

A Complementary Framework for Human-Robot Collaboration with a Mixed AR-Haptic Interface

Xiangjie Yan, Yongpeng Jiang, Chen Chen, Leiliang Gong, Ming Ge, *Senior Member, IEEE*,
Tao Zhang, *Senior Member, IEEE*, and Xiang Li, *Member, IEEE*

Abstract—There is invariably a trade-off between safety and efficiency for collaborative robots (cobots) in human-robot collaborations. Robots that interact minimally with humans can work with high speed and accuracy but cannot adapt to new tasks or respond to unforeseen changes, whereas robots that work closely with humans can but only by becoming passive to humans, meaning that their main tasks suspended and efficiency compromised. Accordingly, this paper proposes a new complementary framework for human-robot collaboration that balances the safety of humans and the efficiency of robots. In this framework, the robot carries out given tasks using a vision-based adaptive controller, and the human expert collaborates with the robot in the null space. Such a decoupling drives the robot to deal with existing issues in task space (e.g., uncalibrated camera, limited field of view) and in null space (e.g., joint limits) by itself while allowing the expert to adjust the configuration of the robot body to respond to unforeseen changes (e.g., sudden invasion, change of environment) without affecting the robot's main task. Additionally, the robot can simultaneously learn the expert's demonstration in task space and null space beforehand with dynamic movement primitives (DMP). Therefore, an expert's knowledge and a robot's capability are both explored and complementary. Human demonstration and involvement are enabled via a mixed interaction interface, i.e., augmented reality (AR) and haptic devices. The stability of the closed-loop system is rigorously proved with Lyapunov methods. Experimental results in various scenarios are presented to illustrate the performance of the proposed method.

Index Terms—Collaborative robots, global adaptive control, null-space interaction, human demonstration.

I. INTRODUCTION

A cobot usually shares its workspace with humans, and thus directly (i.e., physically) or indirectly interacts with humans. It usually has two features: enhanced safety and ease of programming [1], so that it can operate near humans and be deployed flexibly on various tasks. The development of cobots in recent decades has increased production efficiency in manufacturing industries - which had previously reached a near-maximum level - to a new higher level.

X. Yan, Y. Jiang, C. Chen, T. Zhang, and X. Li are with the Department of Automation, Tsinghua University, Beijing, China (e-mail: yanxj20@mails.tsinghua.edu.cn; jyp19@mails.tsinghua.edu.cn; chen-che20@mails.tsinghua.edu.cn; taozhang@tsinghua.edu.cn; xiangli@tsinghua.edu.cn).

L. Gong and M. Ge are with Hong Kong Productivity Council (HKPC), Hong Kong, China. (e-mail: leiliang@hkpc.org; mingge@hkpc.org.)

This work was supported in part by the Science and Technology Innovation 2030-Key Project under grants 2021ZD0201404, in part by the National Natural Science Foundation of China under grants U21A20517 and 52075290, in part by Guangdong-Hong Kong-Macao Innovation Center, Guangzhou through Special Foundation for Applied Research under grants ITPRD-2021-279, and in part by Beijing National Research Center for Information Science and Technology under grants 20201880382. Corresponding author: Xiang Li (xiangli@tsinghua.edu.cn)

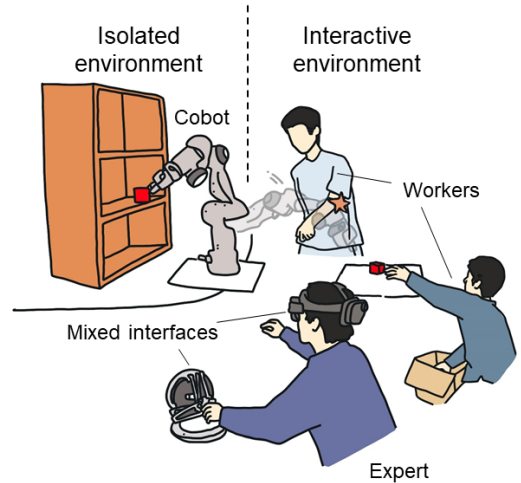


Fig. 1. An illustrative scenario for human-robot collaboration, where the robot needs to carry out tasks and transfer items in both an interactive environment and an isolated environment. An expert can also intervene or collaborate with the robot via the mixed AR-haptic interface.

The trade-off between safety and efficiency is always an open issue for cobots. To guarantee the safety of humans, cobots must typically suspend the ongoing task of end effector [2] and become passive to a human's control efforts, regardless of whether the human intervenes intentionally (i.e., is an expert who wishes to lead the task) or unintentionally. Not until the human ceases intervening can the robot continue its task. This kind of operational process may affect task efficiency because the robot needs to transit between different working modes.

