances of the human right hand holding the object as well

[1] https://github.com/robotology/grasp-pose-gen

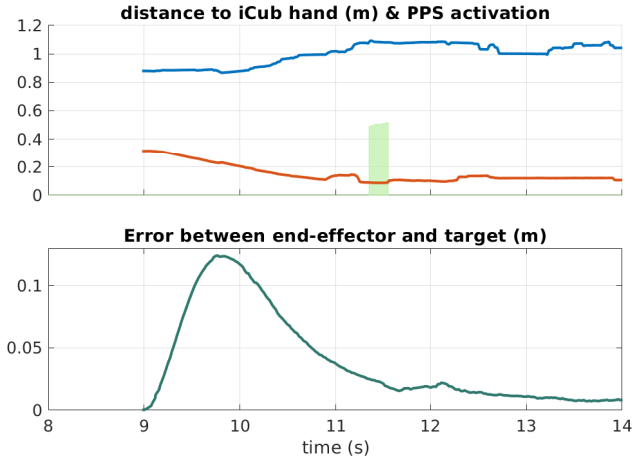


Fig. 5. Experimental results when iCub is approaching the human right hand holding the object. Top: the distance from the robot right hand to the human hands are shown (red for right, blue for left; aggregated PPS visual activation on the robot right hand are shown by the green shaded areas. Bottom: distance between the end-effector and the target object.

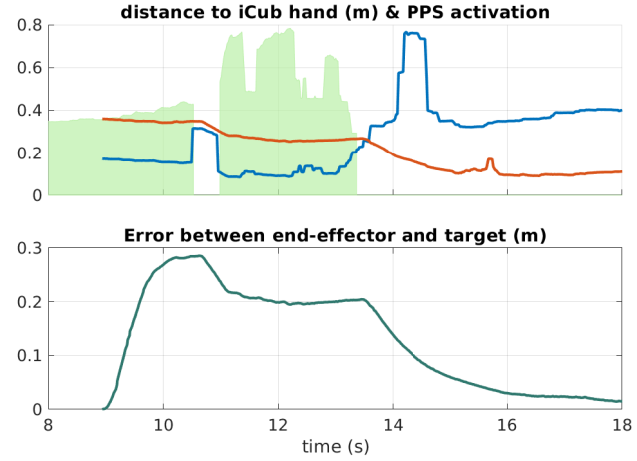


Fig. 6. Experimental results when iCub is approaching the human's right hand holding the object and the human interferes with the robot movements using his left hand. Relevant quantities are explained in the Caption of Fig. 5.

as the object itself are reduced (at $t \approx 9$ s) from normal to approachable values, while the value of the human left hand remains unchanged. As a result, the visual PPS activation on the robot right hand is almost null (barring the short time range $t \in [11.3, 11.5]$ s), even though the human right hand is very close (see Fig. 5-Top). Therefore, the robot can move directly to reach for the target object as long as it does not detect any approaching obstacle, as illustrated by a quickly decreasing distance between the robot end-effector and the object in Fig. 5 ($t \in [10, 12]$ s). This is not the case presented in Fig. 6, where the human moves his left hand to interfere with the robot movements; in fact, the relative distance (blue profile) becomes very close to 0 at $t \approx 11$ s, causing very high PPS activation. The iCub correctly reacts by anticipating its planned movement for safety reasons until the instant the human moves his left hand away ($t \approx 13$ s). Afterwards, iCub continues to move its hand to approach the human right hand to receive the object safely. A detailed analysis of the joint velocities commanded at the robot arm during the interaction can be found in [10].

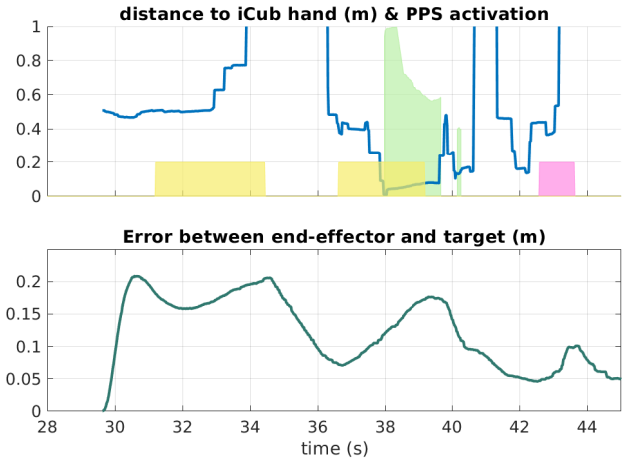


Fig. 7. Experimental results when iCub is approaching the human right hand to hand-over the object. Top: we show only the distance from human left hand (blue) for sake of simplicity; tactile contacts with the robot right forearm (yellow) and right hand (pink) are also depicted.

