

A Visual-Based Shared Control Architecture for Remote Telemanipulation

Firas Abi-Farraj, Nicolò Pedemonte and Paolo Robuffo Giordano

Abstract—Cleaning up the past half century of nuclear waste represents the largest environmental remediation project in the whole Europe. Nuclear waste must be sorted, segregated and stored according to its radiation level in order to optimize maintenance costs. The objective of this work is to develop a shared control framework for remote manipulation of objects using visual information. In the presented scenario, the human operator must control a system composed of two robotic arms, one equipped with a gripper and the other one with a camera. In order to facilitate the operator’s task, a subset of the gripper motion are assumed to be regulated by an autonomous algorithm exploiting the camera view of the scene. At the same time, the operator has control over the remaining null-space motions w.r.t. the primary (autonomous) task by acting on a force feedback device. A novel force feedback algorithm is also proposed with the aim of informing the user about possible constraints of the robotic system such as, for instance, joint limits. Human/hardware-in-the-loop experiments with simulated slave robots and a real master device are finally reported for demonstrating the feasibility and effectiveness of the approach.

I. INTRODUCTION

Robotic telemanipulation is essential to ensure the operator’s safety when performing tasks in hazardous environments, such as nuclear sites [1], outer space [2] or underwater [3]. Several assisted teleoperation frameworks have been proposed in the literature in order to increase the user’s performances for different teleoperation tasks [4]–[9]. A classical assistance technique is virtual fixtures. First introduced in [4], the benefits of virtual fixtures in teleoperation can be compared to the help of a ruler when drawing a line [5]. The use of virtual fixtures leads to a well-proven increase of performances [4], [5]. On the other hand, they are highly task-dependent and the right fixture needs to be defined for every subtask [10]. In contrast with virtual fixtures, shared control frameworks were proposed in order to merge robot autonomy and human supervisory capabilities [6]–[9]. In a typical shared control architecture, the different subtasks are assigned in such a way that the robotic system controls low-level operations and the human operator ensures high-level decision making. The task allocation between the human operator and the robotic system is nevertheless one of the main challenges of shared control [11]. As suggested

by [12], the level of intervention of the robotic system should depend on the task complexity. For complex tasks, a highly-autonomous system is preferable from the user’s point of view [7], whereas a human-centered control architecture may be defined for easier tasks [8].

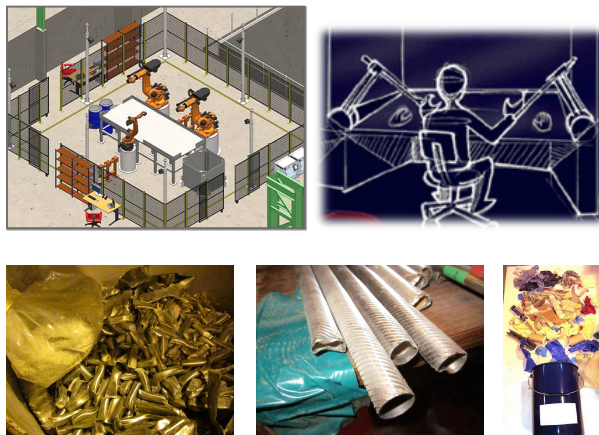


Fig. 1: Top figures: an illustrative sketch of the robotic testbed envisioned for the RoMaNS sort and segregation tasks (left) with a human operator in partial control of the overall motion (right). Bottom figures: examples of the typical waste material to be handled by the robotic cell.

In this paper, we present a novel shared control architecture for remote telemanipulation of objects of interest via visual and haptic feedback. This work is motivated by the European H2020 “Robotic Manipulation for Nuclear Sort and Segregation” (RoMaNS) project¹. In the RoMaNS scenario, a human operator has access to a system consisting of two robotic arms, one equipped with a gripper and the other one with a camera, with the goal of approaching and grasping nuclear waste for sort and segregation purposes (see Fig. 1). As a step towards this goal, we propose here a shared control architecture in which a (visual-based) autonomous algorithm is in charge of regulating a subset of the gripper DOFs for facilitating the approaching phase towards an object of interest. At the same time, the human operator is given the possibility to steer the gripper along the remaining null-space directions w.r.t. the main task by acting on a force feedback device. Due to the complexity of its motion, the camera is instead autonomously controlled so as to keep a good vantage point w.r.t. the scene and, thus, allow for a successful

F. Abi Farraj is with the CNRS at Irisa and Inria Rennes Bretagne Atlantique, Campus de Beaulieu, 35042 Rennes Cedex, France firas.abi-farraj@irisa.fr

N. Pedemonte is with the CNRS at Irisa and Inria Rennes Bretagne Atlantique, Campus de Beaulieu, 35042 Rennes Cedex, France nicolo.pedemonte@irisa.fr

P. Robuffo Giordano is with the CNRS at Irisa and Inria Rennes Bretagne Atlantique, Campus de Beaulieu, 35042 Rennes Cedex, France prg@irisa.fr

¹<http://www.h2020romans.eu/>

reconstruction of both the gripper and object poses. Finally, suitable force cues are fed to the operator in order to assist her/him during the task. In contrast with classical implementations, where the force feedback is, in general, proportional to some position/task error, the novel design of the proposed force cues can provide more structured information about the feasibility of the user’s commands w.r.t. possible constraints of the robotic system such as, e.g., joint limits. This falls into the scope of some recent works in the context of shared control of remote robots where novel criteria besides the sole (and typical) position/task error have been explored for teleoperation tasks, see, e.g., [6], [9]. Finally, we provide an experimental validation of the proposed architecture by using simulated slave robots and a real master device, which shows the effectiveness of our approach.

The rest of the paper is organized as follows. In Sect. II the general problem is introduced, while the shared control architecture is described in detail in Sect. III. Section IV reports the results of several experiments, and Sect. V concludes the paper and discusses some future directions.

II. PROBLEM SETTING

The scenario considered in this paper consists of two 6-dof serial manipulators, one equipped with a monocular (calibrated) camera and the other one with a gripper, aiming at grasping an object of interest (See Fig. 2). We consider three frames of reference: $\mathcal{F}_O : \{O_O; X_O, Y_O, Z_O\}$ attached to the object to be grasped, $\mathcal{F}_G : \{O_G; X_G, Y_G, Z_G\}$ attached to the gripper and $\mathcal{F}_C : \{O_C; X_C, Y_C, Z_C\}$ attached to the camera. We assume that Z_G is aligned with the gripper approaching direction, and that (as usual) Z_C is aligned with the camera optical axis.