To address the open issue, this paper proposes a new complementary framework for human-robot collaboration. Specifically, an illustrative scenario is considered in Fig. 1. First, a cobot grasps the target object in an interactive environment that also contains human workers. Then, it transfers the object to a desired position in an isolated setting, which excludes workers for reasons of safety or cleanliness. Such a scenario is commonly seen in factories, such as chemical factories [3], food-processing factories [4] and flat-panel-displays factories [5]. Cobots in these factories must often overcome one or several of the following challenges:

- human workers invading their workspace;
- an inexactly known environment, such that the relationship between their workspace and the sensory space is uncalibrated;
- joint angles that are subject to several limits (e.g., singu-

larity or constrained environment) and features that leave the field of view (FOV) during large displacements;

- the unavailability of some exact task information (i.e., desired position or reference trajectory) until the task begins.

This paper considers the aforementioned illustrative scenario and proposes a new framework for human-robot collaboration. The main novelty is its complementarity, which enables effective exploitation of the capabilities of a human expert (i.e., fast responses and smart decision-making) and those of a robot (i.e., high repetition and continuous working) and hence achieves a better balance between safety and efficiency. The contributions of this paper can be summarized as follows:

- 1) For a co-existing environment, a new vision-based adaptive controller is proposed to ensure a robot's global stability within the whole workspace; a null-space damping model is also formulated to allow a human expert to get involved at any time without affecting the main task.
- 2) For an isolated environment, a DMP-based planning scheme is developed to drive a robot to learn from expert demonstration and also migrate to new tasks; a mixed AR-haptic interface is also constructed such that a human expert can bi-manually demonstrate in both task space and redundant joint space.
- 3) The stability of closed-loop system in both task space and null space is rigorously proved, with consideration of transition between multiple regional feedback. In addition, experimental results in different scenarios are presented to validate the performance of the proposed method.

This work is an extension of our previous conference paper [6], and the improvement includes: 1) considering the whole pipeline of "grasping \rightarrow human interaction \rightarrow transferring \rightarrow obstacle avoidance \rightarrow placement"; 2) dealing with the joint limits and limited FOV during large displacements and developing a model-free method for online estimation of image Jacobian matrix; 3) learning skills from expert demonstration of both redundant joint and robot end effector; 4) building a mixed interface for more illustrative and intuitive interactions between human and robot to better complement each other; 5) carrying out more ablation studies and real-world experiments.

II. RELATED WORKS

This section reviews related works on cobot control and learning.

Task-Space Control: Task-space control directly specifies a feature or goal in task space, e.g., Cartesian space or vision space. This eliminates the need to solve an inverse kinematic problem, and thus task-space control has now become a standard method applied to robot manipulators. When a robot working in task space is subjected to a large displacement, its global stability is commonly limited by several open issues, i.e., joint limits, limited FOV, and uncalibrated camera.

First, a robot's joint angles may be subjected to several limitations due to singular configurations and constraints limits. Many studies have developed methods to keep the robot away

from these limits, e.g., by replanning the trajectory beforehand [7], exploring the kinematic redundancy [8], or damping the robot's motion when it is near the limit [9].

Second, the problem of limited FOV occurs when the visual feature leaves it during the task. In [10], a switching approach was proposed to switch the control input between a backward motion outside the FOV and a visual servoing method within it. In [11], a new weighted feature was proposed for vision-based control to allow some features to leave the FOV during manipulation. In [12], multiple visual features were kept within the FOV by regulating both the mean and also the variance of multiple features. The visual servoing scheme in [13] set the FOV as visibility constraints in the predefined performance bound, and the robot was controlled to achieve the desired transient response and hence stay within the FOV.

Third, the parameters of a camera deployed in a task-space control system may be unknown, for lacking prior calibration or being subject to adjustment (e.g., changes in focal length) when undertaking different tasks. In [14], Li *et al.* proposed a series of adaptive laws to estimate the parameters of uncalibrated cameras and robot dynamics concurrently. Without estimating the unknown camera parameters, [15] used three feature points distributed in a particular pattern, such that only pixel feedback from a fixed uncalibrated camera was needed to perform stabilization control for a nonholonomic mobile robot.

To address the aforementioned issues together, [16] proposed an adaptive task-space controller with the feedback switching among joint space, Cartesian space and vision space, to achieve the global stability within the whole workspace. Nevertheless, this and other existing task-space control schemes are commonly applicable to isolated environments, or their global stability is affected by the issues of joint limits, limited FOV, or uncalibrated sensors (e.g., [9]).

Human-Robot Collaboration: The scenario of human-robot collaboration (HRC) can be found in some manufacturing applications, where humans and robots perform a task together [1], [17]–[19].

As the robot co-exists with human, it is very important to guarantee the safety. The safety standard for HRC systems is defined in ISO 10218-1 and ISO/TS 15066, which are now used as guidelines for many real-world applications. In [20], an objective-switching method was adopted in an assembly task, which balanced the safety and time efficiency when the robot was approaching and avoiding the co-workers respectively. In [21], an optimization-based trajectory planning framework with iterative online safety module was proposed for HRC. A model recovering human-exerted forces was developed for dyadic cooperative object manipulation in [22], so that only human-applied force was measured to control the robot while the safety was guaranteed. However, most of the existing works have assumed that the perception of humans or obstacles is fully reliable, lacking the ability to deal with suddenly appearing or unforeseen changes.

