# HARMONIOUS – Human-like reactive motion control and multimodal perception for humanoid robots

Jakub Rozlivek, Alessandro Roncone, Ugo Pattacini, Matej Hoffmann

*Abstract*—For safe and effective operation of humanoid robots in human-populated environments, the problem of commanding a large number of Degrees of Freedom (DoF) while simultaneously considering dynamic obstacles and human proximity has still not been solved. We present a new reactive motion controller that commands two arms of a humanoid robot and three torso joints (17 DoF in total). We formulate a quadratic program that seeks joint velocity commands respecting multiple constraints while minimizing the magnitude of the velocities. We introduce a new unified treatment of obstacles that dynamically maps visual and proximity (pre-collision) and tactile (post-collision) obstacles as additional constraints to the motion controller, in a distributed fashion over surface of the upper body of the iCub robot (with 2000 pressure-sensitive receptors). This results in a bio-inspired controller that: (i) gives rise to a robot with whole-body visuo-tactile awareness, resembling peripersonal space representations, and (ii) produces human-like minimum jerk movement profiles. The controller was extensively experimentally validated, including a physical human-robot interaction scenario.

## I. INTRODUCTION

As robots are leaving safety fences and starting to share workspaces and even living spaces with humans, they need to function in dynamic and unpredictable environments. Highly redundant platforms like humanoid robots have the possibility of performing tasks even in the presence of many constraints and obstacles. Given the dynamic nature of obstacles and real-time constraints, reactive motion control rather than planning is the solution. The key to success is whole-body awareness drawing on dynamic fusion of multimodal sensory information, inspired by peripersonal space representations in humans.

We present a reactive motion controller that commands the upper body of the iCub humanoid robot ($2 \times 7$ Degrees of Freedom (DoFs) in every arm; 3 torso DoFs), whereby the two hands can have separate tasks in Cartesian space or a common task (see Fig.1). Unique to our approach, obstacles perceived in three different modalities—visual and proximity (pre-collision) and tactile (post-collision)—are dynamically aggregated and projected first onto locations on individual

Corresponding author: Matej Hoffmann (matej.hoffmann@fel.cvut.cz).

Jakub Rozlivek and Matej Hoffmann are with Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague, Czech Republic.

Alessandro Roncone is with Human Interaction and RObotics (HIRO), Department of Computer Science, University of Colorado Boulder, Boulder, CO 80309 USA.

Ugo Pattacini is with iCub Tech, Istituto Italiano di Tecnologia, 16163 Genova, Italy.



(a) The robot is approaching the correct card while back bending to dodge the collision.
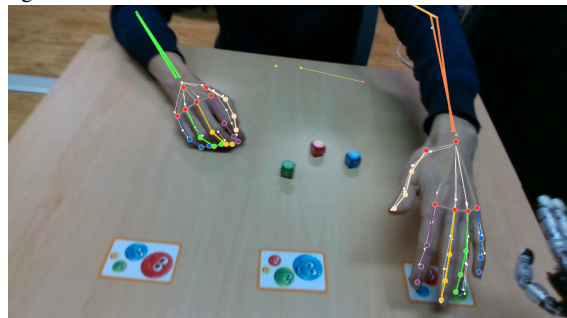


(b) The robot is keeping a safe distance from the opponent's detected hand.

Fig. 1: iCub robot in action during an interactive game demonstration.

body parts and then remapped as constraints into the joint velocity space.

**Contributions.** We have developed *HARMONIOUS*, a real-time reactive motion control system for upper body control of a humanoid robot sensing and avoiding contact with humans in its close proximity. Our specific contributions are the following.

1) *HARMONIOUS* is human-like in two important aspects: (i) it employs multimodal sensing around the whole body of the robot, "visuo-tactile awareness" resembling peripersonal space representations; (ii) it produces minimum jerk movement profiles which are characteristic of human motion. Together, these provide a basis for safe and natural human-robot interaction (HRI)—physical and social.

2) We developed a unified representation of the space around the robot that feeds a reactive motion controller. Dynamically moving obstacles ranging from physical contact with the robot (zero distance, perceived through touch) to obstacles in close or far proximity can be remapped onto the robot's body parts, weighted, and

transformed into kinematic constraints. Thus, there is a dense and multimodal representation of a peripersonal space analogue, which we refer to as the "perirobot space" (see [1] for more details). It is demonstrated here by combining touch, proximity and visual sensors, but other contact-related or range-based sensors could be used.

3) We have extensively and experimentally evaluated *HARMONIOUS* and demonstrated its superior performance to state-of-the-art robot controllers where possible ([2], [3], [4]). Moreover, to our knowledge, this is the first work to show real-time control of a humanoid robot upper torso (in total 17 DoFs, with the possibility of different tasks for the two arms) faced with tens of dynamically moving obstacles around the whole robot body surface. Furthermore, kinematic singularities are handled through velocity damping and preferred postures are rewarded in the optimization problem formulation (see Tab. I for an overview). As the final evaluation, we tested *HARMONIOUS* in an interactive board game scenario with the human player dynamically perceived, simultaneously processing keypoints on its body, proximity signals, and physical contacts on the whole robot body.

4) The problem formulation is highly modular and both the minimization criteria (e.g., motivating preferred postures) or constraints to the quadratic program (e.g., obstacles) can be easily removed or added on the run.

## II. RELATED WORK

### A. Inverse kinematics

Robot tasks are defined in *Cartesian* (also called *operational* or *task*) *space* while they are commanded in joint space, some form of *inverse kinematics (IK)* mapping ($\mathbf{q} = f(\mathbf{x})$) is indispensable. For industrial manipulators, a closed-form analytical solution is often available, especially for 6 DoF arms with a so-called spherical wrist. Closed-form solutions are global and fast to compute. However, such analytical solutions exploit specific geometric relationships and are generally not available for redundant robots with an arbitrary kinematic structure ([5] is an exception). Numerical approaches to IK come with a higher computational load, but are more general and applicable to any kinematic chain. TRAC-IK [6] is a state-of-the-art IK solver that solves IK as a sequential quadratic programming problem. Lloyd et al. [7] presented an IK algorithm that uses the third-order root finding method to increase the speed and robustness of finding IK solutions. The solution consists of a set of target joint angles only—smooth trajectory generation and execution have to be taken care of separately. Pattacini et al. [8] combined a trajectory generator, a non-linear IK solver, and a Cartesian controller to ensure human-like minimum jerk movements in both Cartesian and joint space. None of the above methods can handle dynamic obstacles in the workspace. An overview of the related work from this and the following sections is in Tab. I.

### B. Obstacle avoidance and differential kinematics

Collision avoidance in static environments can be resolved at a higher level by using trajectory optimization for robot motion planning [20], [21], [22]. However, currently, these methods cannot provide solutions for scenarios like humanoid robots interacting with dynamic human-populated environments in real time. The family of methods that alleviate the above-mentioned problems and that are gaining popularity recently rely on differential kinematics, i.e. the formulation of the operational space task in velocity space and a mapping between desired Cartesian velocity and joint velocity through the robot Jacobian. Differential kinematics, also referred to as Resolved-rate motion control (RRMC), dates back to Whitney [23]. This method is inherently local—depends on the current robot configuration and the corresponding Jacobian—and as it provides joint velocities, it can be directly employed for motion control (without separate trajectory generation).

Inverse differential kinematics methods rely on some form of Jacobian inverse ($\dot{\mathbf{q}} = \mathbf{J}^{-1}(\mathbf{q})\dot{\mathbf{x}}$), more often pseudo-inverse when $J$ is not square, like for redundant robots. The damped least squares (DLS) method [9], [10], [11] is preferred over Jacobian pseudo-inverse or transpose as it "dampens" the motions in the vicinity of kinematic singularities. To dodge obstacles in redundant robots, self-motion, i.e. the possibility to use the *null space* of the solution for additional tasks, is exploited [24], [25], [26], [27]. The advantage of these approaches is that the problem has a clear hierarchical structure that allows to set priorities. For example, Albini et al. [24] define obstacle avoidance as the primary task and the reaching target uses the null space. The disadvantage is the limited flexibility of the problem formulation in face of many (some dynamically appearing) constraints or their changing priorities.

Alternatively, the motion control task can be defined using the forward differential formulation, searching for joint velocities $\dot{\mathbf{q}}$ that best match the desired Cartesian velocity of the end effector $\dot{\mathbf{x}}$, thus minimizing $||\dot{\mathbf{x}} - \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}||$. To this end, it is possible to use quadratic programming (QP) optimization with a quadratic cost function and linear constraints [2], [12], [13], [14], [15], [16], [28]. To avoid singular configurations, two options are usually given: add manipulability maximization to the cost function [2], [15] or dampen joint velocities when close to singularities [12], [13], [29]. In addition, undesirable configurations can be avoided by extending the optimization cost function to motivate the robot to be close to the middle of the joint limits [12] or close to the robot resting position [8]. In the case of the QP problem, obstacles are represented as linear inequality constraints allowing motions in tangential directions improving target reachability [2], [12], [13].

Most of the mentioned works do not solve self-collisions explicitly. The exceptions are [13], where self-collisions are solved by adding additional obstacle points for robot links and the end effector, and [17], where a neural network is trained to favor configurations far from self-collision states. Moreover, De Santis et al. [30] presented an algorithm for real-time generation of self-collsion avoidance motions using repulsion forces converted to avoidance torques. Unlike most

| Work | Control problem | React. obs. avoidance | Sensory modalities to perceive obstacles | DoF shown | Singularities handling | Pref. posture motivation | Ori. error minim. | Explicit traj. samp. |
|---|---|---|---|---|---|---|---|---|
| **Ours** | DiffK - QP problem | Lin. ineq. constr | 3 - vision, proximity, touch | 17 | Velocity damping | Yes | Yes | Yes |
| [6] | IK - Sequential QP | - | - | 15 | No | No | Yes | No |
| [7] | IK - Jacobian based | - | - | 16 | Damped variant | No | Yes | No |
| [8] | IK - Nonlinear opt. | - | - | 7 | No | Yes | Yes | Yes |
| [9] | DiffK - Damped LS | Lin. ineq. constr. | 0 - known obstacles | 8 | Damped LS | No | Yes | No |
| [10] | DiffK - Pseudoinverse | Potential field | 0 - known obstacles | 7 | No | No | Yes | Yes |
| [11] | DiffK - Pseudoinverse | Repulsive vector | 1 - vision | 7 | No | No | No | No |
| [2] | DiffK - QP problem | Lin. ineq. constr. | 0 - known obstacles | 7 | Max. manipulability | No | Yes | No |
| [12] | DiffK - QP problem | Lin. ineq. constr. | 2 - proximity, touch | 7 | Velocity damping | Yes | No | No |
| [13] | DiffK - QP problem | Lin. ineq. constr. | 1 - proximity | 7 | Velocity damping | No | No | No |
| [14] | DiffK - QP problem | Lin. ineq. constr. | 0 - known obstacles | 7 | No | No | Yes | No |
| [15] | DiffK - QP problem | - | - | 12 | Max. manipulability | No | Yes | No |
| [16] | DiffK - QP problem | Lin. ineq. constr. | 0 - known obstacles | 6 | No | No | Yes | No |
| [17] | IK - Nonlinear opt. | Cost function | 0 - known obstacles | 8 | Max. condition number | No | Yes | No |
| [18] | Operational space control | Potential field | 0 - known obstacles | 6 | No | No | Yes | No |
| [19] | ERG + PD control | Potential field | 1 - vision | 7 | No | No | Yes | No |
| [3] | IK - Nonlinear opt. | Repulsive vector | 2 - vision, touch | 10 | No | No | Yes | Yes |

TABLE I: Comparison of different approaches to solving the inverse kinematics problem. Control problem acronyms are: differential kinematics (DiffK), inverse kinematics (IK), quadratic programming (QP), least squares (LS), and explicit reference governor (ERG).

of the mentioned controllers that presented solutions of the IK problem for a single-arm robot, Tong et al. [15] proposed a four-criterion-optimization coordination motion scheme for a dual-arm robot that deals with coordination constraints and physical constraints. However, their approach does not consider obstacle avoidance and the torso joints of the dual-arm robot.

To represent obstacles, artificial potential fields [18], where obstacles are repulsive surfaces that repel the robot from the obstacle, can be used [19], [31], [32]. Park et al. [10] presented a dynamic potential field for smoother avoidance movements. Repulsive vectors in Cartesian space are shown in [11] and later used in [3], [4].