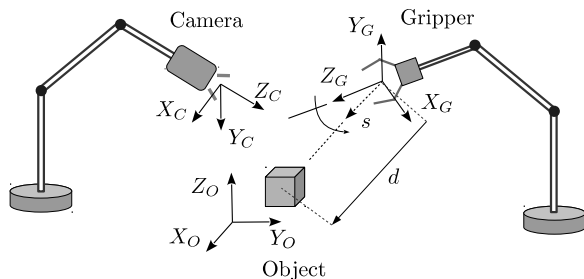


Fig. 2: An illustrative representation of the two 6-dof serial manipulator arms equipped with a camera and a gripper, respectively, together with other quantities of interest

We let $({}^C P_G, {}^C R_G) \in \mathbb{R}^3 \times SO(3)$ represent the 3D pose of \mathcal{F}_G w.r.t. \mathcal{F}_C expressed in \mathcal{F}_C and, similarly, $({}^C P_O, {}^C R_O) \in \mathbb{R}^3 \times SO(3)$ represent the 3D pose of \mathcal{F}_O w.r.t. \mathcal{F}_C expressed in \mathcal{F}_C . In the context of the RoMaNS project, we can assume that an accurate enough 3D model of both the object to be grasped and of the gripper is available beforehand. This allows to leverage any model-based tracker, such as those present in the ViSP library [13], for retrieving *online* a reliable estimation of the camera/object and

camera/gripper relative poses in the camera frame². We can then consider the relative gripper/object poses in the gripper frame \mathcal{F}_G , that is, the quantities ${}^G R_O = {}^G R_C {}^C R_O$ and ${}^G P_O = {}^G R_C ({}^C P_O - {}^C P_G)$, as known from the corresponding reconstructed poses in \mathcal{F}_C . Finally, we let $(v_C, \omega_C) \in \mathbb{R}^6$ and $(v_G, \omega_G) \in \mathbb{R}^6$ represent the camera and gripper linear/angular velocities expressed in their respective frames, which will play the role of control inputs in the following developments.

The goal of the proposed shared control architecture is to (i) let an autonomous algorithm control a part of the gripper/object relative pose for facilitating the approaching phase towards the object to be grasped, (ii) let a human operator control the remaining free DOFs of the gripper/object relative pose via a force-feedback device, (iii) provide the human operator with online force cues informing about the feasibility of her/his motion commands w.r.t. possible constraints of the robotic system, and, finally, (iv) let an autonomous algorithm control the camera motion so as to keep a suitable vantage point w.r.t. the observed scene (i.e., both the gripper and the object).

We now proceed to detail the components of the shared control architecture.

III. SHARED CONTROL ARCHITECTURE

A. Gripper control

As discussed above, the gripper controller should constrain a suitable subset of the gripper DOFs while delegating to a human operator (or to any other external high-level ‘planner’) the possibility of actuating the remaining free DOFs for adjusting the approaching phase towards the object to be grasped. This goal can be achieved by considering the gripper controlled DOFs as a ‘primary task’ to be autonomously regulated towards some desired value, and by allowing the human operator to actuate the resulting null-space motions w.r.t. the main task. To this end, let $s \in \mathbb{R}^m$, $m < 6$, be a m -dimensional subset of the gripper/object relative pose representing the primary (autonomous) task, and let $\dot{s} = L_s \begin{bmatrix} v_G \\ \omega_G \end{bmatrix}$, with $L_s \in \mathbb{R}^{m \times 6}$ being the corresponding interaction matrix with $\text{rank}(L_s) = r \leq m$ and, thus, $\dim(\ker(L_s)) = 6 - r = n$. Let also $N = [\dots n_i \dots] \in \mathbb{R}^{6 \times n}$ be a basis of the n -dimensional null-space of L_s , and $\lambda = [\dots \lambda_i \dots] \in \mathbb{R}^n$ a vector collecting the n pseudo-velocity commands λ_i that will be exploited for actuating the individual null-space motions n_i .

The following control law³

$$\begin{bmatrix} v_G \\ \omega_G \end{bmatrix} = k_G L_s^\dagger (s_d - s) + \sum_{i=1}^n \lambda_i n_i, \quad k_G > 0, \quad (1)$$

with L_s^\dagger representing the Moore-Penrose pseudo-inverse of matrix L_s , achieves the two stated objectives:

²More details on this point will be given in Sect. IV.

³In the Visual Servoing nomenclature, controller (1) would be classified as a eye-to-hand PBVS scheme since it aims at regulating the 3D object/gripper pose reconstructed from an off-board camera [14].

- 1) autonomous regulation of the m ‘controlled DOFs’ s (the primary task) towards a desired value s_d ;
- 2) concurrent actuation of the remaining n null-space motions spanned by vectors \mathbf{n}_i via the corresponding pseudo-velocity commands λ_i (which will be specified online by a human operator).

Clearly, any basis \mathbf{N} for the null-space of \mathbf{L}_s is a valid choice in (1). However, for the sake of providing a usable interface to the human operator, it is important to select a basis for which the motion directions \mathbf{n}_i have a clear/intuitive physical meaning in order to ease the operator’s intervention during the task.

As an illustrative (but significant) case study, in this work we choose to define the primary task variables s as the 3D direction towards the object to be grasped, i.e., the unit-norm vector

$$\mathbf{s} = \frac{{}^G\mathbf{P}_O}{\|{}^G\mathbf{P}_O\|} \in \mathbb{S}^2, \quad (2)$$

see Fig. 2. This is meant to ensure an autonomous/precise control of the gripper alignment w.r.t. the target object during the pre-grasping approaching phase. The interaction matrix associated to this choice of s is then (see, e.g., [15], [16])

$$\mathbf{L}_s = \left[-\frac{1}{d}\mathbf{P}_s \quad [\mathbf{s}]_{\times} \right] \in \mathbb{R}^{3 \times 6} \quad (3)$$

where $\mathbf{P}_s = \mathbf{I} - \mathbf{s}\mathbf{s}^T$ is the orthogonal projector onto the tangent space of the unit sphere \mathbb{S}^2 at s , $d = \|{}^G\mathbf{P}_O\|$, and $[\cdot]_{\times}$ represents the usual skew-symmetric matrix operator. Note that, in this case, $m = 3$ but $r = \text{rank}(\mathbf{L}_s) = 2$ because of the unit norm constraint in (2). Therefore, there will exist $n = 4$ independent directions in the null-space of \mathbf{L}_s . Also, since $\mathbf{L}_s^T \mathbf{s} = \mathbf{0}$ and $\text{span}(\mathbf{L}_s^\dagger) = \text{span}(\mathbf{L}_s^T)$, the control law (1) can be simplified into

$$\begin{bmatrix} v_G \\ \omega_G \end{bmatrix} = k_G \mathbf{L}_s^\dagger s_d + \sum_{i=1}^n \lambda_i \mathbf{n}_i. \quad (4)$$