Various HRC interfaces have also been developed for human involvement. In [23], EMG and IMU sensors were adopted to assess the human motion intention during physical

interaction. Face and gesture recognition were integrated in a collaborative system for assembly tasks [24]. Moreover, some fluency evaluation methods were proposed in [25]. Nevertheless, the aforementioned works are commonly limited to specific and predefined tasks. A general HRC interface in industrial setting is the teach pendant; However, it was found that this kind of interface has decreased the efficiency of and experience bained by humans [26]. Therefore, there is a demand for a human-oriented, intuitive, and general interface for HRC that facilitates convenient human involvement in robot-assisted tasks.

DMP for Robot Learning: Among various techniques of learning-from-demonstration (LfD), DMP has been proven to be an effective and efficient approach [27]–[29]. In DMP, movement is modelled using a spring-damper system, with the addition of a nonlinear forcing term to encode and modulate learning skills [27]. The DMP method has also been extended to Cartesian orientation [30], force adaptation [31], and arbitrary via-point adaptation [32] variations.

A typical LfD setting is to construct a teleoperation system, such that skills from the human expert (the master side) can be transferred to the robot (the slave side) and then adjusted via DMP according to the given task. As the master side is usually mechanically different from the slave side, several issues on how to record and reproduce the demonstration for better execution performance were raised [27], [28]. To better teach the robot via DMP, [33] developed a simulation environment where the human expert demonstrates a task using an AR device, and then transfers the demonstrated skills to the robot using highly transparent feedback. In another approach [34], a teleoperation control interface was developed for bilateral teleoperation, which consists of a three degrees-of-freedom (DOFs) HapticMaster robot and a stiffness control handle, which allows human-in-the-loop teaching and hence results in a better trajectory encoding.

In summary, DMP methods which mainly or solely focus on the task of the robot end effector have been developed in existing works. However, for a complex task such as that illustrated in Fig. 1, it is necessary to regulate both the robot end effector and the robot's body shape to suit the position of the collaborating human expert, avoid collisions, and so on.

III. PRELIMINARIES

This paper considers a cobot with redundant joints, whose forward kinematic model can be described as

$$\mathbf{r} = \mathbf{h}(\mathbf{q}), \quad (1)$$

where $\mathbf{r} \in \mathbb{R}^6$ denotes the position and the orientation of the robot end effector in Cartesian space, $\mathbf{q} \in \mathbb{R}^n$ is the vector of joint angles, $n > 6$ is the number of DOFs, and $\mathbf{h}(\cdot) \in \mathbb{R}^n \rightarrow \mathbb{R}^6$ is a nonlinear function.

Then, the velocity of the robot end effector in Cartesian space is related to the joint-space velocity as follows [35]:

$$\dot{\mathbf{r}} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}, \quad (2)$$

where $\mathbf{J}(\mathbf{q}) \in \mathbb{R}^{6 \times n}$ is the Jacobian matrix from joint space to Cartesian space.

The pseudo-inverse matrix is defined as

$$\mathbf{J}^+(\mathbf{q}) \triangleq \mathbf{J}^T(\mathbf{q})(\mathbf{J}(\mathbf{q})\mathbf{J}^T(\mathbf{q}))^{-1} \in \mathbb{R}^{n \times 6} \quad (3)$$

such that $\mathbf{J}(\mathbf{q})\mathbf{J}^+(\mathbf{q}) = \mathbf{I}_6$, where $\mathbf{I}_6 \in \mathbb{R}^{6 \times 6}$ is an identity matrix. Consequently, the null-space matrix can be introduced as follows [36]:

$$\mathbf{N}(\mathbf{q}) \triangleq \mathbf{I}_n - \mathbf{J}^+(\mathbf{q})\mathbf{J}(\mathbf{q}) \in \mathbb{R}^{n \times n}, \quad (4)$$

where $\mathbf{I}_n \in \mathbb{R}^{n \times n}$ represents an identity matrix. Equation (4) means that $\mathbf{J}(\mathbf{q})\mathbf{N}(\mathbf{q}) = \mathbf{0}$, $\mathbf{N}(\mathbf{q})\mathbf{J}^+(\mathbf{q}) = \mathbf{0}$, and $\mathbf{N}^2(\mathbf{q}) = \mathbf{N}(\mathbf{q})$, which implies that the null-space matrix $\mathbf{N}(\mathbf{q})$ is orthogonal to the Jacobian matrix $\mathbf{J}(\mathbf{q})$.

Because cobots are typically lightweight and operate at relatively low speeds, the control input can be specified at kinematic level, such that

$$\dot{\mathbf{q}} = \mathbf{u}, \quad (5)$$

where $\mathbf{u} \in \mathbb{R}^n$ denotes the control input corresponding to the joint-space velocity.

When a camera is used to measure the robot end effector in vision space, the feature's velocity in vision space is related to the end effector's velocity in Cartesian space [37], i.e.,

$$\dot{\mathbf{x}} = \mathbf{J}_s(\mathbf{r})\dot{\mathbf{r}}, \quad (6)$$

where \mathbf{x} denotes the feature's position (which is the position of the robot end effector) in vision space, and $\mathbf{J}_s(\mathbf{r}) \in \mathbb{R}^{2 \times 6}$ is the image Jacobian matrix. Due to the limited FOV, the visual feature is not available when it is initially outside the FOV or when it temporarily leaves the FOV during manipulation.