*C. Human-like motion generation*

Several controllers can work as explicit local trajectory generators to sample the trajectory between the start and the target pose. The algorithm in [32] combines control with path planning and can handle arbitrary desired velocity profiles for the robot. The Cartesian controller [8] produces human-like quasi-straight trajectories with minimum-jerk profiles characteristic of human motions. A reactive controller in [3], [4] uses the minimum-jerk trajectory sampling as in [8]. Suleiman [16] proposed an algorithm for implicit minimum jerk trajectories by replacing joint velocities by joint jerk as control parameters in the optimization problem.

*D. Sensing of obstacles and contacts*

To make safe movement of a robot in a cluttered or human-populated environment possible, obstacles need to be perceived in real time and processed either before a collision happens or immediately after contact with the robot. We will focus on perception in the context of HRI.

To sense at a distance, RGB-D cameras are frequently adopted [11], [33], [34], [35], [36]. Depth information can be used to compute a set of spheres that represent obstacles [35] or to create a depth space representation that is used to generate repulsive vectors for the robot [11]. Magrini and De Luca [26] estimated the contact force based on the contact point detected by a Kinect sensor. Alternative visual sensing for obstacle detection are tracking systems that follow markers attached to specific body parts, e.g., a wrist [32] or arm joints and a head [19]. Aljalbout et al. [37] used a static RGB camera in simulation and trained a convolutional neural network for obstacle avoidance.

Collisions with a moving robot may be acceptable provided that the contact forces are within limits. Collisions need to be detected, located, identified (e.g., determining the forces), and, possibly following classification of the collision, reacted upon (e.g., stop, slow down, retract, etc.) [38]. Force/torque sensors within the robot structure, together with models of its dynamics, can be employed. Alternatively, contacts can be perceived through tactile sensors. We will focus on large-area sensitive skins. Unlike when force/torque sensors are employed, collision localization/isolation is easier, even for multi-contacts. Cirillo et al. [25] developed a flexible skin, allowing measurement of contact position and three components of contact force. Calandra et al. [39] presented a data-driven method for a fast and accurate prediction of joint torques from contacts detected by tactile skin. Albini et al. used robot skin feedback for HRI [40] as well as robot control in a cluttered environment [24]. Kuehn and Haddadin [41] introduced an artificial robot nervous system, which integrates tactile sensation and reflex reactions into robot control through the concept of robot pain sensation. Some large-area sensitive skins have been safety-rated and are deployed in collaborative robot applications (see Airskin and the analysis of its effects in [42]).

Individual visual sensors—on the robot itself or in other locations in the workspace—are prone to occlusions (which can be mitigated by employing multiple sensors [43]). Large area tactile arrays are distributed over the whole robot body, but only respond after physical contact. Distributed proximity sensors provide an alternative: sensing at a small distance over the whole robot body. Proximity [12], [13] or tomographic [44] sensors can be used for anticipation of collisions and contactless motion guidance.

Typically, only one sensor type is used, and the corresponding robot control is more or less dependent on the nature of

the sensory information. On the other hand, there is a field of multimodal sensor fusion. Dean-Leon et al. [45] presented a fusion of pressure and proximity data from multimodal artificial skin to change the robot's dynamic behavior, e.g., making a stiff robotic system compliant. Hierarchical control is often used together with the integration of several sensors, such as an artificial skin for collision avoidance with a camera for position-based visual servoing [46], or a LIDARr for collision avoidance and a multimodal skin for tactile interaction or distance/force control [47]. In addition to collision avoidance, fusion can be used for other tasks, such as opening the door [48]. Multimodal fusion can integrate more than sensor data as in [49], where contact location is estimated using a contact hypotheses fusion approach from multiple sensor modalities, the context of interaction, and the environment.

Humans posses a dynamically established protective safety margin around the whole body, drawing on visuo-tactile interactions: Peripersonal space (PPS), e.g., [50]. Roncone et al. [51], [52] took inspiration from PPS and implemented visuo-tactile receptive fields on a humanoid robot. Nguyen et al. [3], [4] deployed this in HRI to feed a reactive motion controller. As affordable and increasingly accurate sensors of different types are becoming ubiquitous, it is important and timely to develop a unified treatment of their signals such that they can be processed by a single motion controller (see also [1] where a unified treatment of the perirobot space for collaborative robot applications is proposed).

### E. Comparison of HARMONIOUS with related works

In this work, we present a framework in which RGB-D, proximity, and tactile sensors are simultaneously processed, related to the complete robot body, and used to generate real-time constraints for a whole-body reactive motion controller.

To the extent that this is possible, the main components of our solution are categorized and contrasted with representative works from the literature in Tab. I.

We formulate the control problem (second column in Tab. I) as a quadratic program in velocity space—employing differential kinematics, similarly to many recent works in this area [2], [12], [13], [14], [15], [16]. Reactive obstacle avoidance (second column) is implemented using a linear inequality constraint (LIC). Importantly, this work is to our knowledge the first to employ three different sensory modalities (vision, proximity, touch), in a unified manner, to perceive and avoid obstacles (fourth column). The fifth column (DoF shown) lists the number of DoF employed in our approach and in the literature. HARMONIOUS handles kinematic singularities through velocity damping. The robot is motivated to favor joint positions near a preferred posture, e.g., the center of the joint ranges or a home configuration to prevent undesirable configurations, unlike most of the works in the literature. HARMONIOUS can handle position and orientation targets (Orientation error minimization) and samples the trajectory of end-effector positions (Explicit trajectory sampling), inspired by human reaching movements—quasi-straight trajectories with bell-shaped velocity profiles. The end-effector orientation sampling is calculated using a computer graphics method *slerp*

for constant angular velocities, ensuring smooth rotations and the shortest path between initial and final orientations. In summary, HARMONIOUS employs state-of-the-art control problem and reactive obstacle avoidance formulation, but moves beyond the state of the art in providing a unified treatment of three different sensory modalities, the number of DoFs controlled, and incorporation of preferred position motivation and orientation error minimization. Beyond what could fit in Tab. I, HARMONIOUS also features self-collision avoidance and the possibility of setting targets (tasks) for the two robot arms independently or jointly (bimanual task). To our knowledge, HARMONIOUS is unparalleled in the scale of the problem and the number of features it can tackle.

### III. MATERIALS AND METHODS

Here we present HARMONIOUS – Human-like reactive motion control and multimodal perception for humanoid robots. An overview is provided in Fig. 2. The core of the solution is the motion controller (red box) and the actual QP formulation (magenta) which is online fed by the target trajectory generator (blue box) and the constraints projected from obstacles (green box).

### A. Experimental setup

The experiments were conducted with the humanoid robot iCub ([53], see Fig. 1), specifically on its upper body with 17 degrees of freedom (DoF) consisting of two 7-DoF arms and a common 3-DoF torso. The robot was equipped with an RGB-D camera Intel RealSense D435 placed above his eyes to track obstacles in 3D, an artificial electronic skin [54] to detect contacts and their intensity, and two proximity sensor units [55] with time-of-flight sensors that improve perception in "blind spots" (back of hands).

### B. Motion controller

The main component of our approach is a motion controller (see the red box in Fig. 2). Our motion controller is based on reactive kinematic control, in particular the approach called differential kinematics which is defined as

$$\boldsymbol{\nu} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}} \tag{1}$$

where $\boldsymbol{\nu} \in \mathbb{R}^m$ is the spatial velocity of the end effector ($m$ is the task dimension—6 for single-arm version, 12 for dual-arm version), $\dot{\mathbf{q}} \in \mathbb{R}^n$ are the joint velocities of the robot ($n$ is the number of joints) and $\mathbf{J}(\mathbf{q}) \in \mathbb{R}^{m \times n}$ is the Jacobian of the robot for the joint positions $\mathbf{q} \in \mathbb{R}^n$. This relation represents a trajectory constraint between the robot configuration and Cartesian space. In this work, we assume the reference frame for Cartesian coordinates and Jacobians is the base of the robot.

The spatial velocity of the end effector is computed from the current end-effector pose and the desired end-effector pose in the next time period as

$$\boldsymbol{\nu}^{pos} = \frac{\mathbf{x_t} - \mathbf{x_c}}{t_s}, \qquad \boldsymbol{\nu}^{ori} = \frac{\varphi(\mathbf{R_t}\mathbf{R_c}^T)}{t_s} \tag{2}$$
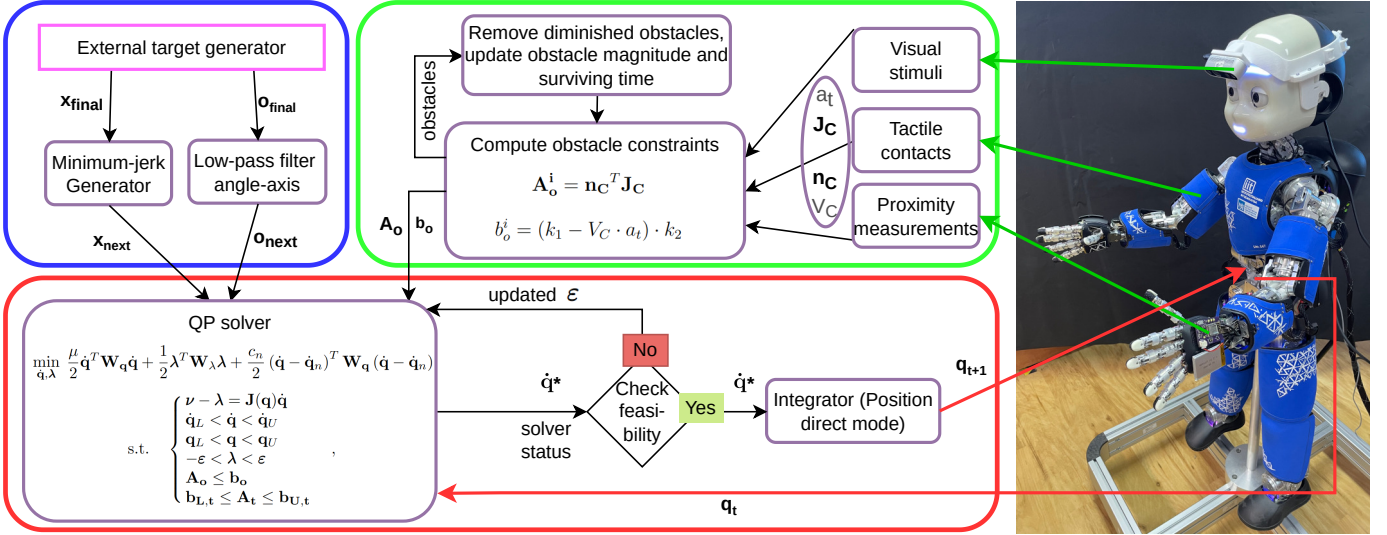
Fig. 2: *HARMONIOUS* – overview. The blue box represents the local trajectory sampling. The trajectory towards the target pose is sampled using a minimum-jerk generator (position) and a low-pass filter (orientation). The target pose is received from an external target generator. The green box represents the obstacle processing. It takes inputs from proximity sensors, skin parts (tactile stimuli), and RGB-D camera (visual stimuli) and computes linear inequality constraints to dodge obstacles or react to the collision that occurred. The obstacle exists virtually for a specific surviving time with a decreasing threat level, giving humans time to react. The red box represents the controller with a QP solver that solves the differential kinematics problem. It takes the next desired pose and current joint positions as input. If the problem is feasible, the computed joint velocities are integrated into new joint positions and sent to the robot. Otherwise, the position constraint is relaxed, and the QP problem is solved again.

where $\boldsymbol{\nu}^{pos} \in \mathbb{R}^3$ is the translational velocity of the end effector, $\mathbf{x_c} \in \mathbb{R}^3$ and $\mathbf{x_t} \in \mathbb{R}^3$ are the current and desired next end effector position, $\boldsymbol{\nu}^{ori} \in \mathbb{R}^3$ is the rotational velocity of the end effector, $\mathbf{R_c} \in \mathbb{R}^{3\times3}$ and $\mathbf{R_t} \in \mathbb{R}^{3\times3}$ are the current and desired next end effector orientation (converted from axis-angle representation), $\varphi$ represents mapping from a rotation matrix to an axis-angle representation to obtain angular velocities using a period time $t_s$ exploiting the infinitesimal rotation assumption (see e.g., [2]).