Among the many possibilities, we found a convenient choice for the null-space basis \mathbf{N} to be

$$\mathbf{n}_1 = \begin{bmatrix} \mathbf{s} \\ \mathbf{0} \end{bmatrix}, \mathbf{n}_2 = \begin{bmatrix} \mathbf{0} \\ \mathbf{s} \end{bmatrix}, \mathbf{n}_3 = \begin{bmatrix} -[\mathbf{s}]_{\times} \mathbf{e}_y \\ -\mathbf{P}_s \mathbf{e}_y / d \end{bmatrix}, \mathbf{n}_4 = \begin{bmatrix} [\mathbf{s}]_{\times} \mathbf{e}_x \\ \mathbf{P}_s \mathbf{e}_x / d \end{bmatrix}, \quad (5)$$

with $\mathbf{e}_x = [1 \ 0 \ 0]^T$ and $\mathbf{e}_y = [0 \ 1 \ 0]^T$, see the Appendix. The advantage of this basis is that it yields an ‘interface’ to the human operator with a clear (and decoupled) physical interpretation. In particular, when plugging (5) in (4) one obtains the following:

- the null-space direction \mathbf{n}_1 realizes a motion along s which results in $\dot{d} = \lambda_1$. Therefore, the operator gains direct control over the rate of change of the distance $d(t)$ along the pointing direction towards the target. Furthermore, $d(t)$ is not affected by any other null-space direction;
- the null-space direction \mathbf{n}_2 realizes a rotation about s with angular speed λ_2 . Therefore, the operator gains direct control over the gripper angular velocity about the pointing direction towards the target. Furthermore,

the rotation about s is not affected by any other null-space direction;

- the null-space directions \mathbf{n}_3 and \mathbf{n}_4 realize two coordinated motions (linear/angular velocity) that displace the gripper over a sphere centered at the target object (with, thus, $d(t) = \text{const}$). In particular, \mathbf{n}_3 makes the gripper move in the positive \mathbf{X}_G direction and \mathbf{n}_4 in the positive \mathbf{Y}_G direction *without* affecting, in both cases, the distance along s (only actuated by \mathbf{n}_1) and the angle about s (only actuated by \mathbf{n}_2).

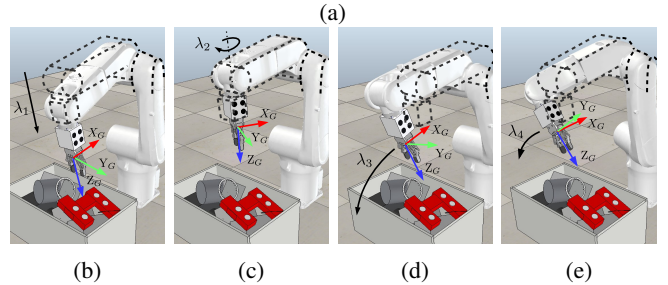
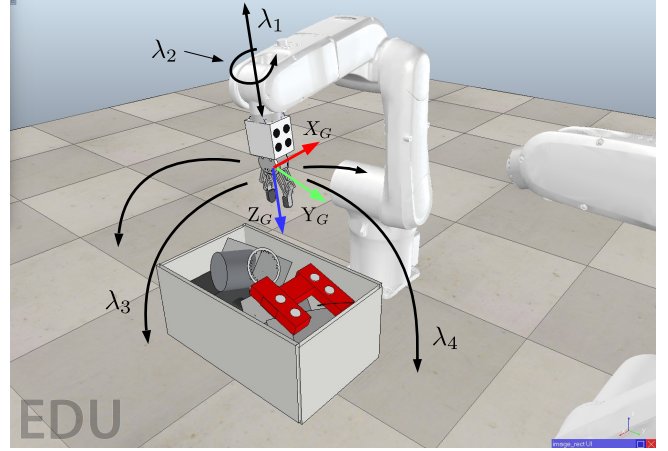


Fig. 3: A visualization of the four null-space directions (5). (a): the four directions and the associated pseudo-velocity commands λ_i with the gripper in a neutral pose. (b–e): an illustration of the motion resulting from actuating each individual direction \mathbf{n}_i

Figure 3 provides a visual illustration of the four null-space directions (5).

We conclude by noting that, as discussed in Sect. IV, the reported experimental case studies always considered a (typical) situation in which $s_d = \mathbf{Z}_G = [0 \ 0 \ 1]^T$ so as to force the target direction $s(t)$ to be aligned with the gripper approaching direction \mathbf{Z}_G . In this case, the null-space motions (5) evaluated at $s = s_d$ reduce to (i) a translation along \mathbf{Z}_G with speed λ_1 , (ii) a rotation about \mathbf{Z}_G with speed λ_2 , a translation along \mathbf{X}_G with speed λ_3 (combined with a rotation about \mathbf{Y}_G with speed λ_3/d), and a translation along \mathbf{Y}_G with speed λ_4 (combined with a rotation about \mathbf{X}_G with speed λ_4/d).

B. Camera control

As discussed in Sect. II, the camera should control its motion in order to keep a suitable vantage point w.r.t. both the gripper and the target object for allowing an accurate 3D reconstruction of their poses. As an illustrative example, we consider here the simple task of maintaining the projection of the object position ${}^C P_O$ and of the gripper position ${}^C P_G$ at some desired locations on the camera image plane for ensuring that the gripper and the object stay within visibility during operation⁴.