If the camera is not calibrated beforehand or if its parameters are adjusted to suit new tasks (e.g., camera autofocus, depth variation), the exact knowledge of the camera parameters may not be available, and hence the image Jacobian matrix is also unknown and is denoted as $\hat{\mathbf{J}}_s(\mathbf{r})$. An example of the vision-based cobot is illustrated in Fig. 2.

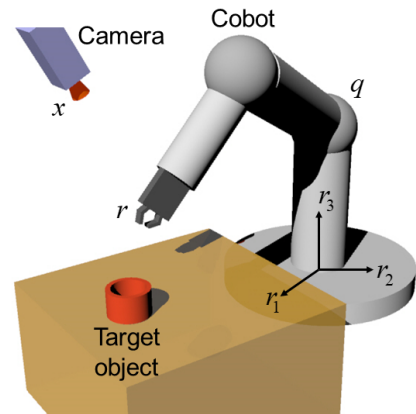


Fig. 2. A vision-based robot manipulator is controlled to grasp a target object, where $\mathbf{r} \in \mathbb{R}^6$ denotes the position and orientation of the robot end effector in Cartesian space, $\mathbf{q} \in \mathbb{R}^n$ is a vector of joint angles, and $\mathbf{x} \in \mathbb{R}^2$ represents the feature's position in vision space.

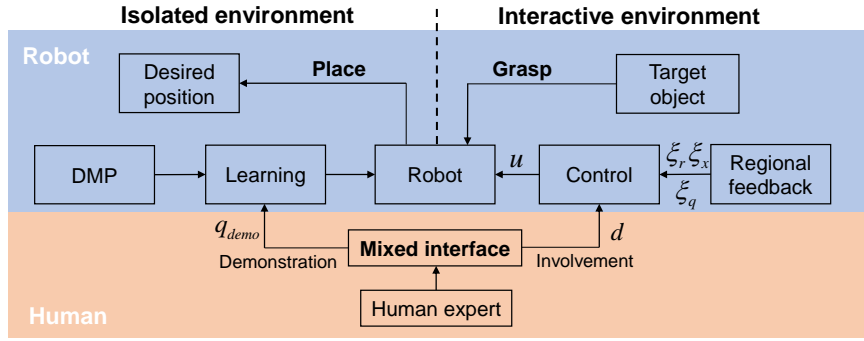


Fig. 3. The overall structure of complementary collaboration framework, where u is the control input, ξ_q, ξ_r, ξ_x are regional feedback vectors, d denotes the human control efforts, and q_{demo} is the demonstration trajectory in joint space. The robot carries out the task of “grasping - placing”, which involves the transition from an interactive environment (where the target object is located) to an isolated environment (where the desired position is located). The human expert can become involved to deal with unforeseen changes that occur during the grasping task, or to demonstrate a movement, which enables his/her expertise to be transferred to the robot before the placement task. Thus the ability of human and robot are complementary. The involvement and demonstration are achieved via a mixed interface, which is detailed in Section VII.

Problem Formulation: *The aim of this study is to design the control input (5) to guarantee the global stability of the robot and the convergence of task-space error to zero, in the presence of joint limits, uncalibrated camera, limited FOV and human involvement.*

IV. MULTIPLE REGIONAL FEEDBACK

This paper considers the scenario illustrated in Fig. 1. The overall structure of complementary collaboration under such a scenario is shown in Fig. 3. That is, the human expert interacts with the robot via the mixed interface. When the robot is controlled to grasp the target object in the co-existing environment, the human expert exerts control efforts in the null space to avoid potential collisions with workers. When the robot learns to place the object at the desired position, the human expert simultaneously demonstrates the reference trajectory in both Cartesian space and null space. Note that the proposed structure can also be extended to many other scenarios involving human-robot interaction.

This section presents the regional feedback [16] for the grasping operation, which is used to solve problems (e.g., joint limits and limited FOV) that may arise during large-displacement transfers. Thus, a series of regional feedbacks is formulated for the whole workspace, and the combination of regional feedback ensures the performance of robot in a global sense.

Joint-Space Feedback: The joint-space feedback is exploited to keep the robot away from the joint limits, which exist due to singularity or constrained workspace. Given that there are m limited configurations, the region function enclosing the i^{th} ($i=1, 2, \dots, m$) configuration is specified as

$$f_i(\mathbf{q}) \leq 0, \quad (7)$$

and the robot is away from this configuration when $f_i(\mathbf{q}) > 0$. For example, the joint-space regions for a 2-DOF planar robot [35] can be specified as: $f_1(\mathbf{q}) = q_2^2 - R_1^2 \leq 0$ and $f_2(\mathbf{q}) = R_2^2 - (q_2 - \pi)^2 \leq 0$, where R_1, R_2 specify the region size, and $q_2 = 0, q_2 = \pi$ are singular configurations.

