Visual and Haptic Cues for Human-Robot Handover*

Marco Costanzo[1], Ciro Natale[1] and Mario Selvaggio[2]

Abstract— **The adoption of robots outside their cages in conventional industrial scenarios requires not only safe human-robot interaction but also intuitive human-robot interactive communication. In human-robot collaborative tasks, the objective is to help humans in performing their job with less physical and cognitive effort. A collaborative task can involve the exchange of objects between the robot and the operator. However, the handover operation should be sufficiently intuitive, fluid, and natural for being accepted by the involved humans. Naturalness strongly depends on the speed of the object exchange and the way of communication. For the latter aspect, this paper proposes a multi-modal communication based on visual and haptic cues. Concerning the handover speed requirement, the paper proposes a high-performance visual servoing based on an Extended Kalman Filter (EKF) estimating object speed during the handover and a homography-based object tracking. The object safety is ensured by proper control of the robot grasp force based on a model-based approach exploiting tactile measurements. The same perception modality is also used as a source of haptic cues that make the handover intuitive and natural. Experiments of human-robot handovers through haptic and visual cues communication demonstrate the effectiveness of the proposed approach.**

I. INTRODUCTION

Even if robotic solutions that automate industrial and logistic processes are widespread, many tasks cannot be executed by robots and require a human partner. As an example, in a supermarket scenario, opening cartons and picking single items out of a box before placing them on the shop shelf are complex operations that demand both cognitive and manipulation skills yet outside the reach of robots. Therefore, such tasks should be performed collaboratively and object exchange is the primary collaboration modality. A robot that receives an object from the human partner (H2R handover) can lower the human workload by reaching placing poses outside of the human workspace or the ergonomic golden zone. At the same time, the dual operation (R2H handover) can be useful both in the logistic and in-house scenarios, where robots could help elderly people to retrieve objects from high or uncomfortable places.

From a formal point of view, the handover is an action between two agents, the *giver* and the *receiver* [1]. It is usually divided into two phases, the pre-handover, and the physical handover. For a recent survey on handover tasks the reader is referred to [1].

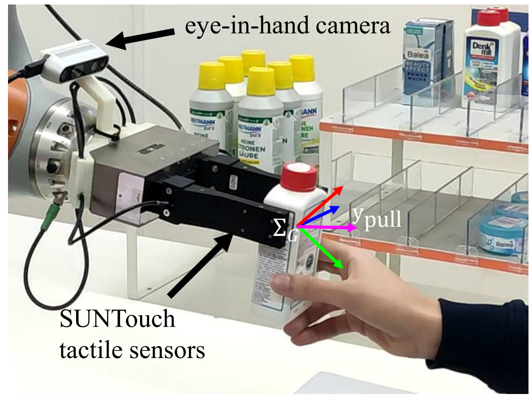

Fig. 1. Experimental setup for the H2R and R2H handover, the frame Σ_G (in RGB convention) is the grasp frame, the magenta arrow represents the pulling direction y_{pull} (defined in Section III).

Both before and during the operation, the communication mechanism is crucial [2]; we adopt both haptic and visual cues as communication tools between the two agents, assuming that the handling device (the gripper) is equipped with force/tactile sensors and an eye-in-hand camera (see the experimental setup in Fig. 1). The haptic cues are based on the interaction force perceived by the agents through tactile sensing at the fingertips while the visual cues are based on hand gestures and visual servoing algorithms as in our former paper [3]. However, here the dynamic performance of the visual servoing controller has been improved in two aspects with respect to the algorithm used in [3]. The dynamic performance has been enhanced through the explicit estimation of the object velocity via an EKF and owing to a novel homography-based tracker. Moreover, an hand gesture detector is adopted to make the human-robot communication more intuitive and interactive compared to [3].

During the H2R operation, the robot visual loop is used to track the target object pose and the handover location, which is chosen by the giver and communicated to the robot with a hand gesture. During the physical handover phase, the object weight is shared between the giver and the receiver. The grasping force of both agents is modulated during the object exchange [4]. In this paper, this phase is built on top of the slipping avoidance algorithm originally proposed by [5]. However, the slipping avoidance strategy alone is not effective enough to achieve a proper handover. The communication between the giver and the receiver is essential and the robot uses haptic cues to communicate its readiness to take the object over.

*This work was partially supported by the European Commission under the Horizon Europe research grant INTELLIMAN, project ID: 101070136, and by the European Union's Horizon 2020 research and innovation programme under grant agreement No 101017008.