Figure 4 illustrates the main quantities of interest with, in particular, \mathbf{p}_O representing the point feature projection of ${}^C P_O$, and \mathbf{p}_G the point feature projection of ${}^C P_G$. Since

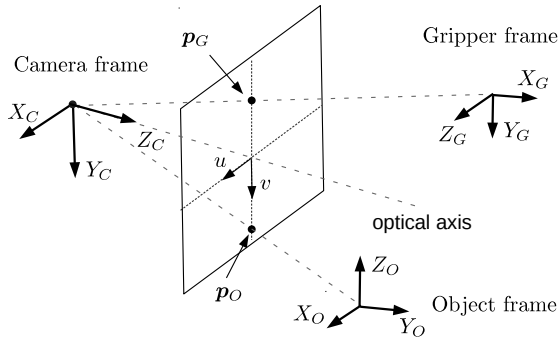


Fig. 4: Feature points \mathbf{p}_G and \mathbf{p}_O exploited for controlling the camera motion

our aim is to have control over the image plane location of ${}^C P_O$ and ${}^C P_G$, we resort to a IBVS approach [14] with the values of \mathbf{p}_O and \mathbf{p}_G obtained from the projection of the (reconstructed) 3D poses ${}^C P_O$ and ${}^C P_G$. As the target object is assumed static in the scene, the dynamics of \mathbf{p}_O is just

$$\dot{\mathbf{p}}_O = \mathbf{L}_p(\mathbf{p}_O, Z_O) \begin{bmatrix} \mathbf{v}_C \\ \omega_C \end{bmatrix}$$

where $\mathbf{L}_p \in \mathbb{R}^{2 \times 6}$ is the well-known interaction matrix for a point feature [14] and Z_O is the depth associated to \mathbf{p}_O . The dynamics of \mathbf{p}_G is instead also affected by the gripper own motion generated by controller (4). In particular, it is straightforward to obtain the relationship

$$\dot{\mathbf{p}}_G = \mathbf{L}_p(\mathbf{p}_G, Z_G) \begin{bmatrix} \mathbf{v}_C - {}^C \mathbf{v}_G \\ \omega_C \end{bmatrix}$$

where ${}^C \mathbf{v}_G = {}^C \mathbf{R}_G \mathbf{v}_G$ is the gripper linear velocity expressed in \mathcal{F}_C , see, e.g., [17]. Letting now $\mathbf{p} = [\mathbf{p}_O^T \mathbf{p}_G^T]^T$, the following IBVS controller

⁴More complex objectives could be considered for the camera control, for instance aiming at maximizing online some performance index of the model-based trackers employed for the gripper/target pose reconstruction. These possibilities will be considered in future extensions of the shared control architecture presented in this work.

$$\begin{bmatrix} \mathbf{v}_C \\ \omega_C \end{bmatrix} = \begin{bmatrix} \mathbf{L}_p(\mathbf{p}_O, Z_O) \\ \mathbf{L}_p(\mathbf{p}_G, Z_G) \end{bmatrix}^\dagger \left(k_C (\mathbf{p}_d - \mathbf{p}) + \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix} \right) + \begin{bmatrix} {}^C \mathbf{v}_G \\ \mathbf{0} \end{bmatrix}, \quad (6)$$

$k_C > 0$, achieves the regulation of \mathbf{p} towards a desired value \mathbf{p}_d while compensating for the (known) gripper motion generated by (4).

C. Haptic Feedback

The final component of the shared control architecture is the design of some suitable force feedback cues which, as discussed in Sect. II, are meant to inform the human operator about the feasibility of her/his pseudo-velocity commands λ w.r.t. possible constraints/requirements of the gripper/arm robotic system.

Following the classical *bilateral force-feedback* framework [18], [19], we then assume the presence of a *master device* upon which the operator can act for sending the commands λ to the *slave side* (the gripper/manipulator arm) and receiving force feedback cues. The master device is modeled as a generic (gravity pre-compensated) mechanical system

$$\mathbf{M}(\mathbf{x}_M) \ddot{\mathbf{x}}_M + \mathbf{C}(\mathbf{x}_M, \dot{\mathbf{x}}_M) \dot{\mathbf{x}}_M = \boldsymbol{\tau} + \boldsymbol{\tau}_h \quad (7)$$

where $\mathbf{x}_M \in \mathbb{R}^n$ is the device configuration vector (with same dimension as the human commands λ), $\mathbf{M}(\mathbf{x}_M) \in \mathbb{R}^{n \times n}$ is the positive-definite and symmetric inertia matrix, $\mathbf{C}(\mathbf{x}_M, \dot{\mathbf{x}}_M) \in \mathbb{R}^{n \times n}$ accounts for Coriolis/centrifugal terms, and $\boldsymbol{\tau}, \boldsymbol{\tau}_h \in \mathbb{R}^n$ are the control and human forces, respectively.

The human control actions are implemented by setting

$$\lambda = \mathbf{K}_\lambda \mathbf{x}_M, \quad (8)$$

with $\mathbf{K}_\lambda \in \mathbb{R}^{n \times n}$ being a diagonal matrix of positive scaling factors. This coupling then allows the operator to directly control the speed along the n null-space directions \mathbf{n}_i by adjusting the position of the master device. The force feedback is instead designed as

$$\boldsymbol{\tau} = -\mathbf{B}_M \dot{\mathbf{x}}_M - \mathbf{K}_M \mathbf{x}_M + \mathbf{f}. \quad (9)$$

Here, $\mathbf{B}_M \in \mathbb{R}^{n \times n}$ is a positive definite damping matrix for stabilizing the haptic device and $\mathbf{K}_M \in \mathbb{R}^{n \times n}$ is a positive definite diagonal matrix meant to implement a ‘soft spring’ centered at the device rest position⁵. Vector $\mathbf{f} = [\dots f_i \dots]^T \in \mathbb{R}^n$ represents instead the force cues provided to the human operator: as explained, the design of these cues is aimed at informing the operator about the feasibility of each motion command λ_i w.r.t. possible constraints/limitations of the gripper/arm system such as, for instance, proximity to joint limits, to singularities, or to collisions with the surrounding environment. We now proceed to detail the general structure of the proposed cueing algorithm which will be then exploited in Sect. IV for

⁵Therefore, by means of this spring the user will be provided with a perception of the distance from a zero-commanded velocity.

implementing a cueing action informative about joint limit avoidance.

Let $\mathbf{q} \in \mathbb{R}^6$ be the joint configuration vector of the manipulator arm carrying the gripper, and $\mathbf{J}_G(\mathbf{q}) \in \mathbb{R}^{6 \times 6}$ be the geometric Jacobian mapping joint velocities $\dot{\mathbf{q}}$ onto the gripper linear/angular velocities $(\mathbf{v}_G, \boldsymbol{\omega}_G)$. Let also $H(\mathbf{q}) \geq 0$ be a scalar cost function quantifying the proximity to any constraint of interest (the closer to a constraint, the larger the value of $H(\mathbf{q})$). Intuitively, our idea is to design each f_i so as to inform about how much $H(\mathbf{q})$ would decrease by moving along the i -th null-space direction \mathbf{n}_i . Such a force feedback can then assist the operator in understanding (i) which directions \mathbf{n}_i are close to be unfeasible because of the robot constraints (thanks to the magnitude of f_i) and (ii) how to actuate the direction \mathbf{n}_i for moving away from the robot constraints (thanks to the sign of f_i). This behavior can be obtained by designing f_i as

$$f_i = - \left(\frac{\partial H(\mathbf{q})}{\partial \mathbf{q}} \right)^T \mathbf{J}_G^{-1}(\mathbf{q}) \mathbf{n}_i, \quad (10)$$

i.e., by projecting the joint motion caused by the i -th null-space direction \mathbf{n}_i onto the negative gradient of $H(\mathbf{q})$.