Then, the potential energy function for the joint-space regions is proposed as

$$P_s(\mathbf{q}) = \sum_{i=1}^m \left\{ \frac{k_{qi}}{2} [\min(0, f_i(\mathbf{q}))]^2 + \frac{k_{ri}}{2} [\min(0, f_{ri}(\mathbf{q}))]^2 \right\}, \quad (8)$$

where k_{qi} and k_{ri} are positive constants, and $f_{ri}(\mathbf{q}) \leq 0$ is a reference region enclosing $f_i(\mathbf{q}) \leq 0$.

An illustration of the potential energy function is shown in Fig. 4. The first term in (8) is to create a high potential energy barrier, such that the robot does not have enough kinematic energy to approach the limited configurations. Hence, k_{qi} is set large to make the gradient of $P_s(\mathbf{q})$ steep (see Fig. 4). However, the steep gradient would cause oscillatory movement of the robot if it is very close to the region boundary $f_i(\mathbf{q}) = 0$. Hence, the second term in (8) is to decelerate the robot in advance and hence to alleviate the potential oscillation, where k_{ri} is relatively small.

Now, a regional feedback vector can be specified in joint space as

$$\xi_q \triangleq \frac{\partial P_s(\mathbf{q})}{\partial \mathbf{q}}, \quad (9)$$

which can be treated as a repulsive force to keep the robot away from the limited configurations. The vector automatically reduces to zero when the robot is outside the joint-space region.

Cartesian-Space Feedback: When the feature is not within the FOV, the Cartesian-space feedback is employed to drive the robot to move towards the feature, such that the feature can be seen inside the FOV. To match the rectangular FOV, a region is formulated in Cartesian space as

$$\mathbf{f}_c(\mathbf{r}) = \begin{bmatrix} f_{c1}(r_1) \\ f_{c2}(r_2) \\ f_{c3}(r_3) \end{bmatrix} = \begin{bmatrix} (r_1 - r_{e1})^2 - 1 \\ (r_2 - r_{e2})^2 - 1 \\ (r_3 - r_{e3})^2 - 1 \end{bmatrix} \leq \mathbf{0}, \quad (10)$$

where $\mathbf{r}_c = [r_{c1}, r_{c2}, r_{c3}]^T \in \mathfrak{R}^3$ denotes a reference position in the Cartesian-space position region (also within the FOV when it is projected to the vision space), and c_1, c_2, c_3 are positive constants. The region $\mathbf{f}_c(\mathbf{r}) \leq \mathbf{0}$ describes a cube in Cartesian space (see Fig. 5a), which matches the FOV in vision space.

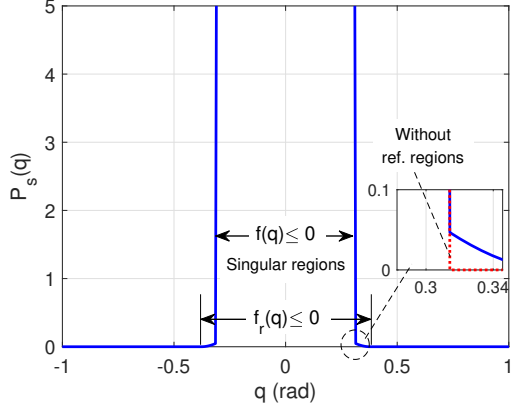


Fig. 4. An illustration of the potential energy function $P_s(q)$. If there is no reference region, the steep gradient of the potential energy may cause oscillatory movement of the robot (red dashed line).

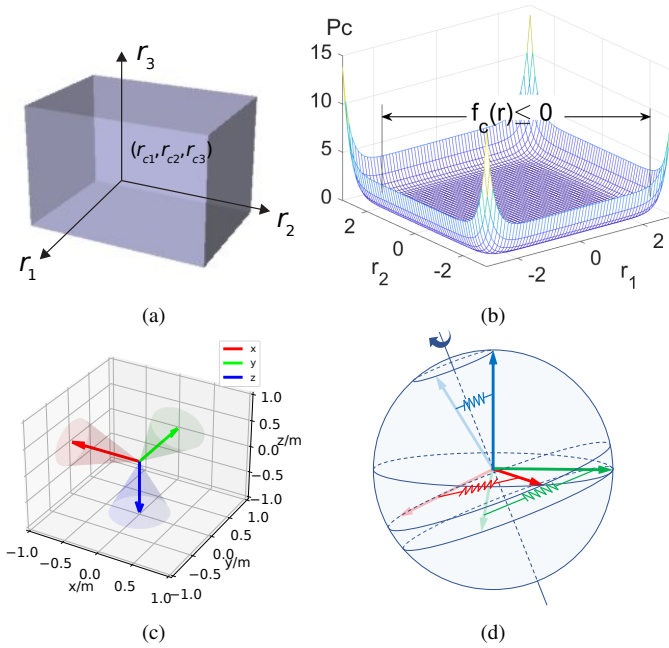


Fig. 5. An illustration of the Cartesian-space region and its potential energy. (a) The Cartesian-space position region is formulated as a rectangular block; (b) The potential energy function in 2D space; (c) The Cartesian-space orientation region is formulated as a “cone”; (d) The potential energy corresponding to the Cartesian orientation region function is intuitively regarded as being stored in the virtual spring system, which is only activated outside the region.