For our purposes, we can relax the trajectory constraint (the "reaching task") by adding a slack variable $\boldsymbol{\lambda} \in \mathbb{R}^m$ to help the controller with finding a solution when other (e.g., safety) constraints appear:

$$\boldsymbol{\nu} - \boldsymbol{\lambda} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}} \tag{3}$$

Then, we formulate the problem as a strictly convex QP problem with joint velocities $\dot{\mathbf{q}}$ and slack variables $\boldsymbol{\lambda}$ as optimization variables. In our cost function, we minimize the sum of the squared Euclidean norm of the joint velocities, the distance from the preferred joint positions, and the squared Euclidean norm of the vector of slack variables. It is formulated as:

$$\min_{\dot{\mathbf{q}},\boldsymbol{\lambda}} \frac{\mu}{2}\dot{\mathbf{q}}^T\mathbf{W_q}\dot{\mathbf{q}} + \frac{1}{2}\boldsymbol{\lambda}^T\mathbf{W_\lambda}\boldsymbol{\lambda} + \frac{c_h}{2}\left(\dot{\mathbf{q}} - \dot{\mathbf{q}}_n\right)^T\mathbf{W_q}\left(\dot{\mathbf{q}} - \dot{\mathbf{q}}_n\right) \tag{4}$$

where diagonal matrices $\mathbf{W_q} \in \mathbb{R}^{n\times n}$ and $\mathbf{W_\lambda} \in \mathbb{R}^{m\times m}$ contain the weights of the individual joints and the relaxations for the trajectory constraint, $\dot{\mathbf{q}}_n$ are the velocities needed to reach the preferred (e.g., home) robot configuration, $c_h$ is a

weight of the motivation to keep robot close to a preferred configuration, and $\mu$ is a damping factor calculated as:

$$\mu = \begin{cases} (1 - \frac{\omega}{\omega_0})^2 + 0.01 & \omega < \omega_0, \quad \omega_0 \text{ is a threshold,} \\ 0.01 & \text{otherwise.} \end{cases} \tag{5}$$

where $\omega$ is a manipulability index [56] computed as $\omega = \sqrt{\det(\mathbf{J}(\mathbf{q})\mathbf{J}(\mathbf{q})^T)} = s_1 s_2 ... s_n$, where $s_i$ is the i-th singular value of $\mathbf{J}(\mathbf{q})$. This damping factor increases when the manipulability index decreases to zero, implying that the robot is close to a singularity position (the manipulability index is zero in that position). The higher damping factor makes the minimization of joint velocities more important, leading to lower joint velocities and singularity avoidance [29]. We assume that the weights ($\mathbf{W_q}$ $\mathbf{W_\lambda}$, $c_h$, and $\mu$) are non-negative to ensure that the problem is convex. The weights in $\mathbf{W_q}$ corresponding to the torso joints are higher than for the arm joints, which, through the minimization term, motivates smaller torso movements, which is more natural.

Apart from the mentioned relaxed trajectory constrained, the minimization is done with respect to several other constraints, such as joint velocity and position bounds (see Sec. III-K), slack variable bounds, collision avoidance (see Sec. III-F), and other task or kinematic constraints (see Sec. III-J). The

constraints can be written as follows:

$$\text{s.t.} \begin{cases} \boldsymbol{\nu} - \boldsymbol{\lambda} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}} \\ \dot{\mathbf{q}}_L < \dot{\mathbf{q}} < \dot{\mathbf{q}}_U \\ \mathbf{q}_L < \mathbf{q} < \mathbf{q}_U \\ -\boldsymbol{\varepsilon} < \boldsymbol{\lambda} < \boldsymbol{\varepsilon} \\ \mathbf{A_o}\dot{\mathbf{q}} \leq \mathbf{b_o} \\ \mathbf{b_{L,t}} \leq \mathbf{A_t}\dot{\mathbf{q}} \leq \mathbf{b_{U,t}} \end{cases} \tag{6}$$

where $\dot{\mathbf{q}}_{L,U} \in \mathbb{R}^n, \mathbf{q}_{L,U} \in \mathbb{R}^n, \boldsymbol{\varepsilon} \in \mathbb{R}^m$ are the lower and upper bounds for the joint velocities, the joint positions, and the slack variables, respectively. $\mathbf{A_o} \in \mathbb{R}^{m_o \times n}$ and $\mathbf{b_o} \in \mathbb{R}^{m_o}$ define $(m_o)$ obstacles avoidance constraints; $\mathbf{A_t} \in \mathbb{R}^{m_t \times n}$, $\mathbf{b}_{L,t} \in \mathbb{R}^{m_t}$, and $\mathbf{b}_{U,t} \in \mathbb{R}^{m_t}$ define other $(m_t)$ task constraints.

*C. Dual-arm trajectory constraint*

Here, we formulate the problem for two arms with a common torso. Since we have a common torso of the robot, we cannot calculate the trajectory constraints for the arms independently of each other. Moreover, it is not always possible to reach targets with both arms simultaneously, conflict situations are managed by prioritizing one of the arms and calling it the "primary arm", and the other is then the "secondary arm". Therefore, the trajectory constraint (Eq. 3) for a robot with a common torso is expanded as:

$$\begin{aligned} \boldsymbol{\nu}_p^{pos} - \boldsymbol{\lambda}_p^{pos} &= \mathbf{J}_{\mathbf{t,pos}}^{\mathbf{P}}(\mathbf{q^t})\dot{\mathbf{q}}^{\mathbf{t}} + \mathbf{J}_{\mathbf{a,pos}}^{\mathbf{P}}(\mathbf{q^P})\dot{\mathbf{q}}^{\mathbf{P}} \\ \boldsymbol{\nu}_p^{ori} - \boldsymbol{\lambda}_p^{ori} &= \mathbf{J}_{\mathbf{t,ori}}^{\mathbf{P}}(\mathbf{q^t})\dot{\mathbf{q}}^{\mathbf{t}} + \mathbf{J}_{\mathbf{a,ori}}^{\mathbf{P}}(\mathbf{q^P})\dot{\mathbf{q}}^{\mathbf{P}} \\ \boldsymbol{\nu}_s^{pos} - \boldsymbol{\lambda}_s^{pos} &= \mathbf{J}_{\mathbf{t,pos}}^{\mathbf{s}}(\mathbf{q^t})\dot{\mathbf{q}}^{\mathbf{t}} + \mathbf{J}_{\mathbf{a,pos}}^{\mathbf{s}}(\mathbf{q^s})\dot{\mathbf{q}}^{\mathbf{s}} \\ \boldsymbol{\nu}_s^{ori} - \boldsymbol{\lambda}_s^{ori} &= \mathbf{J}_{\mathbf{t,ori}}^{\mathbf{s}}(\mathbf{q^t})\dot{\mathbf{q}}^{\mathbf{t}} + \mathbf{J}_{\mathbf{a,ori}}^{\mathbf{s}}(\mathbf{q^s})\dot{\mathbf{q}}^{\mathbf{s}} \end{aligned} \tag{7}$$

where the subscripts $t, p, s$ represent **t**orso, "**p**rimary arm", and "**s**econdary arm" respectively. The superscripts $pos, ori$ represent the relaxation of the constraint of the **pos**ition and **ori**entation trajectory, respectively. The single-arm formulation contains only the torso and the "primary arm" part. $\mathbf{J}_{\mathbf{a,pos}}^{\mathbf{p/s}}(\mathbf{q^{P/s}}) \in \mathbb{R}^{3 \times (n-3)}, \mathbf{J}_{\mathbf{t,pos}}^{\mathbf{p/s}}(\mathbf{q^t}) \in \mathbb{R}^{3 \times 3}$ are the translational velocity components of the Jacobian matrices for arm and torso, respectively, $\mathbf{J}_{\mathbf{a,ori}}^{\mathbf{p/s}}(\mathbf{q^{P/s}}) \in \mathbb{R}^{3 \times (n-3)}, \mathbf{J}_{\mathbf{t,ori}}^{\mathbf{p/s}}(\mathbf{q^t}) \in \mathbb{R}^{3 \times 3}$ are the rotational velocity components of the Jacobian matrices for arm and torso, respectively.

Consistent with the arm prioritization, we distinguish two cases for the limits of the slack variable $\boldsymbol{\varepsilon}$. They are set to $\infty$ (the deviation is minimized only) or to zero (the equality constraint without relaxation). In our solution, $\boldsymbol{\varepsilon}$ is set to $\infty$ for all $\boldsymbol{\lambda}$ except $\boldsymbol{\lambda}_p^{pos} = [0, 0, 0]^T$. If the equality constraint on the desired translation velocity of the "primary arm" makes the problem infeasible, we solve the problem again with the constraint relaxed to the minimization of the deviation. This strategy allows the robot to reach a target position with higher precision if the original problem is feasible and to make it feasible if the equality constraint is too strict.

*D. Local trajectory sampling*

The first input to the reactive controller is the trajectory waypoint for the next time step (see the blue box in Fig. 2).

Given the desired final target pose from an external source (e.g., a high-level target generator), we compute the desired next-time pose separately in position and orientation. We use a low-pass filter in the axis-angle representation for orientation sampling, which ensures a constant-speed transition between two orientations. For position sampling, a generator of approximately minimum-jerk trajectories is employed. Position samples are generated using a third-order linear time-invariant dynamical system [8].

We distinguish between streamed targets (e.g., reference tracking) and individual (reaching) targets. In the first case, the targets can be skipped (e.g., during collision avoidance), and only the newest one is important, and the local trajectory sampling is not applied as it is assumed that the movement smoothness is handled in the external target generator.

*1) Position sampling:* The sampling of a time-varying input $\mathbf{x}_d \in \mathbb{R}^3$ expressing the desired positions for the end effector in Cartesian space is performed by means of a third-order filter, which is capable of generating output trajectories that exhibit the property of being quasi minimum-jerk, thus resembling the human-like movements as described in [57].

We tightly follow the implementation presented in [8], where the idea is to start from the feedback formulation of the minimum-jerk trajectory as the solution of the optimal problem proposed in [58]. The exact solution resorts to a third-order linear time-variant (LTV) dynamical filter applied to each coordinate $x_d(*)$ of the vector $\mathbf{x}_d$:

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dddot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ \frac{-60}{(T-t)^3} & \frac{-36}{(T-t)^2} & \frac{-9}{T-t} \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{60}{(T-t)^3} \end{bmatrix} x_d(*) \tag{8}$$

where $T$ represents the execution time.

To circumvent the difficulty posed by coefficients becoming unbounded for $t \to T$, the authors ran in [8] a second optimization to find out the third-order linear time-invariant (LTI) system that can deliver the best approximation of the output of Eq. 8, minimizing the jerk measure over the same temporal interval $[0, T]$.

The resulting LTI system is:

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dddot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ \frac{a}{T^3} & \frac{b}{T^2} & \frac{c}{T} \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -\frac{a}{T^3} \end{bmatrix} x_d(*) \tag{9}$$

where $a \approx -150.766$, $b \approx -84.981$, and $c \approx -15.967$ are the constant coefficients found by the optimization, and $T$ is a parameter (related to the execution time) that regulates the reactivity of the filter. Throughout our experiments, we set $T = (\|\mathbf{x_d} - \mathbf{x_0}\|/v_t)$ s, where $\mathbf{x_0} \in \mathbb{R}^3$ is the starting position, and $v_t$ is the desired Cartesian speed of movement.

The output $\mathbf{x}(t) \in \mathbb{R}^3$ of the filter in Eq. 9 provides a sampling of the input $\mathbf{x}_d(t)$ that is "quasi" minimum-jerk, being the filter only an approximation of the system (Eq. 8), although the authors in [8] proved that it is reliable and effective in smoothing out sharp transitions while guaranteeing bell-shaped velocity profiles.

*2) Orientation sampling:* We use the axis-angle representation for the orientations of the end effector. The orientation is described by a rotation vector $\mathbf{r} = [r_1, r_2, r_3]^T$, that encodes the axis of rotation $u = \mathbf{r}/||\mathbf{r}||$ and a rotation angle $\Theta = ||\mathbf{r}||$. Orientation sampling is carried out by interpolation between two orientations $\mathbf{r_1}, \mathbf{r_2}$ using a spherical linear interpolation, i.e., Slerp [59], which takes advantage of the fact that the spherical metric of $S^3$ is the same as the angular metric of $SO(3)$ for a constant angular speed of movement [59]. Moreover, it travels along the straightest (and shortest) path on the rounded surface of the quaternion unit sphere.