[1]Marco Costanzo and Ciro Natale are with the Dipartimento di Ingegneria, Università degli Studi della Campania Luigi Vanvitelli, 81031 Aversa (CE), Italy marco.costanzo@unicampania.it

[2]Mario Selvaggio is with the Department of Electrical Engineering and Information Technology, University of Naples Federico II, 80138 Naples, Italy mario.selvaggio@unina.it

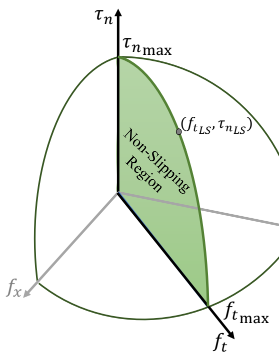

Fig. 2. Limit Surface in the 3D wrench space. $f_{t\max}$: maximum tangential dry friction force; $\tau_{n\max}$: maximum torsional dry friction torque; $(f_{t\mathrm{LS}})$, $(\tau_{n\mathrm{LS}})$: the generic point on the LS, i.e., the maximum external load that the friction can withstand given the grasping force f_n.

In the R2H handover, grasping force modulation is of paramount importance. Releasing the object as soon as a pull is detected might cause a fall if the receiver is not correctly sharing the load. As in our former paper [3], we explicitly detect the load-sharing phase via a Finite State Machine (FMS), now enriched with a new phase managing hand gestures.

II. CONTROLLERS

A. Grasp force controller

The grasp control algorithm aims at modulating the grasping force to automatically avoid slippage. The object is modeled as a *planar slider* and the robot exchanges friction force and torque with the object by means of the fingertips of the SUNTouch force/tactile sensor [6] able to measure the 6D contact wrench. Under the assumption of axisymmetric pressure distribution [7], the contact area is a circle of radius $\rho = \delta f_n^{\gamma}$, where f_n is the grasping force and the two parameters δ and γ can be estimated via the procedure described in [5]. The static friction is modeled with the well-known Limit Surface (LS) concept [8]. The LS is defined in the wrench space (Fig. 2), if the point describing the external load (f_t, τ_n) belongs to the area inside the LS, no slippage occurs. The bigger the grasp force, the bigger the LS and the non-slipping region. By following the same arguments of [9], it is possible to compute the minimum grasping force that keeps the external load inside the LS $f_{n\mathrm{LS}}$.

However, it is well known that such an analysis takes into account constant loads only and, in the case of time-varying forces, the break-away force decreases as soon as the load variation rate increases [10]. This effect can be captured by resorting to the LuGre dynamic friction model [11] applied to the instantaneous rotation about the Center of Rotation (CoR) of the planar slider [12], via the following nonlinear observer

$$\dot{\zeta} = \omega - \frac{\sigma_0}{g(f_n, c)} \zeta \, |\omega| \tag{1}$$

$$\dot{\omega} = l(-\sigma_0 \zeta - \sigma_1(f_n, c)\omega + y), \tag{2}$$

where ζ is the internal dry friction state corresponding to the dry friction $\sigma_0\zeta$. The functions $g(f_n, c)$ and $\sigma_1(f_n, c)\omega$ are the maximum dry friction and viscous friction, respectively, computed as a torque about the CoR depending on the estimation of the CoR position c and the normal force f_n as detailed in [12]. The torque $y = \tau_n - c f_t$ is the generalized friction torque measured at the fingertip. When $|y| > g(f_n, c)$ the dry friction is not enough to counteract the sliding motion and the slipping velocity ω builds up. To counteract both the external load and the break-away force decrease, the estimated slipping velocity is controlled to zero by applying a grasping force computed as the superposition of two contributions

$$f_n = f_{n\mathrm{LS}} + f_{nd}, \tag{3}$$

where $f_{n\mathrm{LS}}$ and f_{nd} are called static and dynamic contributions respectively. In particular, f_{nd} regulates the estimated slipping velocity ω to zero by means of a linear control action. During certain phases of the handover task, the dynamic contribution is turned off (i.e., $f_{nd} = 0$). Thus two control modalities are defined: *static mode* (dynamic contribution disabled), *dynamic mode* (dynamic contribution activated). The strategy to switch between the two modalities during the handover is described in Section III.

B. Visual Servoing

During the H2R handover, the robot picks the object directly from the human hand. The object is presented in an unknown pose in the camera's field of view. Moreover, the object location cannot be considered constant as the human giver could (in)voluntarily move the object during the maneuver. This issue is addressed by resorting to a visual servoing (VS) controller. In order to enhance the VS tracking performance, the object velocity is estimated via an Extended Kalman Filter (EKF) and used as a disturbance cancellation action (see Fig. 3) on the camera velocity.

The objective of the VS controller is to align the current camera image with a target one by controlling the camera velocity. We assume to have an RGB-D image, thus the strategy is to synthetically represent the images with 3D feature points composed by the coordinates of some keypoints on the object surface in the camera frame.