As a final step, we make use of the *passive set-position modulation* (PSPM) algorithm [20] for coping with the typical stability issues of any bilateral force feedback loop because of communication delays, packet losses, master/slave kinematic/dynamic dissimilarities, and other shortcomings. To this end, let $\mathbf{y}(t) = \mathbf{K}_M^{-1} \mathbf{f}(t)$ and rearrange (9) as

$$\boldsymbol{\tau} = -\mathbf{B}_M \dot{\mathbf{x}}_M - \mathbf{K}_M (\mathbf{x}_M - \mathbf{y}(t)). \quad (11)$$

The PSPM action modulates the (arbitrary) signal $\mathbf{y}(t)$ into a possibly attenuated version $\bar{\mathbf{y}}(t)$ which, when plugged into (11), ensures input/output stability (passivity) of the master device. This is then sufficient for guaranteeing stability (passivity) of the overall bilateral teleoperation, see [20] for more details and [21], [22] for some recent examples of the use of the PSPM algorithm.

IV. EXPERIMENTAL RESULTS

We now report the results of several experiments conducted to illustrate and validate the proposed shared control architecture. Figure 5 depicts the experimental setup. The master side consists of the Haption VIRTUOSE 6D haptic device⁶, a high performance force feedback device with three translational DOFs and three rotational DOFs. The maximum force/torque is about 30 [N]/3 [Nm], the workspace has a spherical-like shape with an approximated radius of 0.9 [m], and the device exchanges data over ethernet with a control PC at 1 kHz. Four DOFs of the Haption device were left unconstrained for actuating the $n = 4$ null-space directions \mathbf{n}_i in (5), while the remaining two DOFs were constrained via software to a constant value, see Fig. 5(b). As explained in Sect. III-C, the position of the master along the four free DOFs was coupled to the pseudo-velocity commands $\boldsymbol{\lambda}$ via (8).

⁶www.haption.com.

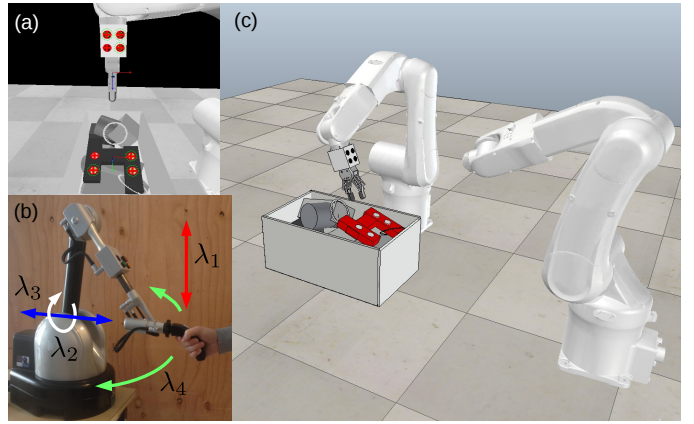


Fig. 5: Experimental Setup: (a) camera image with the segmentation of the fiducial markers exploited by the model-based tracker for reconstructing the gripper and object pose in the camera frame; (b) the VIRTUOSE 6D haptic device with the four DOFs exploited for actuating the null-space motions \mathbf{n}_i in (5); (c) the two simulated 6-DOF manipulators carrying the camera and the gripper.

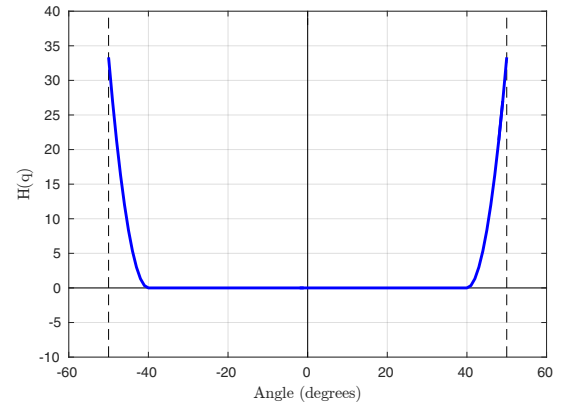


Fig. 6: The quadratic penalty cost function used to generate force cues when approaching joint limits. In the illustrated example, $q_{i,max} = 50$ degrees, $q_{i,min} = -50$ degrees and $q_{th} = 10$ degrees.

The slave side consists of two 6-DOF Viper S850 robotic arms carrying the gripper and the camera, and simulated in the popular V-REP environment⁷. The poses of the gripper and of the target object in the camera frame were reconstructed by feeding the model-based ViSP tracker [13] with the segmented location of some fiducial markers acquired at 30 Hz, see Fig. 5(a).

Finally, in all the reported experiments we set the desired $\mathbf{s}_d = [0 \ 0 \ 1]^T$ in (4) so as to force the object pointing direction $\mathbf{s}(t)$ to be always aligned with the gripper approaching direction \mathbf{Z}_G . Furthermore, we considered proximity to joints limits as a representative constraint of the gripper/manipulator arm system. Function $H(\mathbf{q})$ in (10) was then designed as a (classical) quadratic penalty cost (see Fig. 6):

⁷www.coppeliarobotics.com.

$$H(\mathbf{q}) = k_H \sum_{i=1}^6 h_i(\mathbf{q}),$$

with

$$h_i(\mathbf{q}) = \begin{cases} (q_i - (q_{i,max} - q_{th}))^2, & \text{if } q_i \geq q_{i,max} - q_{th} \\ (q_{i,min} + q_{th} - q_i)^2, & \text{if } q_i \leq q_{i,min} + q_{th} \\ 0, & \text{otherwise} \end{cases}$$

where $(q_{i,max}, q_{i,min})$ are the maximum/minimum range for the i -th joint, q_{th} is a user-defined threshold defining the activation region inside which the user will receive a force feedback, and k_H is a scaling factor.

We now report two sets of experiments conducted for validating the described shared control architecture. The interested reader is also encouraged to watch the video attachment for a better appreciation of the combined gripper/camera motion under the operator's commands.