We first start by creating skew matrices $\mathbf{r_{+,1}}, \mathbf{r_{+,2}}$ from rotation vectors $\mathbf{r_1}, \mathbf{r_2}$. These matrices can be easily converted to rotation matrices $\mathbf{R_1}, \mathbf{R_2}$ by computing their matrix exponentials (*expm* function):

$$\mathbf{R_1} = \text{expm}(\mathbf{r_{+,1}}), \qquad \mathbf{R_2} = \text{expm}(\mathbf{r_{+,2}})$$

We compute the interpolation between rotations $\mathbf{R_1}, \mathbf{R_2} \in \mathbb{R}^{3\times3}$ as

$$\text{slerp}(\mathbf{R_1}, \mathbf{R_2}, \alpha) = \text{expm}\left(\alpha \log\text{m}\left(\mathbf{R_2}\mathbf{R_1^T}\right)\right)\mathbf{R_1}, \quad (10)$$

where the function *logm* converts the rotation matrices to the skew matrices of the rotation vectors and $\alpha$ is an interpolation coefficient given as a period time $t_s$ divided by the trajectory time $T = (||\mathbf{x_d} - \mathbf{x_0}||/v_t)$ s. Finally, the computed rotations are converted back to the axis-angle representation.

### E. Peripersonal space projection

Roncone et al. [51], [52] proposed a distributed representation of the protective safety zone for humanoid robots, called Peripersonal space (PPS). The safety zone is represented by a collection of probabilities that an object from the environment eventually comes into contact with that particular skin taxel. For each taxel, they combined visual information of objects approaching the body and tactile information of eventual physical collision. A visual receptive field around the taxel is represented by a cone that rises from the surface along the normal.

Nguyen et al. [3], [4] made small modifications to PPS and used it to ensure safe interaction between humans and robots. Instead of training the visual receptive fields, they defined them uniformly for all taxels. In addition, they extended the reach of the receptive field to a maximum of 45 cm. Due to compatibility with the original implementation, the taxel receptive field has a discrete representation with 20 bins representing the distance from the obstacle with values of collision probabilities. The Parzen window estimation algorithm interpolates the bins to create a continuous representation. A combination of the receptive fields constructs a safety volume margin around each body part. In this work, we added receptive fields to the upper arm and torso to cover the entire upper body.

### F. Obstacle processing

The processed obstacles are the second input to the reactive controller (see the green box in Fig. 2). In our setup, we operate with obstacles detected by visual, proximity, and tactile sensors. However, our approach is universal in modalities. The processed obstacles, called collision points ($C$), are represented by their projected position on the robot $\mathbf{P}_C \in \mathbb{R}^3$, collision direction $\mathbf{n}_C \in \mathbb{R}^3$, threat level $a_t$ (i.e., scaled sensor measurement) and a gain $V_C$ representing the severity of the information; for example, tactile events can have the highest gain as the collision has already happened. The main advantage of our representation is that it jointly enables the post-collision reaction (tactile sensors) and collision avoidance (proximity and visual sensors).

An obstacle is incorporated into the QP problem as a linear inequality constraint that restricts movements towards the obstacle but allows motion in tangential directions. The constraint limits the approach motion of the collision point $C$ with Cartesian velocity $\dot{\mathbf{x}}_C \in \mathbb{R}^3$ towards the obstacle (in direction $\mathbf{n}_C$) by a maximum approach velocity $\dot{x}_a$ (which can be negative and become the repulsive velocity, Sec. III-G) as

$$\mathbf{n}_C^T\dot{\mathbf{x}}_\mathbf{C} = \mathbf{n}_C^T\mathbf{J}_C\dot{\mathbf{q}}_\mathbf{C} < \dot{x}_a \qquad (11)$$

where $\mathbf{J}_C \in \mathbb{R}^{3 \times n_c}$ is the translational-velocity component of the Jacobian associated with $C$ and $\dot{\mathbf{q}}_C \in \mathbb{R}^{n_c}$ are the joint velocities of the kinematic chain ending with $C$. The maximum approach velocity $\dot{x}_a$ is calculated from the obstacle threat level $a_t$ as

$$\dot{x}_a = (k_1 - V_C \cdot a_t) \cdot k_2 \qquad (12)$$

where $k_1$ and $k_2$ are positive values to adjust the constraint which are described in Sec. III-G. In accordance with Eq. 4, we can write the obstacle constraints in matrix form ($\mathbf{A_o^i} \leq b_o^i$) as

$$\mathbf{n_C}^T\mathbf{J_C} \leq (k_1 - V_C \cdot a_t) \cdot k_2. \qquad (13)$$

To prevent jerky motions induced by vanishing obstacle readings, we apply a linear decay formula to simulate an obstacle slowly moving away from the robot [12]. Every obstacle virtually exists for a specific amount of time (called "surviving time"). For example, the threat disappears once the robot moves away from the tactile collision. Without virtual preservation of the obstacle, the robot would return to its original position even when the obstacle is still there.

Visual obstacles in our experiments are human keypoints detected by an RGB-D camera. The method for their detection is based on the work by Docekal et al. [36]. The detected obstacles (see a schematic illustration in Fig. 3a) are mapped to parts of the skin where a collision is most likely to occur using a PPS projection, similar to [4]. For each body part, we compute one visual obstacle as a weighted average of the mappings to that body part. The collision direction is the vector from the estimated contact position on the skin to the obstacle and the threat level is calculated from their distance.

Proximity sensor data (see a schematic illustration in Fig. 3b) are not projected onto skin taxels, as these sensors are placed to cover parts of the robot without the skin; therefore, there is no skin taxel to which information can be projected. However, they are projected onto the sensor position. The threat level is computed from the measured distance, and the collision direction is the direction vector of the sensor beam.
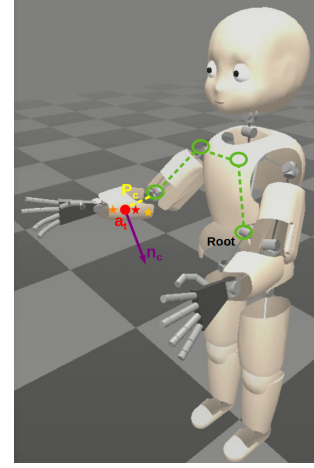
Tactile obstacles (see a schematic illustration in Fig. 3c) are processed in the same way as in [3]. Contacts detected
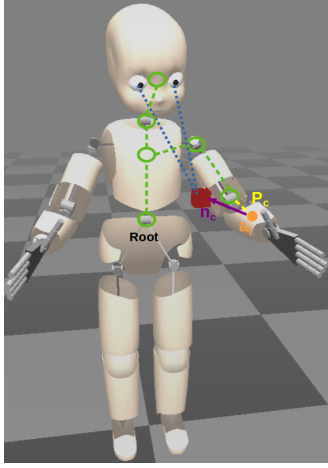
(a) Visual obstacle projected onto the left forearm. The blue lines symbolize the lines of sight of the eyes, not those of the RGB-D camera for simplification.
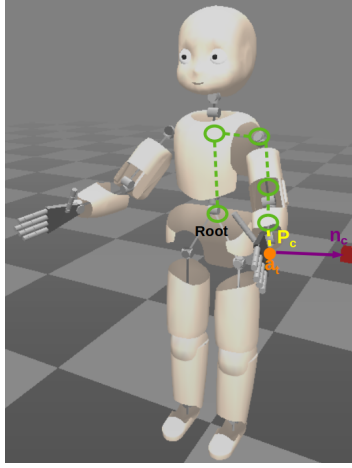
(b) Proximity obstacle mapped to a sensor position on the left hand.

(c) Several tactile stimuli on the right forearm (stars) combined into a super contact $P_C$ with a threat level of the highest measured pressure (red color).

Fig. 3: Processing of obstacles. Obstacles (red cubes or stars) are projected onto the robot body ($P_C$) and connected (yellow line) to the robot kinematic chain (green line). Each collision point is characterized by a normal direction (purple arrow) and threat level (color of the circle, red means highest threat).

by artificial skin sensors (taxels) with a pressure above the threshold value are clustered to super contacts (one per each body part) to diminish spiking effects and eliminate false positive detections. The threat level corresponds to the highest measured pressure in the neighborhood, and the collision direction is the normal vector of the central taxel.

### G. Collision avoidance parameters

Obstacle avoidance constraints (Eq. 12) are parameterized by two positive parameters—additive $k_1$ and multiplicative $k_2$. The parameter $k_1$ defines when the maximum approach velocity becomes the minimum repulsive velocity, that is, when $\dot{x}_a$ is lower than zero. The parameter $k_1$ has a constant value of 0.3 in our experiments. The parameter $k_2$ scales the maximum approach velocity. We found that body links cannot move with the same minimum repulsive velocity due to the kinematic constitution of the robot. The same scale for all parts leads to unfeasible constraints for the upper arm and torso (high value) or small and slow avoidance for the hand and forearm (low value). From our observation, $k_2$ should increase from the torso to the hand. In our experiments, $k_2$ is 0.06, 0.06, 0.33, 0.53 for the body parts of the torso, upper arm, forearm, and hand, respectively.

### H. Self-collision avoidance

For self-collision avoidance, we sampled each body link to obtain sets of points on the robot surface. These points serve as both virtual obstacles and projected positions on the robot, which are the closest points of the robot to obstacles. Due to the kinematic limitations of our humanoid robot, the projected positions are only on the hand and forearm of the controlled arm(s), and the virtual obstacles are on the common torso and the "primary arm" (hand, forearm, upper arm) when the dual-arm version of the controller is used. The virtual obstacles are not on the secondary arm because of the arm prioritization—only the "secondary arm" avoids the collision with the "primary arm" and not vice versa.

We find the closest pair of virtual obstacles and projected positions for each combination of body parts. If their distance $d_C$ is less than a threshold, we add the virtual obstacle to the set of obstacles with $a_t \cdot V_C = 1.2 - 20d_C \geq 0$. The coefficients were empirically determined in simulation and on the real robot to obtain appropriate self-collision avoidance reactions. The collision point is the projected position of the obstacle on the robot ($\mathbf{P}_C$), the collision direction ($\mathbf{n}_C$) is the vector from the collision point to the virtual obstacle. Finally, we compute the linear inequality constraint for these obstacles in the same way as for the external obstacles.

### I. Static obstacles

Sometimes, the experiment contains known static objects that the robot should avoid. The almost same procedure as for self-collision avoidance can also be used in this case. Virtual obstacles are sampled from the static object instead of the robot body. The projected positions are points on the robot body parts that are in danger of colliding with the object. This approach was used, for example, to avoid the table during the interactive game demonstration.

### J. Relative position constraint

Other task restrictions can be incorporated into the problem as (in-)equality constraints. For the bimanual task experiment, we added one constraint per each axis to keep a specified rela-

tive position between the hands (end effectors). The constraint can be written in the vector form as

$$\mathbf{x_{k+1}^P} - \mathbf{x_{k+1}^s} = \mathbf{d}_{\text{rel}}$$

where $\mathbf{x_{k+1}^{P/s}} \in \mathbb{R}^3$ are the next (i.e. in timestep $k+1$) positions of the end effectors, and $\mathbf{d}_{\text{rel}} \in \mathbb{R}^3$ is a relative position vector. This equation can be reformulated for our QP problem with joint velocities as

$$
\begin{aligned}
\mathbf{x_k^P} + \begin{bmatrix} \mathbf{J_{t,pos}^P(q^t)} & \mathbf{J_{a,pos}^P(q^P)} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{q}}^{\mathbf{t}} \\ \dot{\mathbf{q}}^{\mathbf{P}} \end{bmatrix} - \\
\mathbf{x_k^s} - \begin{bmatrix} \mathbf{J_{t,pos}^s(q^t)} & \mathbf{J_{a,pos}^s(q^s)} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{q}}^{\mathbf{t}} \\ \dot{\mathbf{q}}^{\mathbf{s}} \end{bmatrix} = \mathbf{d}_{\text{rel}}
\end{aligned}
\tag{14}
$$

where $\mathbf{x_k^{P/s}} \in \mathbb{R}^3$ are the current (i.e. in timestep $k$) positions of the end effectors, $\mathbf{J_{a,pos}^{P/s}(q^{P/s})} \in \mathbb{R}^{3 \times (n-3)}, \mathbf{J_{t,pos}^{P/s}(q^t)} \in \mathbb{R}^{3 \times 3}$ are the translational velocity components of the Jacobian matrices for arm and torso, respectively.