Let $\boldsymbol{s} = \left[\boldsymbol{s}_1^\top \dots \boldsymbol{s}_N^\top\right]^\top$, $\boldsymbol{s}^\star = \left[\boldsymbol{s}_1^{\star\top} \dots \boldsymbol{s}_N^{\star\top}\right]^\top \in \mathbb{R}^{3N}$ be the vectors of the N current and matched 3D features $\boldsymbol{s}_i$, $\boldsymbol{s}_i^\star \in \mathbb{R}^3$, the relation between the variation of $\boldsymbol{s}$ and the camera body velocity screw $\boldsymbol{v} = \left[\mathbf{v}^\top \, \boldsymbol{\omega}^\top\right]^\top$ is

$$\dot{\boldsymbol{s}} = \boldsymbol{L}(\boldsymbol{s})\boldsymbol{v}, \tag{4}$$

where $\boldsymbol{L}(\boldsymbol{s})$ is the so-called *interaction matrix* [13] which, considering 3D feature points, can be written as

$$\boldsymbol{L}(\boldsymbol{s}) = \begin{bmatrix} -\boldsymbol{I}_3 & \dots & -\boldsymbol{I}_3 \\ \boldsymbol{S}^\top(\boldsymbol{s}_1) & \dots & \boldsymbol{S}^\top(\boldsymbol{s}_N) \end{bmatrix}^\top, \tag{5}$$

where $\boldsymbol{I}_3$ is the 3×3 identity matrix and $\boldsymbol{S}(\cdot)$ is the skew symmetric operator.

Taking into account the discrete-time nature of the control algorithm, in the following, we will derive the exact sampled data version of the system (4). First, considering the structure of the matrix $\boldsymbol{L}(\boldsymbol{s})$, (4) can be written in a form that is linear with respect to the state, i.e.,

$$\dot{\boldsymbol{s}} = \bar{\boldsymbol{S}}^{\top}(\boldsymbol{\omega})\boldsymbol{s} - \bar{\mathbf{v}}, \tag{6}$$

where $\bar{\boldsymbol{S}}(\boldsymbol{\omega}) = \boldsymbol{I}_N \otimes \boldsymbol{S}(\boldsymbol{\omega})$, $\bar{\mathbf{v}} = \mathbf{1}_N \otimes \mathbf{v}$, $\otimes$ represent the Kronecker product, $\boldsymbol{I}_N$ is the $N \times N$ identity matrix and $\mathbf{1}_N = [1\,1\ldots 1]^{\top} \in \mathbb{R}^N$. Let T be the control sampling time, in the time interval $[kT,\,(k+1)T]$ the control input is kept constant at the value $\boldsymbol{v}_k = \left[\mathbf{v}_k^{\top}\,\boldsymbol{\omega}_k^{\top}\right]^{\top}$ and the system behaves as a linear one inside the sampling time intervals. Thus, it is possible to write the discrete-time feature dynamics in the sampling instants as

$$\boldsymbol{s}_{k+1} = e^{-\bar{\boldsymbol{S}}(\boldsymbol{\omega}_k)T}\boldsymbol{s}_k - \int_0^T e^{-\bar{\boldsymbol{S}}(\boldsymbol{\omega}_k)\sigma}\,\mathrm{d}\sigma\bar{\mathbf{v}}_k. \tag{7}$$

Finally, by expanding the exponential matrices in power series, it is possible to write

$$\boldsymbol{s}_{k+1} = \boldsymbol{s}_k + \boldsymbol{P}(\boldsymbol{\omega}_k)\boldsymbol{L}(\boldsymbol{s}_k)\boldsymbol{v}_k, \tag{8}$$

with the matrix function $\boldsymbol{P}(\cdot) : \mathbb{R}^3 \;\to \mathbb{R}^{3N\times 3N}$ defined as

$$\boldsymbol{P}(\boldsymbol{\omega}) = \int_0^T e^{-\bar{\boldsymbol{S}}(\boldsymbol{\omega})\sigma}\,\mathrm{d}\sigma. \tag{9}$$

The velocity $\boldsymbol{v}$ in equation (4) represents the relative camera/object velocity that, in ideal conditions with a static object, coincides with the camera velocity. In the general case, the object motion can be modeled as an additional disturbance velocity $\boldsymbol{v}_d = [\mathbf{v}_d^{\top}\,\boldsymbol{\omega}_d^{\top}]^{\top}$ with respect to the camera frame, namely

$$\boldsymbol{v} = \boldsymbol{v}_c + \boldsymbol{v}_d, \tag{10}$$

where $\boldsymbol{v}_c = [\mathbf{v}_c^{\top}\,\boldsymbol{\omega}_c^{\top}]^{\top}$ is the actual camera velocity command. Note that, $\boldsymbol{v}_d$ can be used to model both the object motion and any inaccuracy of the robot in generating the velocity $\boldsymbol{v}_c$. However, since (9) assumes that the continuous time velocity is a piece-wise constant signal, we are implicitly assuming that also the disturbance $\boldsymbol{v}_d$ is piece-wise constant. This is true if object acceleration is sufficiently low compared to the adopted sampling time T.