A. First Experiment

The first experiment is meant to illustrate the main features of our approach, i.e., the possibility of actuating the n null-space directions \mathbf{n}_i while receiving a force feedback informing about the proximity in violating the robot joint limits. The experiment is split into three main phases: during the first phase ($0 \text{ [s]} \leq t \leq 32 \text{ [s]}$), the operator keeps the haptic device at its neutral position and gives no commands along the directions \mathbf{n}_i ($\lambda = \mathbf{0}$). During the second phase ($32 \text{ [s]} \leq t \leq 105 \text{ [s]}$), the operator starts actuating the null-space directions \mathbf{n}_i one at the time with the aim of isolating the effects of each individual command. Then, during the last phase ($t \geq 105 \text{ [s]}$), the operator provides a generic motion command that actuates all the null-space directions at once.

Figures 7(a–e) report the experimental results. In particular, Figs. 7(a–b) show the behavior of the target object direction $\mathbf{s}(t)$ and of the two point features $\mathbf{p}(t)$ during motion. One can then verify how the gripper/camera controllers (4)–(6) are able to regulate the values of $\mathbf{s}(t)$ and of $\mathbf{p}(t)$ towards their desired (constant) values during the whole experiment despite the various null-space motions commanded by the operator (as expected). Presence of noise in the plots is mainly due to the 3D pose estimation by the ViSP model-based tracker which, as in any vision-based reconstruction, propagates the image noise in segmenting the fiducial markers on the gripper/object.

Figures 7(c–d) report the behavior of the pseudo-velocity commands $\lambda(t)$ and of the force cues $\mathbf{f}(t)$, while Fig. 7(e) shows the behavior of the cost function $H(\mathbf{q}(t))$ over time. During the second phase of the experiment ($20 \text{ [s]} \leq t \leq 105 \text{ [s]}$) the user commanded the n individual null-space motions until (intentionally) approaching the joint limits and, as a consequence, she/he received a corresponding force feedback cue. One can verify how the force cues \mathbf{f}_i were activated only when approaching a joint limit with a corresponding increase of $H(\mathbf{q})$. It is also interesting to note that, although the user commanded an individual null-space direction \mathbf{n}_i at the time, force cues along multiple axes were

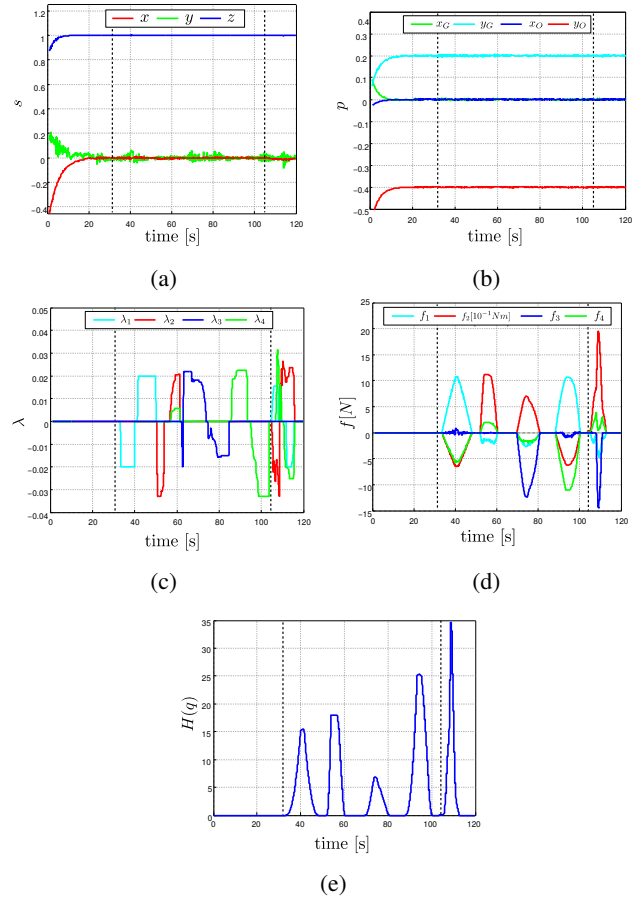


Fig. 7: Results of the first experiment. (a) behavior of the primary task $\mathbf{s}(t)$; (b) behavior of the gripper and object positions $\mathbf{p} = [\mathbf{p}_G^T \ \mathbf{p}_O^T]^T$ on the image plane; (c) Behavior of the four operator's commands λ for actuating the null-space directions \mathbf{n}_i . (d) Behavior of the four force cues \mathbf{f}_i displayed to the human operator during the robot motion; (e) behavior of the scalar cost function $H(\mathbf{q})$ quantifying the proximity to joint limits.

generated when approaching a joint limit. This is expected since, obviously, multiple directions in the joint space can potentially lead to a decrease of $H(\mathbf{q})$. Nevertheless, the direction of the main force cue (in terms of magnitude) was always well correlated with the direction actuated by the human operator, who thus felt the largest “resistant force” opposing her/his commands.

Presence of additional force cues along different axes than the one individually actuated can, anyway, have a beneficial role. Indeed, as discussed in Sect. III-C, this kind of feedback can help the operator understanding which directions to (potentially) actuate in order to make the cost function $H(\mathbf{q})$ decrease. The operator can then keep on commanding a null-space motion along a direction of interest (e.g., for approaching the target object), while being automatically guided along the other null-space directions so as to stay away as much as possible from any joint limits. As explained, these considerations can be straightforwardly generalized to

any additional criterium of interest (such as singularities or collision avoidance) by a proper shaping of the cost function $H(\mathbf{q})$.

B. Second Experiment

The second experiment is meant to complement the previous one by showing the effectiveness of the force cues \mathbf{f} in guiding the operator away from the considered robot constraints (joint limits). In this experiment, the operator first intentionally steers the robot towards a joint limit by actuating one of the four motion directions \mathbf{n}_i and resisting to the received force feedback. As the force cues become significant, the operator then stops applying her/his command and passively follows the master device guided by the forces f_i . This sequence is repeated twice over two different motion directions, and the obtained results are illustrated in Fig. 8(a–e).