### K. Joint position constraints

The joint position bounds for the QP problem must be set in such a way that the robot joint limits are not exceeded. In our solution, we are inspired by the shaping policy to avoid joint limits in [8] to maximize reachability in the arm workspace. It consists of a flat region in most of the joint range. We replaced the original hyperbolic tangent functions with linear and constant functions to fulfill the constraint condition of the QP problem. Since the problem is solved at the velocity level, the joint position limits are expressed using the joint velocities for the joint $i$ as:

$$\dot{q}_l^i = \dot{q}_{\min}^i \cdot c_{\min}^i, \qquad \dot{q}_u^i = \dot{q}_{\max}^i \cdot c_{\max}^i$$

where $\dot{q}_{\min}^i$ and $\dot{q}_{\max}^i$ are minimum and maximum permissible joint velocity respectively, $c_{\max}^i$ and $c_{\min}^i$ are obtained from the joint limit avoidance policy as:

$$
\begin{aligned}
c_{\min}^i &= \begin{cases} 0 & q^i < g_L^i \\ \frac{q^i - g_L^i}{g_H^i - g_L^i} & g_L^i <= q^i <= g_H^i \\ 1 & g_H^i < q^i \end{cases} \\
c_{\max}^i &= \begin{cases} 0 & G_H^i < q^i \\ \frac{q^i - G_H^i}{G_L^i - G_H^i} & G_L^i <= q^i <= G_H^i \\ 1 & q^i < G_L^i \end{cases}
\end{aligned}
\tag{15}
$$

where $q_i$ is a current joint position, $g_L^i$ and $G_H^i$ are the lowest and highest permissible joint position value respectively, and $g_H^i$ and $G_L^i$ are the thresholds for the joint position constraints.

## IV. EXPERIMENTS AND RESULTS

We prepared seven experiments with a simulated iCub robot and three with a real one to validate the performance of the proposed solution and compare it with other solutions. An overview of the experiments can be seen in Tab. II.

| Exp | Type | Movement | Purpose | Targets | Obstacles |
|-----|------|----------|---------|---------|-----------|
| 1 | Sim | P2P | Reachability | 3x3x3 grid | - |
| 2 | Sim | P2P | Smoothness | 2 poses | - |
| 3 | Sim | Circular | Reference tracking | r = 0.08 m | - |
| 4 | Sim | Circular | Self-Coll. avoidance | r = 0.1 m | - |
| 5-1 | Sim | P2P | Collision avoidance | 2 poses | 1 with vel. [0, -0.05, 0] |
| 5-2 | Sim | P2P | Collision avoidance | 2 poses | 1 with vel. [0.05, 0, 0] |
| 5-3 | Sim | P2P | Collision avoidance | 2 poses | 1 with vel. [0, 0, -0.05] |
| 11 | Real | P2P | Obstacle management | 1 pose | multiple obstacles |
| 12 | Real | P2P | Obstacle management | 2 poses | multiple obstacles |
| 13 | Real | Bimanual | Whole solution | 1 pose | multiple obstacles |

TABLE II: Experiments overview; P2P stands for point-to-point movement.

### A. Experiments in simulation

In simulation, we evaluated the performance of the controller (Sec. III-B) and the trajectory sampling (Sec. III-D). The first set of experiments aims to assess reachability, movement smoothness, and reference tracking, without obstacles nearby. Reachability is tested in a reaching experiment (Exp 1) with the targets taken from a grid of 3x3x3 positions in Cartesian space in random order and randomly switched two orientations. The positions are the Cartesian product of $x = $ [-0.23, -0.19, -0.15] m, $y = $ [0.11, 0.15, 0.19] m, $z = $ [0.08, 0.12, 0.16] m, and the orientations are $o_1 = $ [-0.15, -0.79, 0.59, 3.06], $o_2 = $ [-0.11, 0.99, 0.02, 3.14] in angle-axis notation. Smooth movement is evaluated in a point-to-point experiment (Exp 2), where the robot moves between two alternating poses $p_1 = $ [-0.23, 0.26, 0.02, -0.15, -0.79, 0.59, 3.06] and $p_2 = $ [-0.26, 0.03, 0.03, -0.11, 0.99, 0.01, 3.14], where the first three values are position coordinates in meters and the last four values are the orientation in angle-axis notation. Finally, a circular movement experiment (Exp 3) is chosen to validate the reference tracking. The overview of the experiments can be found in Tab. II.

In addition to the proposed solution, other alternatives and state-of-the-art methods are also evaluated. Specifically, we compare — 1) *HARMONIOUS* – the proposed solution in the single-arm version; 2) *HARMONIOUS2* – the proposed solution in the dual-arm version when the goal of the other arm is to hold a specified position; 3) *NEO* – the controller by Haviland and Corke [2]; 4) *React* by Ngyuen et al. [3]; 5) *Cart* – Cartesian controller for the iCub robot [8], which is evaluated only in Exp 1 and Exp 2 as it primarily serves for point-to-point movements and does not feature avoidance constraints. Controllers 1–4 are evaluated with and without local trajectory sampling (LTS), described in Sec. III-D. *Cart* has its own local trajectory optimization.

The results of the reachability experiment (Exp 1) are shown in Tab. III. In this experiment, our controller has the highest number of targets that are reached in 6D pose (i.e., position and orientation). The target is reached when the end effector is, at the same time, closer than 5 mm and 0.1 rad to the

| LTS | HARMON | HARMON2 | NEO | React | Cart |
|---|---|---|---|---|---|
| Yes | 91.1 % | 90.4 % | 77.8 % | 86.7 % | 90.4 % |
| No | 90.4 % | 89.6 % | 77.8 % | 82.2 % | |

TABLE III: Reachability experiment (Exp 1). Success rate of combined reachability in position and orientation. LTS stands for local trajectory sampling. *Cart* has its own local trajectory optimization. The success rate is computed from 5 repetitions (with a different order of the targets each time) of the experiment (in total 135 targets).

target. The success rate is computed from 5 repetitions (with a different order of the targets each time) of the experiment (in total 135 targets). The impact of the local trajectory sampling is visible as all controllers have better (or at least the same) reachability results with sampling than without sampling.

The results of the movement smoothness experiment (Exp 2) show the main impact of the local trajectory sampling (see Fig. 4a). Position sampling (Fig. 4a left) creates a bell-shaped Cartesian translation velocity profile typical of human arm motion [57]. The velocity profile of *Cart* is the most similar to the exact minimum-jerk movement, as it is specially optimized for it. Other versions are almost identical; only *React* does not reach zero velocity after 5 s. All versions without local sampling have snap onset and slow jerky decay. Orientation sampling (Fig. 4a right) creates a torque-minimal path for rotation as it travels along the straightest path of the rounded surface of a sphere and keeps the rotational velocity constant during movement [59], resulting in a visually smooth transition between two rotations. *HARMONIOUS* and *NEO* have a smooth angular velocity profile; *React* velocity profile is not smooth and does not reach zero after 5 s. *Cart* uses the same sampling for orientation as for position, and thus the resulting angular velocity profile is similar to that for translational velocity.

In the case of circular movement in the reference tracking experiment (Exp 3), there is no effect of local trajectory sampling, as a new target is generated for each period (the tracked reference), meaning that the smoothness of the movement is handled at a higher level (an external target generator). As shown in Fig. 4b, *HARMONIOUS* outperformed others in reference tracking with the lowest position and orientation error.

The next experiment is a circular movement experiment with both arms in opposite directions (Exp 4). It shows how the dual-arm solution works, particularly arm prioritization and self-collision avoidance. As the reference controllers are for single arms only (or separately), only our solution is shown (without trajectory sampling). Figure 4c shows the trajectories of both end effectors making circular movements. In this case, the right arm was selected as the "primary arm". Therefore, the right arm successfully tracks the circle all the time and the left arm prevents self-collision by keeping a safe distance instead of tracking the circle when the arms are too close.

The last set of simulation experiments tests collision avoidance. For simplicity, the obstacle is virtual and is perceived without noise, omitting the influence of sensor data processing on the results. All three experiments are based on Exp 2 and

differ only in the direction of the obstacle: 1) the obstacle moves along the movement, i.e., in the y-axis direction (Exp 5-1); 2) the obstacle moves towards the robot, i.e., in the x-axis direction (Exp 5-2); 3) the obstacle falls in the z-axis direction (Exp 5-3). In this case, *Cart* is not included in the comparison because it cannot handle obstacles. *React* is also omitted because it was outperformed already in the experiments without obstacles. More restrictive repulsive vectors were used instead of linear inequality constraints that allow motions in tangential directions. Therefore, we compare only *HARMONIOUS* and *NEO*. Figure 4d shows the fraction of targets reached during the experiment, the fraction of QP problems successfully solved during the experiments, and the distance between the obstacle and the closest point of the robot, where we assume that a distance of less than 25 mm is too dangerous. There is no significant difference between controllers when the obstacle moves along the robot's movement (Exp 5-1). On the other hand, the obstacle that moves towards the robot (Exp 5-2) is problematic for *NEO*. Half of the time, the targets are not reached within a time limit and the controller cannot solve the QP problem in almost 40 % runs. Furthermore, small distances between the robot and the obstacle sometimes lead to collisions. Unlike *NEO*, *HARMONIOUS* has no problem with collisions or reaching in this experiment (see video S1 in Multimedia Materials). Our solution also works adequately in Exp 5-3. *NEO* did not reach all targets within the time limit, but the distance to an obstacle is safe.

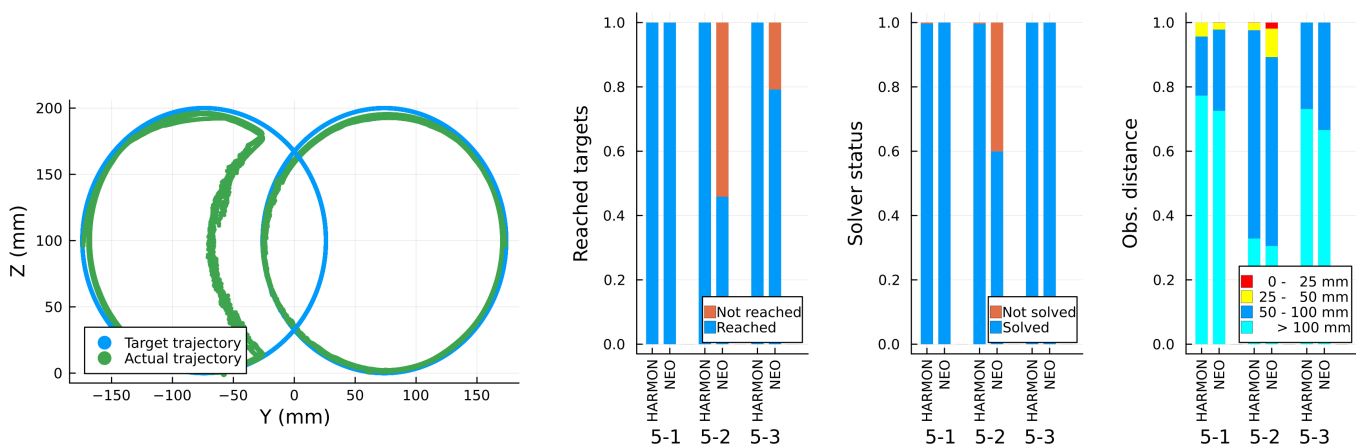### B. Real robot experiments – collision avoidance

The experiments with the real robot were prepared to evaluate the obstacle processing part of the solution, described in Sec. III-F. We show collision avoidance (visual and proximity modalities) and post-collision reaction (tactile modality) and their combination while the robot keeps position (Exp 11, see video S2 in Multimedia Materials) or moves back and forth between two points as in Exp 2 (Exp 12, see video S3 in Multimedia Materials). When visual obstacles are involved, the iCub gaze controller [60] is used to point the robot gaze to an appropriate position in the scene, thus compensating for the robot torso joint movements.