B. Safe robot-human hand-over task

In this experiment, the robot has to find the object lying on *Table 2*, grasp it properly (Section II-E.2) to finally perform hand-over toward the human partner. The cooperation task is initialized with the following dialogue between the two agents:

PARTNER : Hi iCub, can you give me the OCTOPUS?
ICUB : I have the OCTOPUS on my table. I will give it to you.
(ICUB *then grasps the* OCTOPUS *from* TABLE 2 *and shows the* OCTOPUS *to* PARTNER)

More specifically, iCub looks for the requested object lying on *Table 2* and calculates the best grasping pose using the procedure described in Section II-E.2. If a suitable grasp is found, iCub reaches for the object to perform a power grasp. With the object in hand, the robot brings the object to the hand-over location that best suits the estimated human pose. As displayed in Fig. 7, the iCub movements toward the desired pose are continuously adapted to accommodate for any incoming visual obstacle, represented in this case by the human left hand ($t \in [38, 40]$ s) as well as any unexpected physical contacts ($t \in [31.2, 34.2]$ s and $t \in [36.5, 39]$ s). Both visual and tactile events do constrain the admissible range of joint velocities, generating the avoiding behavior of the robot, as it is visible in Fig 7-Bottom in terms of the distance between the end-effector and the hand-over position (blue profile). This behavior guarantees a safe physical interaction between the human and the iCub in *pre-* and *post*-collision phases.

C. Quantitative assessment of the interactions

In this section we report on the success rates of the two hand-over experiments carried out with a set of different objects. We ground our quantitative analysis to the details level of 4 sub-tasks per interaction that need to be completed in sequence. In particular, we identify the sub-tasks $\{$*Recognize, Localize, Receive, Drop*$\}$ and the sub-tasks $\{$*Detect, Plan, Grasp, Give*$\}$ for the *human-robot* and the *robot-human* hand-over sequences, respectively. For each object, we repeat the whole experiment 10 times. Note that the sub-task corresponding to the control of the robot movements is omitted as the reactive controller guarantees the safety of the action at all times.

TABLE I
SUCCESS RATES OF HUMAN-ROBOT HAND-OVER TASK

Object	Sub-tasks			
	Recognize	Localize	Receive	Drop
Octopus	100%	90%	100%	100%
Duck	100%	100%	100%	90%
Bottle	100%	90%	100%	100%

TABLE II
SUCCESS RATE OF ROBOT-HUMAN HAND-OVER TASK

Object	Sub-tasks			
	Detect	Plan	Grasp	Give
Ladybug	100%	80%	100%	100%
Box	100%	90%	90%	90%
Bottle	100%	90%	80%	100%

The high success rates recorded in the two hand-over experiments and reported in Table I and Table II demonstrate the effectiveness of our solution in scenarios where both the physical and social properties of the interaction are relevant during the human-robot collaboration.

IV. CONCLUSIONS & DISCUSSIONS

We introduced a compact, fully integrated and scalable architecture that fills in the gap between physical and social HRI with the following key features: (i) a markerless 3D context-aware visual perception system, (ii) a multi-modal visuo-tactile reactive controller along with a fast and efficient grasp planner to enable safe interaction, and (iii) a simple database for storing symbolic knowledge. We showed the complete system working in real-time controlling a robot in the human-robot and robot-human object hand-over tasks while guaranteeing safety for the human experimenter. Moreover, we believe that it is feasible to adapt our architecture to different robots equipped with a similar set of sensors (*i.e.* stereo-vision, tactile and/or force/torque sensing).

Future work will include integration with a state-of-the-art cognitive architecture. In particular, safe behaviors generated through our visuo-tactile component recall and advance the mechanisms of the *Somatic* and *Reactive* layers of the DAC-h3 architecure as described in [7]–[9]. Likewise, our visual pipeline does tightly connect to the *World* and *Action* layers available in DAC-h3 at least within the limits of our simplified task planning. Thereby, in this context, we can incorporate almost seamlessly further DAC-h3 functional modules such as the *Synthetic Sensory Memory*, the *Perspective Taking* and the *Autobiographical Memory* in order to enrich the current repertoire of capabilities as for example adding action recognition skills. If we compare with the architecture of Lemaignan *et al.* [4], we do not share functional modules as such, however, our overall structure is similar to [4] at the symbolic layer. This similarity may

pave the way to a future integration of functionalities that are missing in our design but readily accessible in [4], such as the *human-aware task planning*.

In conclusion, we aim to further develop the present system with the goal of implementing a general and principled cognitive architecture, by taking advantages of the integration with other existing approaches. Paramount for effective HRI is to improve action planners to tackle fast dynamic environments [23], while taking into account ergonomics, as discussed for example in [24].

ACKNOWLEDGMENT

Phuong D.H. Nguyen was supported by a Marie Curie Early Stage Researcher Fellowship (H2020-MSCA-ITA, SECURE 642667). M. H. was supported by the Czech Science Foundation under Project GA17-15697Y. The authors would like to thank Vadim Tikhanoff and Giulia Pasquale for their valuable assistance with the integration of the object recognition system.