The estimation of the disturbance velocity $\boldsymbol{v}_d$ is carried out by means of an EKF assuming a constant velocity model, i.e.,

$$\boldsymbol{v}_{d\,k+1} = \boldsymbol{v}_{d\,k} + \boldsymbol{\nu}_k \tag{11}$$

$$\begin{aligned}\boldsymbol{s}_{k+1} &= \boldsymbol{s}_k + \boldsymbol{P}(\boldsymbol{\omega}_{c\,k} + \boldsymbol{\omega}_{d\,k})\boldsymbol{L}(\boldsymbol{s}_k)(\boldsymbol{v}_{c\,k} + \boldsymbol{v}_{d\,k}) + \boldsymbol{\eta}_k \\ &= \boldsymbol{f}(\boldsymbol{s}_k, \boldsymbol{v}_{d\,k}) + \boldsymbol{\eta}_k \end{aligned} \tag{12}$$

$$\boldsymbol{y}_k = \boldsymbol{s}_k + \boldsymbol{\chi}_k, \tag{13}$$

where $\boldsymbol{y}_k$ is the measurable output, and $\boldsymbol{\nu}_k$, $\boldsymbol{\eta}_k$, $\boldsymbol{\chi}_k$ are Gaussian process noise with covariance matrices $\boldsymbol{N}$, $\boldsymbol{H}$ and $\boldsymbol{X}$, respectively.

The visual servoing control scheme is shown in Fig. 3. The velocity control output is the sum of two terms

$$\boldsymbol{v}_c = -\hat{\boldsymbol{v}}_d + \boldsymbol{v}_{\mathrm{cl}}, \tag{14}$$

where $\hat{\boldsymbol{v}}_d$ is the disturbance velocity estimated by the EKF with the purpose to counteract the disturbance velocity $\boldsymbol{v}_d$,

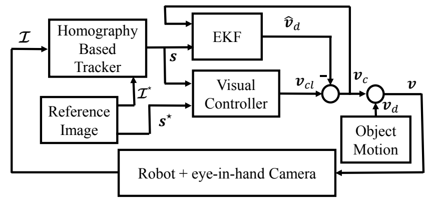

Fig. 3. Visual control system block scheme.

while $\boldsymbol{v}_{\mathrm{cl}}$ is the closed-loop control action designed to ensure stability. Such a component is computed as in [13] as

$$\boldsymbol{v}_{\mathrm{cl}} = -\lambda \boldsymbol{L}^{\dagger}(\boldsymbol{s}_k)\boldsymbol{e}_k, \tag{15}$$

where $\boldsymbol{e}_k = \boldsymbol{s}_k - \boldsymbol{s}^{\star}$ is the feature error and $\lambda > 0$ is the control gain.

The 3D feature points $\boldsymbol{s}_i$ can be measured by tracking 2D features in pixel coordinates on the object surface and then, by means of the depth sensor and the camera intrinsic parameters, transform the 2D features in the 3D metric space [14]. Assuming that the object has a planar textured face to be tracked, let $[u_i\;v_i]^{\top}$ and $[u_i^{\star}\;v_i^{\star}]^{\top}$ be the pixel coordinates corresponding to the 3D features $\boldsymbol{s}_i$ and $\boldsymbol{s}_i^{\star}$, respectively. If we choose the 3D features on a planar face of the object, the relation between all the current and reference features is given by the homography matrix $\boldsymbol{\mathcal{H}} \in \mathbb{R}^{3\times 3}$ [15], namely,

$$\mu_{i\,k}\begin{bmatrix} u_{i\,k} \\ v_{i\,k} \\ 1 \end{bmatrix} = \boldsymbol{\mathcal{H}}_k \begin{bmatrix} u_{i\,k}^{\star} \\ v_{i\,k}^{\star} \\ 1 \end{bmatrix} \quad \forall i, \tag{16}$$

where $\mu_{i\,k}$ is an auxiliary variable. Thus, instead of tracking the 2D features, we estimate the homography matrix via a template-tracking algorithm based on ZNCC [16]. The target 2D features $[u_i^{\star}\;v_i^{\star}]^{\top}$ are selected arbitrarily on the target image and the current ones are generated via the homography matrix (16). This approach has an advantage compared to local trackers, such as KTL [17], which track 2D features individually and errors can accumulate over time or the tracked features could move independently even if they belong to the same 3D rigid body. A template-tracking algorithm, instead, does not accumulate errors because it always compares the current image with the target one. Moreover, reconstructing the 2D features from the homography estimation ensures that the features always respect the rigid body constraint.

III. HANDOVER

We propose a framework that uses both visual and haptic cues for efficient H2R and R2H handover under the following assumptions: the robot receiver knows the object to be used as the target for the visual servoing controller; the object is texture-rich with at least one planar surface to be used for the homography estimation; the robot is endowed with a sensorized hand able to estimate the contact wrench and an eye-in-hand RGB-D camera (see Fig. 1).