Note that the region is only used to regulate the position of robot end effector, such that the end effector is visible after it enters the Cartesian-space region.

Then, the corresponding potential energy function for the above region is proposed as

$$P_t(\mathbf{r}) = \sum_{i=1}^3 \left\{ \frac{k_{ci}}{2} [\max(0, f_{ci}(\mathbf{r}))]^2 \right\}, \quad (11)$$

where k_{ci} are positive constants. An illustration of the potential energy function is shown in Fig. 5b. From (11) and Fig. 5b, it can be seen that the potential energy drives the robot end effector to enter the region where $f_c(\mathbf{r}) \leq 0$ (which is also inside the FOV) and then reduces to zero.

However, the pose of the robot end effector may not be suitable for grasping if only the position is regulated. To address the problem, another Cartesian-space region is introduced to control the orientation of robot end effector, i.e.,

$$f_o(\mathbf{r}) = \alpha_o \|\log(\mathbf{p} * \mathbf{p}_g^{-1})\|_2 - 1 \leq 0, \quad (12)$$

where α_o is a positive constant which is related to the tolerance of orientation error, \mathbf{p}_g and \mathbf{p} are the quaternions representing the goal and the robot end effector, respectively, $(*)$ denotes the Hamilton product, and $\log(\cdot)$ describes the quaternion logarithm. The use of quaternions avoids the representation singularity. A simple example for the orientation region (12) can be given as: $\|\log(\mathbf{p} * \mathbf{p}_g^{-1})\|_2$, which describes the distance between \mathbf{p} and \mathbf{p}_g .

Similarly, the corresponding potential energy function is formulated as

$$P_o(\mathbf{r}) = \frac{1}{2} k_o [\max(0, f_o(\mathbf{r}))]^2, \quad (13)$$

where k_o is a positive scaling factor. The overall potential energy function in Cartesian space is the sum of $P_t(\mathbf{r})$ and $P_o(\mathbf{r})$, i.e.,

$$P_c(\mathbf{r}) = P_t(\mathbf{r}) + P_o(\mathbf{r}). \quad (14)$$

Next, the regional feedback vector is specified in Cartesian space as

$$\xi_r \triangleq \frac{\partial P_c(\mathbf{r})}{\partial \mathbf{r}} = \frac{\partial P_t(\mathbf{r})}{\partial \mathbf{r}} + \frac{\partial P_o(\mathbf{r})}{\partial \mathbf{r}}, \quad (15)$$

which can be treated as an attractive force that drives the robot end effector to enter the Cartesian-space regions, such that the end effector becomes visible and its orientation can be adjusted to a configuration suitable for grasping. The derivative of the potential energy function $\frac{\partial P_o(\mathbf{r})}{\partial \mathbf{r}}$ depends on whether analytical Jacobian or geometric Jacobian is applied in (27), which is detailed in the appendix. Note that the choice of Jacobian in the experiment had a slight impact on the control performance.

Vision Feedback: The vision feedback is employed by the robot end effector to grasp the target object. First, a region function is specified in vision space as

$$f_v(\mathbf{x}) = \left(\frac{x_1 - x_{d1}}{b_1} \right)^2 + \left(\frac{x_2 - x_{d2}}{b_2} \right)^2 - 1 \leq 0, \quad (16)$$

where $b_1, b_2 > 0$ are constants representing the half size of the FOV in the coordinates of x_1 and x_2 , respectively, and $\mathbf{x}_d = [x_{d1}, x_{d2}] \in \mathbb{R}^2$ is the desired position, which is also the position of the target object in vision space.

Accordingly, the potential energy function in vision space is introduced as

$$P_v(\mathbf{x}) = \frac{k_v}{2} \{1 - [\min(0, f_v(\mathbf{x}))]^2\}, \quad (17)$$

where k_v is a positive constant. The potential energy is shown in Fig. 6; its gradient is zero outside the region (i.e., $f_v(\mathbf{x}) > 0$) and non-zero inside the region, which drives the robot end effector to converge to the desired position for grasping.

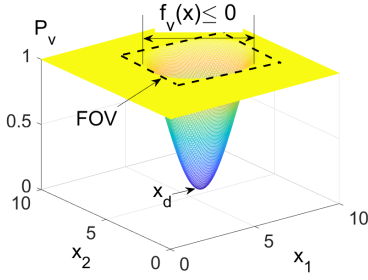


Fig. 6. An illustration of the potential energy function $P_v(\mathbf{x})$. Note that the vision-space region $f_v(\mathbf{x}) \leq 0$ is inside the FOV.

The regional feedback vector in vision space can now be specified in a similar way as

$$\xi_x \triangleq \frac{\partial P_v(\mathbf{x})}{\partial \mathbf{x}}, \quad (18)$$

which is activated inside the vision-space region where $f_v(\mathbf{x}) \leq 0$ (which is also inside the FOV). To ensure that the robot end effector can move from the Cartesian-space region to the vision-space region, the Cartesian-space region can be set smaller than the corresponding FOV, such that there is overlapping between each other.