Analogously to the previous case, Figs. 8(a–b) illustrate how the gripper/camera controllers (4)–(6) effectively regulated the quantities $\mathbf{s}(t)$ and $\mathbf{p}(t)$ at their desired values for the whole experiment duration (as, again, expected). Figures 8(c–e) show the operator’s commands $\boldsymbol{\lambda}(t)$, the received force cues $\mathbf{f}(t)$ and the behavior of $H(\mathbf{q}(t))$. During the first phase of the experiment ($22 \text{ [s]} \leq t \leq 37 \text{ [s]}$), the operator intentionally steered the manipulator towards a joint limit by actuating λ_3 . As shown in Fig. 8(e), this caused an increase of the cost function $H(\mathbf{q})$ and an associated activation of the force cues f_i along some axes of the master device. As the operator stopped commanding her/his motion ($t = 37 \text{ [s]}$) and passively followed the received cues, the cost function $H(\mathbf{q}(t))$ could quickly decrease thus moving away from the joint limit constraints. We also note the good correspondence between the activated force feedback signals (mainly f_1 and f_3) and the actuated motion directions (mainly λ_1 and λ_3) during this latter phase. A similar pattern can also be found during the second part of the experiment ($70 \text{ [s]} \leq t \leq 75 \text{ [s]}$) in which the operator commanded a different motion direction (λ_4) for steering the robot towards the joint limits and then passively followed the received force cues. Again, the received cues were helpful in quickly guiding the operator towards a configuration far from any joint limit.

In conclusion, the chosen force cues proved to be both informative and efficient in assisting the user in keeping the gripper/manipulator away from undesired configuration by either moving back along the operator’s commanded direction, or by manoeuvring over the other available motion directions as a function of the magnitude (and sign) of the received haptic information.

V. CONCLUSIONS

In this work we presented a shared control framework for allowing a human operator being in partial control of the pre-grasp approaching phase towards an object of interest via two serial manipulator arms, one carrying a gripper and another one a camera looking at the scene (gripper and object). A visual-servoing control law was implemented for

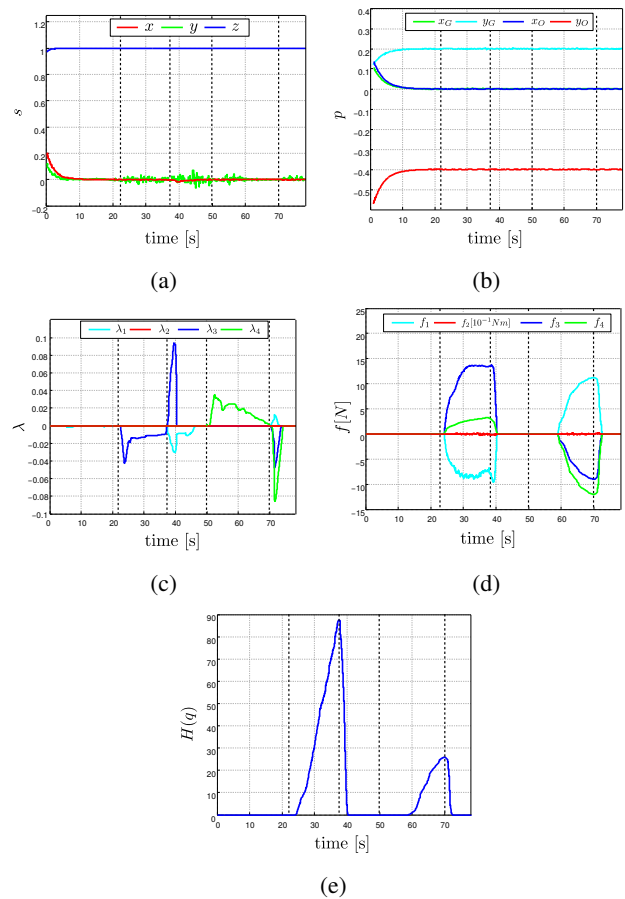


Fig. 8: Results of the second experiment. (a) behavior of the primary task $\mathbf{s}(t)$; (b) behavior of the gripper and object positions $\mathbf{p} = [\mathbf{p}_G^T \ \mathbf{p}_O^T]^T$ on the image plane; (c) Behavior of the four operator’s commands $\boldsymbol{\lambda}$ for actuating the null-space directions \mathbf{n}_i . (d) Behavior of the four force cues f_i displayed to the human operator during the robot motion; (e) behavior of the scalar cost function $H(\mathbf{q})$ quantifying the proximity to joints limits.

the autonomous part of the architecture (control of a subset of the gripper/object DOFs), and the operator was given the possibility of directly commanding the remaining null-space directions w.r.t. the main visual servoing task. Moreover, an informative high-level haptic feedback was proposed with the aim of informing the operator about the feasibility of her/his motion commands against possible constraints of the considered robotic system (in particular, joint limits of the manipulator arm carrying the gripper). Finally, the camera motion was optimized so as to keep a good vantage point w.r.t. the scene (object and gripper). The theoretical claims of the paper were successfully tested via a set of a human/hardware-in-the-loop simulation where a human operator acted on a real force-feedback master device for commanding two simulated serial manipulator arms.

We are currently working towards a fully experimental validation of the shared control architecture by employing two real 6-DOF manipulator equipped with a gripper/camera.

To this end, we are also considering to use vision-based reconstruction methods able to handle more complex scenes than the ones considered in this work. Furthermore, we are interested in evaluating other criteria for generating force cues than the sole proximity to joint limits. These could range from simple manipulability and collision avoidance measures, to more sophisticated indicators reflecting, for example, the quality of the gripper/object visual reconstruction given the current pose of the camera in the scene. The use of a higher level shared control scheme combining trajectory planning (such as in [6], [9]) with a grasping pose generator able to generate grasping poses online depending on the operator's inputs will also to be considered. Finally, we aim at running user studies for assessing the effectiveness of the described algorithm in a principled way.

APPENDIX

We first show that vectors \mathbf{n}_i in (5) span the null-space of \mathbf{L}_s in (3): (i) $\mathbf{L}_s \mathbf{n}_1 = \mathbf{P}_s \mathbf{s} / d = \mathbf{0}$ because of the properties of \mathbf{P}_s ; (ii) $\mathbf{L}_s \mathbf{n}_2 = [\mathbf{s}]_{\times} \mathbf{s} = \mathbf{0}$; (iii) $\mathbf{L}_s \mathbf{n}_3 = (\mathbf{P}_s [\mathbf{s}]_{\times} - [\mathbf{s}]_{\times} \mathbf{P}_s) \mathbf{e}_y / d = \mathbf{0}$ since $\mathbf{P}_s [\mathbf{s}]_{\times} = [\mathbf{s}]_{\times} \mathbf{P}_s = [\mathbf{s}]_{\times}$, being \mathbf{P}_s the orthogonal projector onto the plane orthogonal to \mathbf{s} which is spanned by $[\mathbf{s}]_{\times}$; (iv) $\mathbf{L}_s \mathbf{n}_4 = ([\mathbf{s}]_{\times} \mathbf{P}_s - \mathbf{P}_s [\mathbf{s}]_{\times}) \mathbf{e}_x / d = \mathbf{0}$ because of the same reasons.