REFERENCES

[1] S. Gu *et al.*, "Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates," in *IEEE Int. Conf. on Robotics and Automation*, May 2017, pp. 3389–3396.

[2] A. De Luca *et al.*, "Integrated control for pHRI: Collision avoidance, detection, reaction and collaboration," in *Biomedical Robotics and Biomechatronics (BioRob), IEEE Int. Conf. on*, 2012, pp. 288–295.

[3] S. Haddadin *et al.*, "Towards the robotic co-worker," in *Robotics Research*, Springer Berlin Heidelberg, 2011, pp. 261–282.

[4] S. Lemaignan *et al.*, "Artificial cognition for social human–robot interaction: An implementation," *Artificial Intelligence*, vol. 247, pp. 45–69, Jun. 2017.

[5] E. A. Sisbot *et al.*, "A human-aware manipulation planner," *IEEE Trans. Robot.*, vol. 28, no. 5, pp. 1045–1057, Oct. 2012.

[6] J. Mainprice *et al.*, "Planning human-aware motions using a sampling-based costmap planner," in *IEEE Int. Conf. on Robotics and Automation*, May 2011, pp. 5012–5017.

[7] C. Moulin-Frier *et al.*, "DAC-h3: A proactive robot cognitive architecture to acquire and express knowledge about the world and the self," *IEEE Trans. Cogn. Develop. Syst.*, no. 99, pp. 1–1, 2017.

[8] T. Fischer *et al.*, "iCub-HRI: A software framework for complex human–robot interaction scenarios on the iCub humanoid robot," *Frontiers in Robotics and AI*, vol. 5, p. 22, 2018.

[9] C. Moulin-Frier *et al.*, "Embodied artificial intelligence through distributed adaptive control: An integrated framework," *Joint IEEE Int. Conf. on Development and Learning and Epigenetic Robotics*, pp. 324–330, 2017.

[10] D. H. P. Nguyen *et al.*, "Compact real-time avoidance on a humanoid robot for human-robot interaction," in *ACM/IEEE Int. Conf. on Human-Robot Interaction*, ACM, 2018, pp. 416–424.

[11] G. Metta *et al.*, "The iCub humanoid robot: An open-systems platform for research in cognitive development," *Neural Networks*, vol. 23, no. 8, pp. 1125–1134, Oct. 2010.

[12] E. Insafutdinov *et al.*, "Deepercut: A deeper, stronger, and faster multi-person pose estimation model," in *ECCV*, Springer, 2016, pp. 34–50.

[13] G. Pasquale *et al.*, "Teaching iCub to recognize objects using deep convolutional neural networks," in *Machine Learning for Interactive Systems*, 2015, pp. 21–25.

[14] A. Krizhevsky *et al.*, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[15] P. Maiolino *et al.*, "A flexible and robust large scale capacitive tactile system for robots," *IEEE Sensors J.*, vol. 13, no. 10, pp. 3910–3917, 2013.

[16] S. Haddadin *et al.*, "Robot collisions: A survey on detection, isolation, and identification," *IEEE Trans. Robot.*, vol. 33, no. 6, pp. 1292–1312, 2017.

[17] S. Lallee *et al.*, "How? Why? What? Where? When? Who? Grounding ontology in the actions of a situated social agent," *Robotics*, vol. 4, pp. 169–193, 2015.

[18] A. Roncone *et al.*, "Peripersonal space and margin of safety around the body: Learning visuo-tactile associations in a humanoid robot with artificial skin," *PLOS ONE*, vol. 11, no. 10, e0163713, 2016.

[19] G. Vezzani *et al.*, "A grasping approach based on superquadric models," *IEEE Int. Conf. on Robotics and Automation*, pp. 1579–1586, 2017.

[20] A. Makhal *et al.*, "Grasping unknown objects in clutter by superquadric representation," in *2nd IEEE Int. Conf. on Robotic Computing*, 2018, pp. 292–299.

[21] M. Ester *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *2nd Int. Conf. on Knowledge Discovery and Data Mining*, AAAI Press, 1996, pp. 226–231.

[22] U. Pattacini *et al.*, "An experimental evaluation of a novel minimum-jerk cartesian controller for humanoid robots," *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 1668–1674, Oct. 2010.

[23] P. D. Nguyen *et al.*, "A fast heuristic Cartesian space motion planning algorithm for many-DoF robotic manipulators in dynamic environments," in *Humanoid Robots, IEEE-RAS Int. Conf. on*, 2016, pp. 884–891.

[24] W. Kim *et al.*, "Anticipatory robot assistance for the prevention of human static joint overloading in human-robot collaboration," *IEEE Robotics and Automation Letters*, vol. 3, no. 1, pp. 68–75, Jan. 2018.