V. VISION-BASED GLOBAL ADAPTIVE CONTROL

This section presents the global adaptive controller with multiple regional feedback, which drives the robot to interact with humans and grasp the target object in the presence of joint limits, limited FOV and uncalibrated cameras. Specifically, the control input is proposed as

$$\mathbf{u} = -\mathbf{J}^+(\mathbf{q})(\hat{\mathbf{J}}_s^T(\mathbf{r})\xi_x + \xi_r) + \mathbf{N}(\mathbf{q})c_d^{-1}(\mathbf{d} - \xi_q), \quad (19)$$

where c_d is a positive scalar and $\mathbf{d} \in \mathbb{R}^n$ denotes the control efforts exerted by the human expert on robot joints via mixed interfaces (e.g., Microsoft HoloLens 2 in [38]). The first term on the right side of (19) is to drive the end effector to carry out the main task in vision space, and the second term is regulate the redundant joints in null space to collaborate with the expert and also avoid joint limits, without affecting the main task. The objective of null-space control term can also be described as a damping model, i.e.,

$$\mathbf{N}(\mathbf{q})(c_d\dot{\mathbf{q}}) = \mathbf{N}(\mathbf{q})(\mathbf{d} - \xi_q). \quad (20)$$

where c_d can be considered as the desired damping parameter.

Next, the entry of the unknown image Jacobian transpose is approximated with adaptive neural network (NN) as

$$\hat{j}_s(\mathbf{r})_{i,j} = \hat{\mathbf{w}}_{i,j}^T \boldsymbol{\theta}(\mathbf{r}), \quad (21)$$

where $\hat{j}_s(\mathbf{r})_{i,j}$ is the (i,j) th entry of the matrix $\hat{\mathbf{J}}_s^T(\mathbf{r})$, $i = 1, 2, \dots, m, j = 1, 2$. $\hat{\mathbf{w}}_{i,j} \in \mathbb{R}^{n_k}$ is the corresponding weight, and $\boldsymbol{\theta}(\mathbf{r}) : \mathbb{R}^m \rightarrow \mathbb{R}^{n_k}$ is the nonlinear function of neurons. Radial basis function (RBF) is utilized as the neuron, where the i th entry is

$$\theta_i(\mathbf{r}) = \exp\left(-\frac{1}{2\sigma_i^2}\|\mathbf{r} - \mathbf{c}_i\|_2^2\right), \quad (22)$$

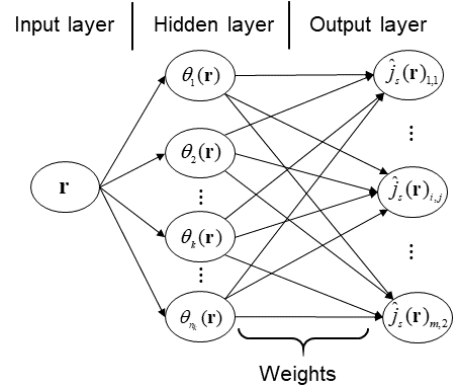


Fig. 7. The structure of the radial basis function neural network, where the input layer receives information on the position and orientation of the robot end effector (i.e., the vector of \mathbf{r}), the hidden layer consists of a series of RBF neurons (i.e., $\theta_1(\mathbf{r}), \theta_2(\mathbf{r}), \dots$), and the output layer generates the Jacobian matrix $\mathbf{J}_s(\mathbf{r})$.

where \mathbf{c}_i and $\sigma_i^2, i = 1, \dots, n_k$ are the centers and the variances, respectively. These parameters are manually predefined. The structure of the NN is shown in Fig. 7.

For simplicity, we rewrite (21) in the following vectorized form

$$\text{vec}(\hat{\mathbf{J}}_s^T(\mathbf{r})) = \hat{\mathbf{W}}\boldsymbol{\theta}(\mathbf{r}), \quad (23)$$

where $\hat{\mathbf{W}} = [\hat{\mathbf{w}}_{1,1}^T; \dots; \hat{\mathbf{w}}_{m,1}^T; \hat{\mathbf{w}}_{1,2}^T; \dots; \hat{\mathbf{w}}_{m,2}^T] \in \mathbb{R}^{2m \times n_k}$.

Thus, the weight of NN is updated with the following online adaptation law:

$$\dot{\hat{\mathbf{W}}} = -\left[\mathbf{L}\boldsymbol{\theta}(\mathbf{r})(\hat{\mathbf{J}}_s^T(\mathbf{r})\xi_x + \xi_r)^T \xi_x'\right]^T, \quad (24)$$

where $\mathbf{L} \in \mathbb{R}^{n_k \times n_k}$ is a positive-definite matrix, and ξ_x' is a matrix that reformulates the entries of $\xi_x = [\xi_{x1}, \xi_{x2}]^T$ as

$$\xi_x' = [\xi_{x1}\mathbf{I}_m, \xi_{x2}\mathbf{I}_m] \in \mathbb{R}^{m \times 2m}. \quad (25)$$

which has the following property:

$$\xi_x' \text{vec}(\hat{\mathbf{J}}_s^T(\mathbf{r})) = \hat{\mathbf{J}}_s^T(\mathbf{r})\xi_x. \quad (26)$$

The advantages of the proposed control scheme (19) are summarized as follows.