To recognize hand gestures, we use the ROS Hand Gesture Recognition package [18], which relies on the MediaPipe Gesture Recognizer, an open-source machine learning-based

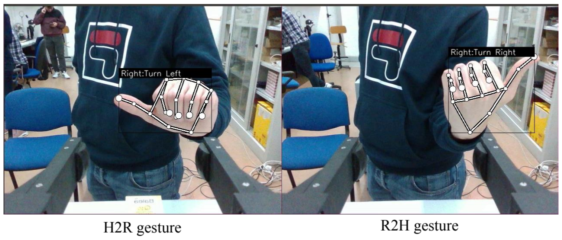

Fig. 4. Recognised gestures used for the H2R (left) and R2H (right) visual cues.

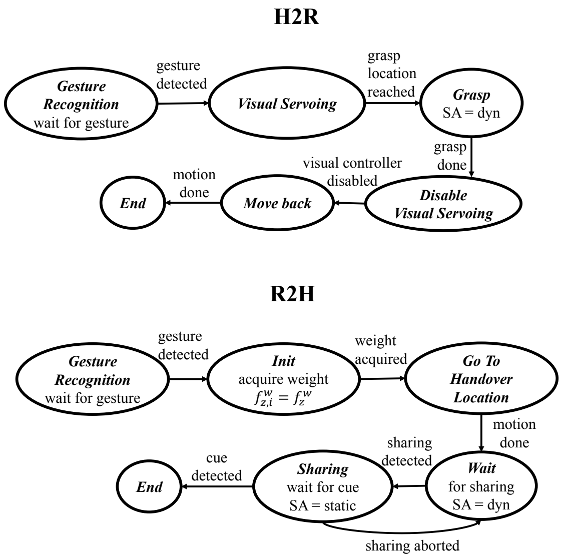

Fig. 5. Representation of the FSMs for the H2R (top) and R2H (bottom) handover algorithms.

package from Google, that provides the recognized hand gesture results in real time along with 21 hand landmarks coordinates of the detected hand into standard ROS topics. The package can classify hand signs starting from image data streamed from an RGB-D camera making use of a pre-trained machine learning model. We used the two hand signs *turn left* and *turn right* showed in Fig. 4 to activate the H2R and R2H phases, respectively.

A. H2R Handover

During the H2R handover, the human operator presents an object to the robot that has to grasp it directly from the human hand. The algorithm is presented in the FSM diagram in Fig. 5-top. Since, at the beginning, there is no contact between the manipulator and the object, the robot relies on visual cues only.

In the *Gesture Recognition* phase, the robot vision algorithm awaits for the visual H2R hand gesture shown in Fig. 4 to initiate the handover action.

As soon as the robot recognizes the gesture, the visual servoing algorithm described in Sec. II-B is activated. Thanks to the disturbance velocity estimated by the EKF, the giver can move the object to a different location while the visual controller tracks the object without compromising the success of this phase.

The grasp location is considered reached when the norm of the visual servoing error $\boldsymbol{e}$ goes below a desired accuracy $\varepsilon_{\mathrm{vs}}$. Then, the gripper closes the fingers and the dynamic slipping avoidance algorithm is activated. At this time the object is considered secured and the visual servoing is deactivated. Note that, the visual servoing is still active after reaching the grasp location until this phase, this is because the giver could still move the object.

When the object is secured, the robot moves the end effector back, away from the reached handover location (*Move Back* state). This is unconsciously perceived by the human giver as a haptic cue, i.e., an increased tangential force in the robot's pulling direction. This means that the robot has handled the object and the human can securely release it.

B. R2H Handover

The R2H handover strategy is described in the FSM diagram in Fig. 5-bottom.

The robot starts holding the object in *dynamic slipping avoidance* mode. Once again, the robot vision algorithm awaits for the visual R2H hand gesture shown in Fig. 4.

As soon as the human intention to initiate the handover is confirmed, the robot measures the object's weight by means of the sensorized fingertips. The weight $f^w_{z,i}$ can be estimated as the force component f^w_z along the z-axis of the world frame (assumed as opposed to gravity) measured at the beginning of this phase when the robot is still and the human does not exchange forces with it. Once the weight is acquired the robot reaches the handover location acquired by the visual gesture cue and it goes into the *Wait* state. In this state, the robot is still in dynamic slipping avoidance mode to counteract any disturbance applied to the object. In the very first phase of the actual handover, the receiver grasps the object by partially holding the load, this haptic cue is caught by the robot that enters in the *Sharing* state. This is possible by comparing the actual force component f^w_z with the weight measured beforehand, i.e.,

$$f^w_z > \nu_s f^w_{z,i}, \quad 0 < \nu_s < 1, \tag{17}$$

where the factor ν_s defines the percentage of the load that the receiver has to hold before the robot enters the *Sharing* state. By checking such a condition, the robot establishes the receiver's intention to share the object's weight to successively grab it. Note that, in the *Sharing* state the slipping avoidance is switched to the *static* modality. This is because the receiver is actually helping in holding the object and any slight trembling of the human receiver would cause unnecessary reactions of the dynamic controller.