Figure 5 graphically illustrates the experiment using snapshots. Figure 6 shows the operation of the main components of *HARMONIOUS* on plots, focusing on the left arm of the robot (the "primary arm" in these experiments). Note that obstacles at or near the robot hand (visual obstacles near the hand or forearm, proximity at the back of the hand, or touch at the hand or forearm) inevitably compromise the task—staying close to the target—whereas visual obstacles or touch at more proximal body parts give the robot the possibility to exploit its kinematic redundancy and fulfill the task while simultaneously avoiding contacts. Parts of Exp 11—robot tasked with keeping the end effector position while complying with constraints from obstacles spawned by the presence of the human—are shown in Fig. 6a. In Fig. 6a (top), the robot starts in a steady state without obstacles around characterized by a zero target distance (first two seconds, green line). Once a proximity obstacle is detected, the end effector is moving away until the
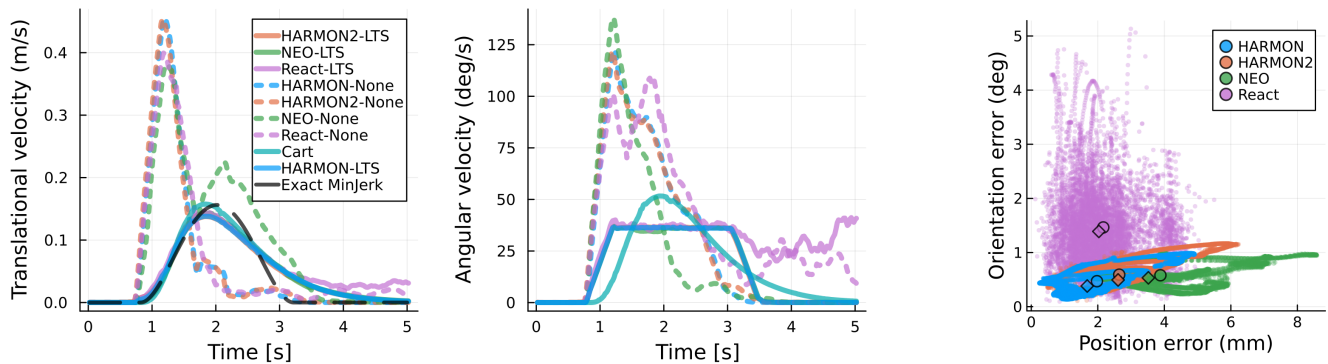
(a) Exp 2: End-effector translational (left) and angular (right) velocity profiles during point-to-point movement and computed exact minimum-jerk translational velocity profile.

(b) Exp 3: Reference tracking error between desired and actual 6D pose. Circles and diamonds represent mean and median pose error, respectively.

(c) Exp 4: Left and right arms movement for our solution (right set as "primary").

(d) Exp 5: Collision avoidance experiments (Exp 5-1, 5-2, 5-3).

Fig. 4: Simulation experiments results. LTS stands for local trajectory sampling.

proximity obstacle constraint disappears from the controller and the target position is reached again. Physical contact of the human with the robot hand (around 5 s) and the forearm (around 7 and 9 s) generate constraints and trigger avoidance. Tactile collision with upper arm (12-13 s) and torso (14-15 s) generate avoidance but do not significantly compromise the task, as the robot kinematic redundancy is exploited by the controller.

In Fig. 6a (center), the first 10 seconds are very similar to the top one. Touch on the forearm causes deviation from the target while contact with the upper arm is evaded while simultaneously staying close to the target. Touch at the hand (5 s) and proximity signal at the hand (7 s) cause deviation from the target. At around 12 s, a proximity obstacle at the hand is evaded, but then avoidance stops as the robot is blocked by contact with the forearm. Once the touch disappears, the proximity collision avoidance continues. Finally, when the proximity obstacle disappears, a tactile stimulus on the hand causes a fast movement towards the target position.

Figure 6a (bottom) shows a part of the experiment where, in addition to proximity and tactile obstacles, visual obstacles are spawned by human keypoint detections in the camera input. At

first, distant visual stimuli for the torso and hand cause only small end-effector deviations from the target. Then, at 5 s, a physical collision with the torso and upper arm occurred, outside the field of view. In contrast, the subsequent tactile input at the hand (8 s) is also detected by vision. Visual stimuli persist once the collision ends, preventing the robot from reaching the target until they disappear. At the end (12-15 s), we can see the Peripersonal space (PPS) projection to all parts of the skin causing deviation from the target.

In Exp 12, the robot is asked to move between two targets every 9 s. Figure 6b show the main system components in action. Figure 6b (top) starts with an undisturbed reaching movement to the target, followed by touch on the torso (4 s) and upper arm (6 s) causing only small deviations from the target. To evade contact with the forearm (7 s), distance to target has to increase. In Fig. 6b (center), the human physically interferes with the forearm during the approach phase, so the arm has to go around the obstacle to reach the target (2-4 s). After the target is reached, a proximity obstacle pushes the hand away from the target again (6-9 s). The effect of visually detected obstacles is shown in Fig. 6b (bottom). At first, visual stimuli block the direct way to the target; thus, the arm goes
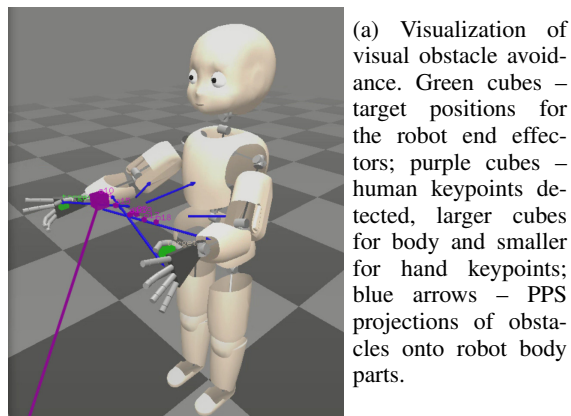
(a) Visualization of visual obstacle avoidance. Green cubes – target positions for the robot end effectors; purple cubes – human keypoints detected, larger cubes for body and smaller for hand keypoints; blue arrows – PPS projections of obstacles onto robot body parts.
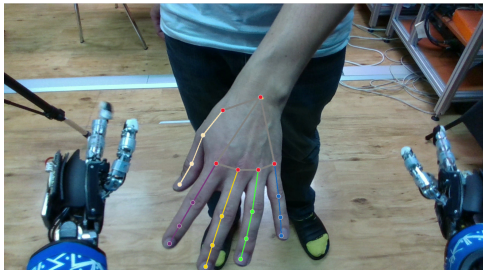


(b) RGB-D camera view of a human approaching the robot with keypoint detections.
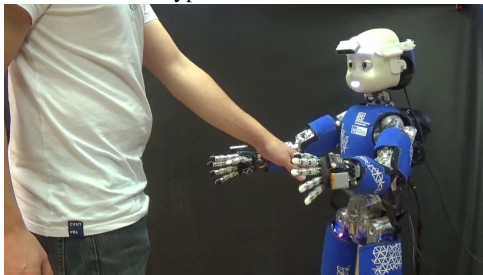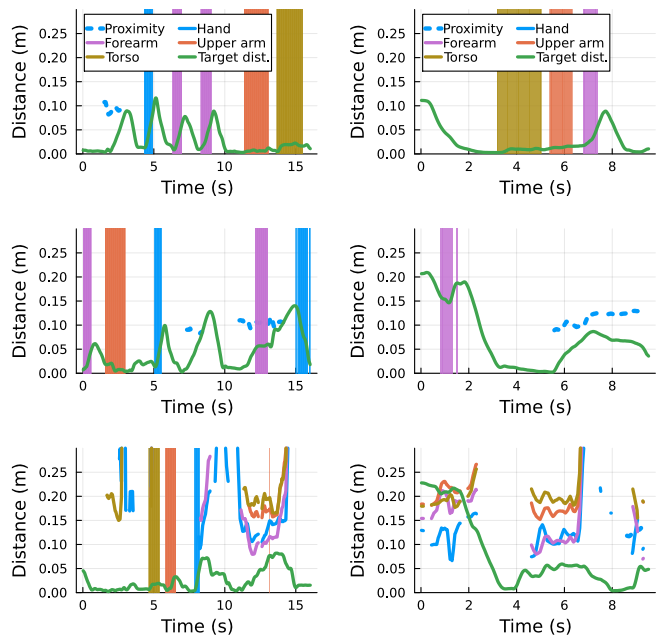


(c) External view – human approaching the robot.

Fig. 5: Visualizations from real-world collision avoidance experiments.



(a) Exp. 11 – Left arm keeping position. Tactile and proximity collision avoidance (top and center), tactile and visual collision avoidance (bottom).

(b) Exp. 12 – Left arm reaching. Tactile collision avoidance (top), proximity and tactile collision avoidance (center), visual collision avoidance (bottom).

Fig. 6: Real-world collision avoidance experiments. Green line – distance of end effector (robot hand) to target; solid lines in other colors – distance of visual obstacles to the closest body part; dotted line — distance to back of robot hand from proximity sensor; vertical lines/stripes — tactile activations –– contacts with corresponding body parts.

around the obstacles to prevent collisions (0-4 s). Once the target is reached, new visual obstacles appear (5-7 s), forcing the end effector deviate from the target again. Note that no particular evasive maneuvers to go around an obstacle to the target are built in. Such behaviors are rather emergent from the local behavior of the controller.

*C. Real robot – bimanual task*

We prepared a bimanual task experiment (Exp 13, see video S4 in Multimedia Materials) in which the robot holds a sponge in its hands (see Fig. 7a). The target is prescribed for the "primary arm" (left) and an additional constraint is introduced to keep the sponge between the two hands. At the same time, dynamic obstacles spawn additional constraints and avoidance. For this experiment, tactile inputs from the robot hands (palms) are ignored (the robot holds the sponge). This experiment aims to verify that the dual-arm version works and that other task constraints, such as the relative distance vector between the end effectors in this case, can be easily incorporated into the
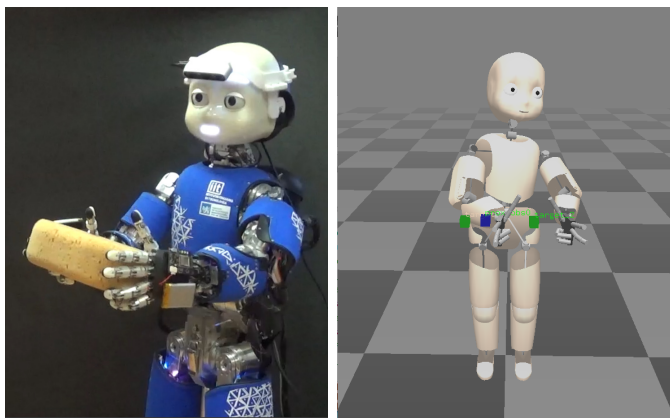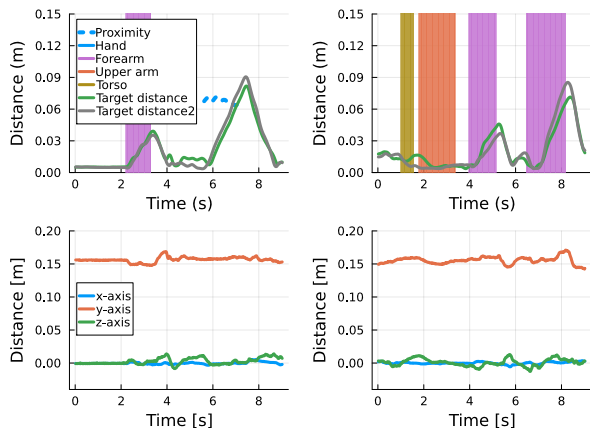
problem definition (the relative position constraint described in Sec. III-J).

Figure 7c shows plots from two sequences of the bimanual task experiment—distances to the target, presence of obstacles for the right and left arms, and the relative position of the end effectors. In the left column, we can see the evasive action of the arms caused by contact with the left forearm and then a proximity obstacle impinging on the left hand. The relative position between both hands remained almost the same as the arms simultaneously moved away from the obstacles, respecting the bimanual constraint. In the right column, we can again see how the high number of degrees of freedom helps to keep the target positions while reacting to contacts on the proximal body parts—torso and upper arm. Contact with the left forearm causes deviation from the target but not losing or squishing the sponge.