Next, we show the orthogonality among the null-space vectors: $\mathbf{n}_1^T \mathbf{n}_2 = 0$ by inspection, $\mathbf{n}_1^T \mathbf{n}_3 = -\mathbf{s}^T [\mathbf{s}]_{\times} \mathbf{e}_y = 0$, $\mathbf{n}_1^T \mathbf{n}_4 = \mathbf{s}^T [\mathbf{s}]_{\times} \mathbf{e}_x = 0$, $\mathbf{n}_2^T \mathbf{n}_3 = -\mathbf{s}^T \mathbf{P}_s \mathbf{e}_y / d = 0$, $\mathbf{n}_2^T \mathbf{n}_4 = \mathbf{s}^T \mathbf{P}_s \mathbf{e}_x / d = 0$.

ACKNOWLEDGMENTS

This work was supported by the EU H2020 RoMaNS project 645582.

REFERENCES

- [1] R. Schilling, "Telerobots in the nuclear industry: a manufacturer's view," *Industrial Robots*, vol. 19, no. 2, pp. 3–4, 1992.
- [2] J. Wright, A. Trebi-Ollennu, F. Hartman, B. Cooper, S. Maxwell, J. Yen, and J. Morrison, "Driving a rover on mars using the rover sequencing and visualization program," in *International Conference on Instrumentation, Control and Information Technology (Okayama University, Okayama 2005)*, 2005.
- [3] R. R. Murphy, K. L. Dreger, S. Newsome, J. Rodocker, E. Steimle, T. Kimura, K. Makabe, F. Matsuno, S. Tadokoro, and K. Kon, "Use of remotely operated marine vehicles at minamisanriku and rikuzentakata japan for disaster recovery," in *Safety, Security, and Rescue Robotics (SSRR), 2011 IEEE International Symposium on*. IEEE, 2011, pp. 19–25.
- [4] L. B. Rosenberg, "Virtual fixtures: Perceptual tools for telerobotic manipulation," in *Virtual Reality Annual International Symposium, 1993., 1993 IEEE*. IEEE, 1993, pp. 76–82.
- [5] J. J. Abbott, P. Marayong, and A. M. Okamura, "Haptic virtual fixtures for robot-assisted manipulation," in *Robotics research*. Springer, 2007, pp. 49–64.
- [6] C. Masone, A. Franchi, H. H. Bühlhoff, and P. Robuffo Giordano, "Interactive Planning of Persistent Trajectories for Human-Assisted Navigation of Mobile Robots," in *2012 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2012, pp. 2641–2648.
- [7] K. Hauser, "Recognition, prediction, and planning for assisted teleoperation of freeform tasks," *Autonomous Robots*, vol. 35, no. 4, pp. 241–254, 2013.
- [8] H. Boessenkool, D. A. Abbink, C. J. Heemskerk, F. C. van der Helm, and J. G. Wildenbeest, "A task-specific analysis of the benefit of haptic shared control during telemanipulation," *Haptics, IEEE Transactions on*, vol. 6, no. 1, pp. 2–12, 2013.
- [9] C. Masone, P. Robuffo Giordano, H. H. Bühlhoff, and A. Franchi, "Semi-autonomous Trajectory Generation for Mobile Robots with Integral Haptic Shared Control," in *2014 IEEE Int. Conf. on Robotics and Automation*, 2014, pp. 6468–6475.
- [10] C. Passenberg, A. Peer, and M. Buss, "A survey of environment-, operator-, and task-adapted controllers for teleoperation systems," *Mechatronics*, vol. 20, no. 7, pp. 787–801, 2010.
- [11] T. Inagaki, "Adaptive automation: Sharing and trading of control," *Handbook of cognitive task design*, vol. 8, pp. 147–169, 2003.
- [12] A. D. Dragan and S. S. Srinivasa, "A policy-blending formalism for shared control," *The International Journal of Robotics Research*, vol. 32, no. 7, pp. 790–805, 2013.
- [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 Robotics and Automation Magazine*, vol. 12, no. 4, pp. 40–52, 2005.
- [14] F. Chaumette and S. Hutchinson, "Visual servo control, Part I: Basic approaches," *IEEE Robotics and Automation Magazine*, vol. 13, no. 4, pp. 82–90, 2006.
- [15] N. Guenard, T. Hamel, and R. Mahony, "A practical visual servo control for an unmanned aerial vehicle," *IEEE Trans. on Robotics*, vol. 2, no. 24, pp. 331–340, 2008.
- [16] R. Spica, P. Robuffo Giordano, and F. Chaumette, "Active Structure from Motion: Application to Point, Sphere and Cylinder," *IEEE Trans. on Robotics*, vol. 30, no. 6, pp. 1499–1513, 2014.
- [17] A. De Luca, G. Oriolo, and P. Robuffo Giordano, "Feature depth observation for image-based visual servoing: Theory and experiments," *International Journal of Robotics Research*, vol. 27, no. 10, pp. 1093–1116, 2008.
- [18] P. F. Hokayem and M. W. Spong, "Bilateral teleoperation: An historical survey," *Automatica*, vol. 42, no. 12, pp. 2035–2057, 2006.
- [19] E. Nuño, L. Basañez, and R. Ortega, "Passivity-based control for bilateral teleoperation: A tutorial," *Automatica*, vol. 47, no. 3, pp. 485–495, 2011.
- [20] D. J. Lee and K. Huang, "Passive-set-position-modulation framework for interactive robotic systems," *IEEE Trans. on Robotics*, vol. 26, no. 2, pp. 354–369, 2010.
- [21] A. Franchi, C. Masone, V. Grabe, M. Ryll, H. H. Bühlhoff, and P. Robuffo Giordano, "Modeling and Control of UAV Bearing-Formations with Bilateral High-Level Steering," *International Journal of Robotics Research*, vol. 12, no. 31, pp. 1504–1525, 2012.
- [22] D. Lee, A. Franchi, H. I. Son, H. H. Bühlhoff, and P. Robuffo Giordano, "Semi-Autonomous Haptic Teleoperation Control Architecture of Multiple Unmanned Aerial Vehicles," *IEEE/ASME Trans. on Mechatronics*, vol. 4, no. 18, pp. 1334–1345, 2013.