The sharing alone does not ensure a successful handover as the receiver could decide to abort the operation. For this reason, there is a loop on the FSM diagram (Fig. 5-bottom). The sharing abortion is detected with the haptic cue

$$f^w_z < \nu_w f^w_{z,i}, \tag{18}$$

with $1 > \nu_w > \nu_s > 0$. To avoid useless switches between the *Wait* and *Sharing* state, the scale factors ν_w and ν_s should be selected such that $(\nu_w - \nu_s)\left|f_{z,i}^w\right|$ is greater than the measurement noise.

In the sharing phase, the receiver could complete the handover by pulling the object. Once again, this is detected with a haptic cue, i.e.,

$$f_z^w > \phi_z \ \vee \ f_{\text{pull}} > \phi_p, \quad \phi_z, \phi_p > 0, \tag{19}$$

where f_{pull} is the force component in the pulling direction, defined as the projection of the gripper approach axis on the xy-plane of the world frame. The cue detected by the condition (19) considers the case of the handover partner pulling the object upwards or towards the receiver.

IV. EXPERIMENTS

Figure 1 represents the experimental setup built to test the handover task based on visual and haptic cues. A Kuka LBR iiwa 7 is equipped with a WSG-50 parallel gripper and an Intel Realsense D435i RGB-D camera mounted in an eye-in-hand configuration to retrieve visual cues. The haptic cues are detected by means of the SUNTouch tactile sensors [6] mounted on the fingertips and able to measure the 6D contact wrench. The camera runs at 60 Hz, the tactile sensor at 500 Hz, the gripper accepts velocity commands at 50 Hz, while the robot is controlled at 1 kHz. The control and design parameters are reported in Tab. I.

During the H2R handover, after a visual cue, the robot has to grasp an object directly from the human hand and place it in a predefined position on the shelf. The R2H experiments are the opposite, after a different visual cue, the robot grasps the object from the shelf to handover it to the human operator. Four objects are considered: a full and an empty plastic bottle, and two different cardboard boxes. Since the results are very similar for all the objects, only the first two are reported here but the accompanying video shows all the experiments and the related plots.

The first experiment is only devoted to testing the novel visual servoing controller with the disturbance estimated and compensated via the EKF. The results are shown in Fig. 6. Only for this experiment, the visual servoing accuracy ε_{vs} is set to zero so that the algorithm never exits from the visual servoing phase. The plot shows the visual servoing error (top) and the linear and angular control velocity norms (middle and bottom plot, respectively). The closed loop components depending on the error $\boldsymbol{e}$ are represented in blue, while the control action synthesized from the disturbance estimate $\hat{\boldsymbol{v}}_d$ is reported in red. During the experiment, the human operator randomly moves the object to generate both translational and rotational disturbances. After the initial exponential decrease, the error remains almost constant to near-zero values even if the human operator significantly moves the object. In fact, in this phase, the robot moves almost only due to the disturbance estimate control component (red lines).

TABLE I

CONTROL AND DESIGN PARAMETERS USED IN THE EXPERIMENTS

ν_w	0.98	ν_s	0.5	ϕ_z	1.0	ϕ_p	0.6
ε_{vs}	0.0075	λ	0.8	l	4000	σ_0	50
$\boldsymbol{N}$	$\text{diag}(5\boldsymbol{I}_3, 20\boldsymbol{I}_3)10^{-7}$		$\boldsymbol{H}$	$10^{-7}\boldsymbol{I}_{3n}$		$\boldsymbol{X}$	$10^{-7}\boldsymbol{I}_{3n}$

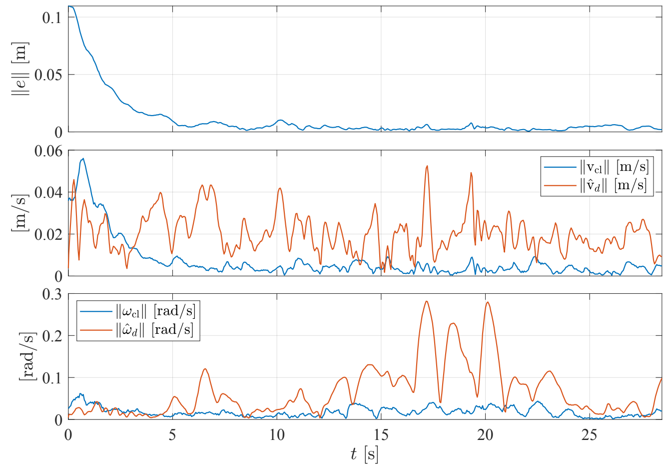

Fig. 6. First experiment: visual serving performance evaluation. Visual serving error norm (top); Control translational (middle) and rotational (bottom) velocities norm. The closed loop components are reported in blue, while the estimated disturbance is in red.