Collision avoidance during the bimanual task is more challenging, as it requires a synchronized movement of both arms to keep the specified relative position between them. Therefore, the avoidance movements are smaller compared to previous experiments. The proximity obstacle avoidance is portrayed in Fig. 7b.

choosing the appropriate card, the robot selects which hand to use and initiates the reach movement. At all times and in real time, visual, proximity, and tactile streams are simultaneously processed, spawning obstacles if appropriate and dynamically adding constraints to the robot controller in order to warrant the safety of the human player at all times. The game itself is rather challenging for the human player and hence in order to compete with the robot, he often cheats by pushing the robot physically away or by blocking certain volumes of the workspace by placing his hands over it. This was a deliberate choice to put *HARMONIOUS* at test.

The controller runs in dual-arm mode during the game. Depending on the position of the card in every round, the left or right arm is chosen and set as "primary' (and hence its position reaching target is set as a constraint). Additional constraints were added to the optimization problem by adding static obstacle points (see Sec. III-I) to prevent the robot from colliding with the table. We prepared a high-level Python script to play the game and used the gaze controller [60] to always look at the table. The program loop starts with the processing of rolled dice (numbers sent to the program in the command line) and the current RGB-D image, followed by the detection of cards on the table, the decision of the correct card and computing its 3D position. Then one arm approaches the card position, while the other arm goes to a specific position outside the field of view of the RGB-D camera. After the reach approach (successful or not), the approaching arm returns to a specific position to clear the view for the next round.

The robot with *HARMONIOUS* running successfully managed to run the game and cope with the unstructured interaction, dynamically processing a large number of obstacles perceived through three different sensory modalities, without endangering the human player.

## V. CONCLUSION, DISCUSSION AND FUTURE WORK

We have developed a real-time reactive motion control system for upper body control of a dual-arm robot with a common torso. The proposed solution (*HARMONIOUS*) is designed to enable safe and effective interaction with humans in close proximity. In the absence of obstacles, we systematically tested our controller's reach ability on a grid of position and orientation targets in the workspace and compared it to *NEO* [2] and *React* [3], [4], demonstrating superior performance. We found that assigning different weights to position and orientation tasks is advantageous (unlike in [2]). Preferring arm over torso movements by adjusting weights in the minimization leads to more natural-looking movements. Uniquely to our solution, we commanded both arms and the torso of the iCub humanoid robot (in total 17 DoFs) and demonstrated both the setup where every arm has a separate task and one, set as "primary arm", commands the torso joints, and a bimanual task where the hands hold an object between them. Kinematic singularities are handled through velocity damping and natural kinematic postures are rewarded in the optimization problem formulation. Smooth human-like motions with a bell-shaped Cartesian velocity profile are generated using biologically inspired local trajectory sampling. Overall, several components



(a) Photo from the experiment.

(b) Visualization of hand targets (green cubes) and proximity obstacle (blue cube).

(c) Two sequences from the bimanual task experiment in columns. Obstacle and target distances (top): green and gray lines – distance of end effectors (robot hands) to target; dotted line – distance to back of robot hand from proximity sensor; vertical lines/stripes – tactile activations – contacts with corresponding body parts. Relative position of the end effectors (bottom).

Fig. 7: Real robot – bimanual task.

### D. Real robot – Interactive game demonstration

To demonstrate the operation of *HARMONIOUS* at scale and in a genuine unstructured human-robot interaction scenario, we taught the robot to play a children's board game called Bubbles[1]. Based on the numbers that appear on the dice and their colors, the players—the robot and the human in our case—compete to pick the card on the table where the size of the colored bubbles corresponds to the colors of the dice, ordered by the numbers – see Fig. 1 and video S5 in Multimedia Materials. This game was chosen because there is a shared physical space with unstructured interaction. The robot processes the visual inputs and recognizes the individual cards on the table. After the roll of the dice, the player who first places his hand over the right card wins the round. After

---

[1] A description of the game is available at https://www.piatnik.com/en/games/board-games/children-games/bubbles. For our purposes, we simplified the game, with only six cards, three dice, only the human rollings the dice, and without grasping the cards (only pointing at the selected one). The robot recognizes the cards but is told the numbers on the dice after the roll (as these are too small to be recognized).

act synergistically to produce human-like naturally looking movements.

The solution shows the online dynamic incorporation of obstacles perceived through three different sensory modalities and their on-the-run embedding as whole-body motion constraints into a motion controller with human-like characteristics. Two of these modalities, vision and proximity, sense at a distance, "pre-collision", and the other one—tactile sensors on large areas of the robot body (2000 individual sensors)—is providing contact or post-collision information. The performance of *HARMONIOUS* at scale was finally demonstrated in unstructured physical human-robot interaction while playing a game.

The merit of *HARMONIOUS* lies in the ability to perceive objects in the environment, relate them to the complete surface of the robot body, and, when appropriate, transform them into constraints for the motion controller. Information about obstacles from all three modalities—sometimes overlapping, at other times complementary—was processed and transformed into a unified representation dynamically generating constraints for the whole-body motion controller.

The number or type of sensors used in this work is by no means a requirement for the described system to work. The solution will work with less dense coverage of the perirobot space, covered by only one sensor type, for example. However, dense and multimodal coverage is advantageous for the safety of the interaction. At the same time, additional sensor types could be easily incorporated. Visual obstacles and the corresponding constraints will work for any range-based sensor. Physical contact is implemented here through tactile sensors and acts as a constraint along the contact normal. If a contact force vector was available, it could be incorporated preserving the information about the direction of the constraint (as is currently done for the visual obstacles). One limitation of our current implementation is that the obstacles perceived by one sensory modality are combined into a single motion constraint per robot body part. This was an implementation decision on the obstacle processing layer. Alternatively, multiple constraints from a single modality could be preserved, transformed into constraints, and fed into the motion controller. Or, the information about obstacles from mutliple modalities could be first combined, preserving more information about the context (see contact hypotheses fusion [49]) and only then transformed into motion constraints.

*HARMONIOUS* is a versatile and extendable whole-body motion controller which, however, remains at the kinematic level. The problem could be approached using robot dynamics and formulated at the level of forces and torques. This may be advantageous (solving inverse dynamics rather than inverse kinematics) and ultimately more general, but at the same time it would require accurate identification of the robot inertial parameters. In general, a "dynamics controller" constitutes an alternative, different approach. With *HARMONIOUS* we have shown the potential and versatility of the kinematic approach, which in our view is simpler to deploy in any robot where a kinematic model and position or velocity control are available.

The problem formulation is highly modular and both the minimization criteria (e.g. motivating preferred postures) or constraints to the quadratic program (e.g. obstacles) can be easily removed or different ones added. For example, additional task constraints were added to the problem formulation for the interactive game scenario. On the other hand, our solution guarantees strict task prioritization only between constraints (e.g., obstacle avoidance) and objective function minimization (e.g., target reaching) and not between individual tasks. To enforce strict task priority, a framework of prioritized QP-based kinematic control with a null space projection can be incorporated [61].

Our immediate future work involves adding active gaze control such that the robot can better perceive what is relevant and, at the same time, reduce the noise in the sensory streams using gaze stabilization, for example.

## REFERENCES

[1] J. Rozlivek, P. Svarny, and M. Hoffmann, "Perirobot space representation for HRI: measuring and designing collaborative workspace coverage by diverse sensors," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 5958–5965.

[2] J. Haviland and P. Corke, "NEO: A novel expeditious optimisation algorithm for reactive motion control of manipulators," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1043–1050, Apr. 2021.

[3] P. D. Nguyen, F. Bottarel, U. Pattacini, M. Hoffmann, L. Natale, and G. Metta, "Merging Physical and Social Interaction for Effective Human-Robot Collaboration," in *2018 IEEE-RAS International Conference on Humanoid Robots*, 2018, pp. 710–717.

[4] D. H. P. Nguyen, M. Hoffmann, A. Roncone, U. Pattacini, and G. Metta, "Compact Real-time Avoidance on a Humanoid Robot for Human-robot Interaction," in *ACM/IEEE International Conference on Human-Robot Interaction*, Feb. 2018, pp. 416–424.

[5] H. A. Park, M. A. Ali, and C. S. G. Lee, "Closed-form inverse kinematic position solution for humanoid robots," *International Journal of Humanoid Robotics*, vol. 9, no. 3, p. 1250022, 2012.

[6] P. Beeson and B. Ames, "TRAC-IK: An open-source library for improved solving of generic inverse kinematics," in *2015 IEEE-RAS International Conference on Humanoid Robots*, 2015, pp. 928–935.

[7] S. Lloyd, R. A. Irani, and M. Ahmadi, "Fast and Robust Inverse Kinematics of Serial Robots Using Halley's Method," *IEEE Transactions on Robotics*, vol. 38, no. 5, pp. 2768–2780, Oct. 2022.

[8] U. Pattacini, F. Nori, L. Natale, G. Metta, and G. Sandini, "An experimental evaluation of a novel minimum-jerk Cartesian controller for humanoid robots," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2010, pp. 1668–1674.

[9] K. Glass, R. Colbaugh, D. Lim, and H. Seraji, "Real-Time Collision Avoidance for Redundant Manipulators," *IEEE Transactions on Robotics and Automation*, vol. 11, no. 3, pp. 448–457, 1995.

[10] D. H. Park, H. Hoffmann, P. Pastor, and S. Schaal, "Movement reproduction and obstacle avoidance with dynamic movement primitives and potential fields," in *2008 IEEE-RAS International Conference on Humanoid Robots*, 2008, pp. 91–98.

[11] F. Flacco, T. Kröger, A. De Luca, and O. Khatib, "A depth space approach to human-robot collision avoidance," in *2012 IEEE International Conference on Robotics and Automation*, 2012, pp. 338–345.

[12] C. Escobedo, M. Strong, M. West, A. Aramburu, and A. Roncone, "Contact Anticipation for Physical Human–Robot Interaction with Robotic Manipulators using Onboard Proximity Sensors," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sep. 2021, pp. 7255–7262.

[13] Y. Ding and U. Thomas, "Collision Avoidance with Proximity Servoing for Redundant Serial Robot Manipulators," in *2020 IEEE International Conference on Robotics and Automation*, May 2020, pp. 10 249–10 255.

[14] D. Guo and Y. Zhang, "A New Inequality-Based Obstacle-Avoidance MVN Scheme and Its Application to Redundant Robot Manipulators," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 6, pp. 1326–1340, Nov. 2012.

[15] Y. Tong, J. Liu, X. Zhang, and Z. Ju, "Four-Criterion-Optimization-Based Coordination Motion Control of Dual-Arm Robots," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 15, no. 2, pp. 794–807, 2023.

[16] W. Suleiman, "On inverse kinematics with inequality constraints: New insights into minimum jerk trajectory generation," *Advanced Robotics*, vol. 30, no. 17-18, pp. 1164–1172, Sep. 2016.

[17] D. Rakita, H. Shi, B. Mutlu, and M. Gleicher, "Collisionik: A per-instant pose optimization method for generating robot motions with environment collision avoidance," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 9995–10 001.

[18] O. Khatib, "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots," *Autonomous Robot Vehicles*, pp. 396–404, 1986.

[19] K. Merckaert, B. Convens, C. ju Wu, A. Roncone, M. M. Nicotra, and B. Vanderborght, "Real-time motion control of robotic manipulators for safe human–robot coexistence," *Robotics and Computer-Integrated Manufacturing*, vol. 73, p. 102223, Feb. 2022.

[20] J. Schulman, J. Ho, A. Lee, I. Awwal, H. Bradlow, and P. Abbeel, "Finding Locally Optimal, Collision-Free Trajectories with Sequential Convex Optimization," in *Robotics: Science and Systems IX*, 2013, pp. 1–10.

[21] S. Zimmermann, M. Busenhart, S. Huber, R. Poranne, and S. Coros, "Differentiable Collision Avoidance Using Collision Primitives," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 2022, pp. 8086–8093.

[22] R. Bordalba, T. Schoels, L. Ros, J. M. Porta, and M. Diehl, "Direct collocation methods for trajectory optimization in constrained robotic systems," *IEEE Transactions on Robotics*, vol. 39, no. 1, pp. 183–202, 2023.

[23] D. E. Whitney, "Resolved Motion Rate Control of Manipulators and Human Prostheses," *IEEE Transactions on Man-Machine Systems*, vol. 10, no. 2, pp. 47–53, 1969.

[24] A. Albini, F. Grella, P. Maiolino, and G. Cannata, "Exploiting Distributed Tactile Sensors to Drive a Robot Arm through Obstacles," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4361–4368, Jul. 2021.