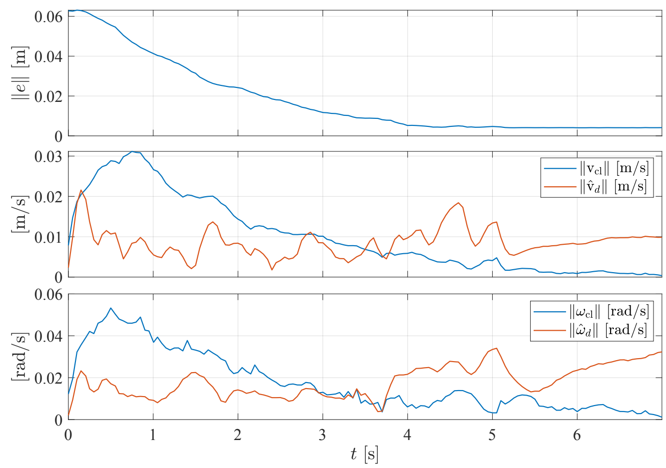

Fig. 7. Second experiment: H2R handover of a plastic bottle. Visual serving error norm (top); Control translational (middle) and rotational (bottom) velocities norm.

The second experiment consists in the handover of the plastic bottle in Fig. 1. The robot first waits for the visual gesture cue in Fig. 4-left, at the same time the corresponding hand location is acquired so that the robot can go in front of it in an open-loop fashion. After that, the Visual Servoing state of the H2R FSM is activated. The result is shown in Fig. 7 which is very similar to the previous experiment. The error exponentially decreases towards zero and, at the same time, the disturbance estimated by the EKF is applied as a corrective action to compensate for any motion imposed by the human hand. Figure 8-top shows the forces measured by the tactile sensors. At $t = 5$ s the object is grasped and the slipping avoidance algorithm is activated. Between $t = 5$ and $t = 7$ s we have the so-called *load sharing*

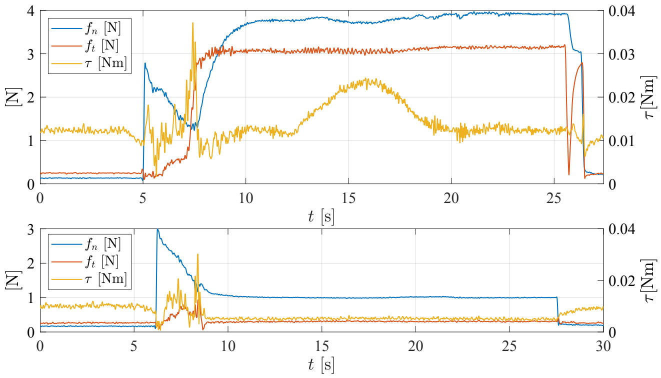

Fig. 8. Forces exchanged during the H2R operation with a full (top) and empty (bottom) plastic bottle.

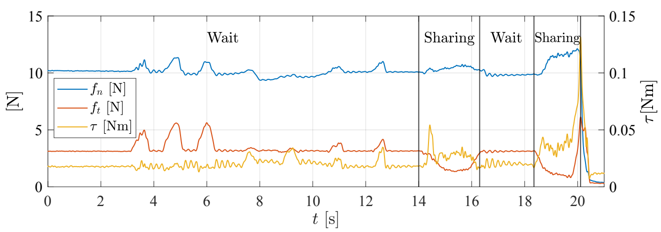

Fig. 9. Exchanged forces during the R2H handover. The FSM state transition phases are highlighted.

phase and the object weight is shared between the giver and the receiver. When the robot receiver moves back the human operator instinctively releases the grasp and the object holding becomes a receiver duty. At this time the tangential force (i.e., the weight) is 3 N and the slippage controller automatically imposes a grasping force of 3.7 N. After the handover, the robot is commanded to place the bottle on the shelf. During the robot motion, the measured torque τ_n varies as the gripper orientation varies while the object is fixed in hand. Finally, at $t = 26$ s the robot places the object on the shelf. Figure 8-bottom shows the same experiment repeated with an empty bottle. Note that, in such a case, even if the shape of the signals is almost the same, the measured force and torque are much smaller and the controller automatically applies a grasping force as low as 1 N.

Figure 9 shows the R2H experiment. The FSM (Fig. 5-bottom) waits for the R2H visual cue (Fig. 4-right). The hand location is acquired as in the H2R case, but this time the robot goes to the shelf to pick the object. The object weight is measured in the *Init* state and the robot presents the object in the handover location acquired beforehand during the gesture recognition phase. Then, the actual R2H handover driven by the haptic cues begins. The robot is in the *Wait* state and counteracts any external force applied to the object. This is evident between $t = 3$ and $t = 14$ s where the human operator touches the object from different directions (see also the accompanying video). At the same time the controller reacts by increasing the grasping force for each applied disturbance. At $t = 14$ s the human grasps the object, this is detected by condition (17), the FSM goes into the *Sharing* state and the robot is ready to release the object if the appropriate haptic cue is detected. In this experiment, at $t = 16.5$ s the human aborts the handover and releases the object (simulating a change of intention of the receiver), this is detected by the haptic cue in (18) and the FSM returns into the *Wait* state. At $t = 18.5$ s the sharing state is detected again, this time the human receiver pulls the object to fully grasp it, the haptic cue (19) is detected and the robot opens the gripper, thus completing the handover.

V. CONCLUSIONS

This work presented and experimentally validated a H2R and R2H handover strategy based on both visual and haptic cues. The experiments demonstrated that visual perception is important to both initiate the handover and to reliably reach the handover location, while haptic is of paramount importance during the physical handover to accurately modulate the grasping force and receive haptic cues from the partner.

REFERENCES

[1] V. Ortenzi, A. Cosgun, T. Pardi, W. Chan, E. Croft, and D. Kulic, "Object handovers: a review for robotics," *arXiv:2007.12952*, 2020.

[2] K. Strabala, M. K. Lee, A. Dragan, J. Forlizzi, S. S. Srinivasa, M. Cakmak, and V. Micelli, "Toward seamless human-robot handovers," *J. Hum.-Robot Interact.*, vol. 2, no. 1, p. 112–132, feb 2013.

[3] M. Costanzo, G. De Maria, and C. Natale, "Handover control for human-robot and robot-robot collaboration," *Frontiers in Robotics and AI*, vol. 8, 2021.

[4] A. H. Mason and C. L. MacKenzie, "Grip forces when passing an object to a partner," *Experimental Brain Research*, vol. 163, no. 2, pp. 173–187, 2005.

[5] M. Costanzo, G. De Maria, and C. Natale, "Two-fingered in-hand object handling based on force/tactile feedback," *IEEE Transactions on Robotics*, vol. 36, no. 1, pp. 157–173, 2020.

[6] M. Costanzo, G. De Maria, C. Natale, and S. Pirozzi, "Design and calibration of a force/tactile sensor for dexterous manipulation," *Sensors*, vol. 19, no. 4, p. 966, Feb 2019.

[7] N. Xydas and I. Kao, "Modeling of contact mechanics and friction limit surfaces for soft fingers in robotics, with experimental results," *The International Journal of Robotics Research*, vol. 18, no. 9, pp. 941–950, 1999.

[8] R. D. Howe and M. R. Cutkosky, "Practical force-motion models for sliding manipulation," *The International Journal of Robotics Research*, vol. 15, no. 6, pp. 557–572, 1996.

[9] A. Cavallo, M. Costanzo, G. De Maria, and C. Natale, "Modeling and slipping control of a planar slider," *Automatica*, vol. 115, 2020.

[10] V. Johannes, M. Green, and C. Brockley, "The role of the rate of application of the tangential force in determining the static friction coefficient," *Wear*, vol. 24, no. 3, pp. 381–385, 1973.

[11] K. Johanastrom and C. Canudas-de Wit, "Revisiting the lugre friction model," *IEEE Contr Syst Mag*, vol. 28, no. 6, pp. 101–114, 2008.

[12] M. Costanzo, "Control of robotic object pivoting based on tactile sensing," *Mechatronics*, vol. 76, p. 102545, 2021.

[13] E. Marchand, F. Spindler, and F. Chaumette, "Visp for visual servoing: a generic software platform with a wide class of robot control skills," *IEEE Rob & Aut Mag*, vol. 12, no. 4, pp. 40–52, 2005.

[14] F. Chaumette and S. Hutchinson, "Visual servo control. I. basic approaches," *IEEE Robot. Autom. Mag.*, vol. 13, no. 4, pp. 82–90, 2006.

[15] S. Benhimane and E. Malis, "Homography-based 2d visual tracking and servoing," *The International Journal of Robotics Research*, vol. 26, no. 7, pp. 661–676, 2007.

[16] L. Di Stefano, S. Mattoccia, and F. Tombari, "Zncc-based template matching using bounded partial correlation," *Pattern Recognition Letters*, vol. 26, no. 14, pp. 2129–2134, 2005.

[17] C. Tomasi and T. Kanade, "Detection and tracking of point features," School of Comp Sci, Carnegie Mellon Univ., Tech. Rep., 1991.

[18] [Online]. Available: https://github.com/TrinhNC/ros_hand_gesture_recognition