[25] A. Cirillo, F. Ficuciello, C. Natale, S. Pirozzi, and L. Villani, "A conformable force/tactile skin for physical human-robot interaction," *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 41–48, Jan. 2016.

[26] E. Magrini and A. De Luca, "Hybrid force/velocity control for physical human-robot collaboration tasks," in *2016 IEEE International Conference on Intelligent Robots and Systems*, Nov. 2016, pp. 857–863.

[27] N. Mansard, O. Stasse, P. Evrard, and A. Kheddar, "A versatile generalized inverted kinematics implementation for collaborative working humanoid robots: The stack of tasks," in *International Conference on Advanced Robotics (ICAR)*, June 2009, p. 119.

[28] C. Escobedo, N. Nechyporenko, S. Kadekodi, and A. Roncone, "A Framework for the Systematic Evaluation of Obstacle Avoidance and Object-Aware Controllers," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2022, pp. 8117–8124.

[29] Y. Nakamura and H. Hanafusa, "Inverse Kinematic Solutions With Singularity Robustness for Robot Manipulator Control," *Journal of Dynamic Systems, Measurement, and Control*, vol. 108, no. 3, pp. 163–171, 09 1986.

[30] A. De Santis, A. Albu-Schaffer, C. Ott, B. Siciliano, and G. Hirzinger, "The skeleton algorithm for self-collision avoidance of a humanoid manipulator," in *2007 IEEE/ASME international conference on advanced intelligent mechatronics*, 2007, pp. 1–6.

[31] A. Dietrich, T. Wimböck, A. Albu-Schäffer, and G. Hirzinger, "Reactive whole-body control: Dynamic mobile manipulation using a large number of actuated degrees of freedom," *IEEE Robotics and Automation Magazine*, vol. 19, no. 2, pp. 20–33, 2012.

[32] S. Haddadin, H. Urbanek, S. Parusel, D. Burschka, J. Roßmann, A. Albu-Schäffer, and G. Hirzinger, "Real-time reactive motion generation based on variable attractor dynamics and shaped velocities," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 3109–3116.

[33] A. Lambert, M. Mukadam, B. Sundaralingam, N. Ratliff, B. Boots, and D. Fox, "Joint Inference of Kinematic and Force Trajectories with Visuo-Tactile Sensing," in *2019 International Conference on Robotics and Automation*, 2019.

[34] K. He, R. Newbury, T. Tran, J. Haviland, B. Burgess-Limerick, D. Kulić, P. Corke, and A. Cosgun, "Visibility Maximization Controller for Robotic Manipulation," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 8479–8486, Jul. 2022.

[35] D. Kulić and E. A. Croft, "Real-time safety for human–robot interaction," *Robotics and Autonomous Systems*, vol. 54, no. 1, pp. 1–12, Jan. 2006.

[36] J. Docekal, J. Rozlivek, J. Matas, and M. Hoffmann, "Human keypoint detection for close proximity human-robot interaction," in *2022 IEEE-RAS 21st International Conference on Humanoid Robots (Humanoids)*, 2022, pp. 450–457.

[37] E. Aljalbout, J. Chen, K. Ritt, M. Ulmer, and S. Haddadin, "Learning Vision-based Reactive Policies for Obstacle Avoidance," in *Proceedings of the 2020 Conference on Robot Learning*. PMLR, 2020, pp. 2040–2054.

[38] S. Haddadin, A. De Luca, and A. Albu-Schäffer, "Robot collisions: A survey on detection, isolation, and identification," *IEEE Transactions on Robotics*, vol. 33, no. 6, pp. 1292 – 1313, 2017.

[39] R. Calandra, S. Ivaldi, M. P. Deisenroth, and J. Peters, "Learning torque control in presence of contacts using tactile sensing from robot skin," in *2015 IEEE-RAS International Conference on Humanoid Robots*, 2015, pp. 690–695.

[40] A. Albini, S. Denei, and G. Cannata, "Enabling natural human-robot physical interaction using a robotic skin feedback and a prioritized tasks robot control architecture," in *2017 IEEE-RAS International Conference on Humanoid Robots*, 2017, pp. 99–106.

[41] J. Kuehn and S. Haddadin, "An Artificial Robot Nervous System to Teach Robots How to Feel Pain and Reflexively React to Potentially Damaging Contacts," *IEEE Robotics and Automation Letters*, vol. 2, no. 1, pp. 72–79, Jan. 2017.

[42] P. Svarny, J. Rozlivek, L. Rustler, M. Sramek, Özgür Deli, M. Zillich, and M. Hoffmann, "Effect of active and passive protective soft skins on collision forces in human–robot collaboration," *Robotics and Computer-Integrated Manufacturing*, vol. 78, p. 102363, 2022.

[43] F. Flacco and A. De Luca, "Real-time computation of distance to dynamic obstacles with multiple depth sensors," *IEEE Robotics and Automation Letters*, vol. 2, no. 1, pp. 56–63, 2016.

[44] S. Mühlbacher-Karrer, M. Brandstötter, D. Schett, and H. Zangl, "Contactless Control of a Kinematically Redundant Serial Manipulator Using Tomographic Sensors," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 562–569, Apr. 2017.

[45] E. Dean-Leon, F. Bergner, K. Ramirez-Amaro, and G. Cheng, "From multi-modal tactile signals to a compliant control," in *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, 2016, pp. 892–898.

[46] A. E. H. Martin, E. Dean-Leon, and G. Cheng, "Tacto-selector: Enhanced hierarchical fusion of pbvs with reactive skin control for physical human-robot interaction," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 9121–9127.

[47] S. Armleder, E. Dean-Leon, F. Bergner, and G. Cheng, "Interactive force control based on multimodal robot skin for physical human-robot collaboration," *Advanced Intelligent Systems*, vol. 4, no. 2, p. 2100047, 2022. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/aisy.202100047

[48] S. Wieland, D. Gonzalez-Aguirre, N. Vahrenkamp, T. Asfour, and R. Dillmann, "Combining force and visual feedback for physical interaction tasks in humanoid robots," in *2009 9th IEEE-RAS International Conference on Humanoid Robots*, 2009, pp. 439–446.

[49] J. Felip, A. Morales, and T. Asfour, "Multi-sensor and prediction fusion for contact detection and localization," in *2014 IEEE-RAS International Conference on Humanoid Robots*, 2014, pp. 601–607.

[50] J. Cléry and S. B. Hamed, "Frontier of self and impact prediction," *Frontiers in psychology*, vol. 9, p. 1073, 2018.

[51] A. Roncone, M. Hoffmann, U. Pattacini, L. Fadiga, and G. Metta, "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, pp. 1–32, 10 2016.

[52] A. Roncone, M. Hoffmann, U. Pattacini, and G. Metta, "Learning peripersonal space representation through artificial skin for avoidance and reaching with whole body surface," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 3366–3373.

[53] G. Metta, L. Natale, F. Nori, G. Sandini, D. Vernon, L. Fadiga, C. von Hofsten, K. Rosander, M. Lopes, J. Santos-Victor, A. Bernardino, and L. Montesano, "The iCub humanoid robot: An open-systems platform for research in cognitive development," *Neural Networks*, vol. 23, no. 8, pp. 1125–1134, 2010.

[54] P. Maiolino, M. Maggiali, G. Cannata, G. Metta, and L. Natale, "A flexible and robust large scale capacitive tactile system for robots," *IEEE Sensors Journal*, vol. 13, no. 10, pp. 3910–3917, 2013.

[55] K. Watanabe, M. Strong, M. West, C. Escobedo, A. Aramburu, K. C. Kodur, and A. Roncone, "Self-contained kinematic calibration of a novel whole-body artificial skin for human-robot collaboration," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 1778–1785.

[56] T. Yoshikawa, "Manipulability of robotic mechanisms," *The International Journal of Robotics Research*, vol. 4, no. 2, pp. 3–9, 1985.

[57] T. Flash and N. Hogan, "The coordination of arm movements: an experimentally confirmed mathematical model," *Journal of Neuroscience*, vol. 5, no. 7, pp. 1688–1703, 1985.

[58] R. Shadmehr and S. Wise, "Supplementary documents for "computational neurobiology of reaching and pointing"," https://storage.googleapis.com/wzukusers/user-31382847/documents/5a7253343814f4Iv6Hnt/minimumjerk.pdf, 2005, accessed: October 13, 2023.

[59] K. Shoemake, "Animating rotation with quaternion curves," *SIGGRAPH Comput. Graph.*, vol. 19, no. 3, p. 245–254, jul 1985.

[60] A. Roncone, U. Pattacini, G. Metta, and L. Natale, "A Cartesian 6-DoF Gaze Controller for Humanoid Robots," in *Proceedings of Robotics: Science and Systems*, AnnArbor, Michigan, June 2016.

[61] O. Kanoun, F. Lamiraux, P.-B. Wieber, F. Kanehiro, E. Yoshida, and J.-P. Laumond, "Prioritizing linear equality and inequality systems: Application to local motion planning for redundant robots," in *2009 IEEE International Conference on Robotics and Automation*, 2009, pp. 2939–2944.

**Ugo Pattacini** (Member, IEEE) is a Technologist at the Istituto Italiano di Tecnologia (IIT) of Genoa, in Italy. He graduated with honors in Electronic Engineering at the University of Pisa (2001) and he subsequently gathered 7 years of working experience in hi-tech companies in the field of Formula 1 applications (Magneti Marelli Racing Dep. and Toyota F1 Team) and aerospace (Thales Alenia Space). Ugo Pattacini received his Ph.D. in Robotics, Neurosciences, and Nanotechnologies from IIT in 2011 and he is currently involved in the development of the humanoids iCub, ergoCub, and R1, focusing on the advancement of robot motor capabilities. Ugo has also led the IIT research activities in collaboration with Fondazione Don Gnocchi ONLUS, which aims to exploit R1 as an assistive robot for rehabilitation tasks in domestic and hospital settings. Additionally, since 2023 he has been supervising the IIT team responsible for the activities of the Joint Lab with Camozzi Automation to develop smart grippers equipped with AI-enabled capabilities for autonomous and adaptive grasping. Ugo participated in the EU-funded projects RobotCub, CHRIS, EFAA, WYSIWYD, TACMAN, and ETAPAS. He is a member of IEEE, RAS, and CSS.

**Jakub Rozlivek** is a PhD student at the Humanoid and Cognitive Robotics Laboratory (Faculty of Electrical Engineering, CTU in Prague) under Matej Hoffmann. He received his BSc. and MSc. degrees in Cybernetics and Robotics from the Faculty of Electrical Engineering, CTU in Prague in 2019 and 2022, respectively. His research interests include robot kinematic calibration, physical and social human robot interaction, robot kinematic control, and active perception.

**Matej Hoffmann** (Senior Member, IEEE) received the Ph.D. degree in Informatics from Artificial Intelligence Lab, University of Zurich, Zurich, Switzerland, in 2012. From 2013 to 2016, he conducted postdoctoral research with the iCub Facility of the Italian Institute of Technology, Genoa, Italy, supported by a Marie Curie Intra-European Fellowship. In 2017, he joined the Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague, where he is currently an Associate Professor and the Coordinator of the Humanoid and Cognitive Robotics Group. His research interests include humanoid, cognitive developmental, and collaborative robotics, as well as active perception for robot manipulation and grasping.

**Alessandro Roncone** (Senior Member, IEEE) is Assistant Professor in the Computer Science Department at University of Colorado Boulder. He received his B.Sc. *summa cum laude* in Biomedical Engineering in 2008, and his M.Sc. *summa cum laude* in NeuroEngineering in 2011 from the University of Genoa, Italy. In 2015 he completed his Ph.D. in Robotics, Cognition and Interaction Technologies from the Italian Institute of Technology [IIT], working on the iCub humanoid in the Robotics, Brain and Cognitive Sciences department and the iCub Facility. From 2015 to 2018, he was Postdoctoral Associate at the Social Robotics Lab in Yale University, performing research in Human-Robot Collaboration. He joined as faculty at CU Boulder in August 2018, where he is Director of the Human Interaction and Robotics Group (HIRO) and Associate Director of the MS/PhD Program in Robotics. His work is at the intersection of Robotics, Artificial Intelligence, and Human-Centered Computing to design robots that help and augment human capabilities.