- When the robot nears the joint limits, the regional feedback vector ξ_q is activated to drive the robot away.
- The regional feedback vector ξ_r is used to drive the robot end effector to approach the desired position, such that both the feature and the desired position can be seen by the camera.
- The regional feedback vector ξ_x is activated only when both the feature and the target object are visible, such that the robot can grasp the target object in the presence of uncalibrated cameras.
- The online adaptation is driven by the regional feedback to deal with the unknown parameters concurrently.

By substituting (19) into (5), the closed-loop equation is obtained as

$$\dot{\mathbf{q}} = -\mathbf{J}^+(\mathbf{q})(\hat{\mathbf{J}}_s^T(\mathbf{r})\xi_x + \xi_r) + \mathbf{N}(\mathbf{q})c_d^{-1}(\mathbf{d} - \xi_q). \quad (27)$$

We are now in a position to state the following theorem.

Theorem: When the proposed vision-based adaptive control scheme described by (19) and (24) is applied to a collaborative robotic system, both the global stability of the closed-loop system and the convergence of task errors to zero are guaranteed, in the presence of joint limits, limited FOV, uncalibrated cameras and human's interaction.

Proof: See the appendix.

VI. ROBOT LEARNING FROM DEMONSTRATION

After the robot grasps the object, it transfers the object to the desired position in an isolated environment, as illustrated in Fig. 3. The desired position and the trajectory to this position are learnt from human's demonstration via the DMP approach. Such a proposed formulation has the following advantages.

- 1) Learning from demonstration can effectively exploit the expert's knowledge to set the desired position (e.g., a specific grasping pose) and suit the constrained space (e.g., path planning in the existence of the cabinet in Fig. 1);
- 2) DMP allows the learnt trajectory to be conveniently adjusted in response to the task generalization (e.g., modified goal positions on other shelves of the cabinet, or modified reaching speeds).

Basically, the DMP model for learning the trajectory of a single joint can be described as

$$\tau^2 \ddot{q} = \alpha_q [\beta_q (g - q) - \tau \dot{q}] + \zeta(z), \quad (28)$$

$$\tau \dot{z} = -\alpha_z z, \quad (29)$$

where (28) specifies a transformation system; (29) describes a canonical system; τ is a positive time constant, q , \dot{q} , \ddot{q} represent the angle, angular velocity and acceleration of the joint, respectively; $\alpha_q, \beta_q, \alpha_z$ are positive gain constants. In addition, $\zeta(z)$ is the forcing term, which is formulated as a linear combination of nonlinear basis functions, i.e.,

$$\zeta(z) = \frac{\sum_{i=1}^N \psi_i(z) \omega_i}{\sum_{i=1}^N \psi_i(z)} z (g - q_0), \quad (30)$$

where ω_i is the weight of the i th basis function, N is the total number of basis functions, q_0 is the initial position at $t = 0$, and $\psi_i(z)$ is the i th basis function. The latter is chosen as a radial basis function to allow discrete movement, i.e.,

$$\psi_i(z) = \exp \left[-\frac{1}{2\sigma_i^2} (z - c_i)^2 \right], \quad (31)$$

where c_i and σ_i^2 denote the center and the variance, respectively. The generation of a learnt trajectory consists of two steps: a *learning phase* and a *reproducing phase*.

Learning Phase: First, the human expert demonstrates a trajectory in terms of $\{q_{\text{demo}}(t), \dot{q}_{\text{demo}}(t), \ddot{q}_{\text{demo}}(t)\}_{t=0}^{\mathbb{T}}$. By referring to this demonstration, a desired forcing term can be calculated by transposing (28) as

$$\zeta_d = \tau^2 \ddot{q}_{\text{demo}} - \alpha_q [\beta_q (g - q_{\text{demo}}) - \tau \dot{q}_{\text{demo}}]. \quad (32)$$

Next, the locally weighted quadratic error is defined as the optimization target, i.e.,

$$\text{Cost}_i = \sum_{t=1}^{\mathbb{T}} \psi_i(z(t)) [\zeta_d(t) - \omega_i z(t)(g - q_0)]^2, \quad (33)$$

which forms a standard weighted linear regression problem, with the solution as

$$\omega_i = \frac{\mathbf{s}^T \Gamma_i \zeta_d}{\mathbf{s}^T \Gamma_i \mathbf{s}}, \quad (34)$$

where $i = 1, \dots, N$, and

$$\mathbf{s} = [z(1)(g - q_0), z(2)(g - q_0), \dots, z(\mathbb{T})(g - q_0)]^T, \quad (35)$$

$$\Gamma_i = \text{diag}(\psi_i(1), \psi_i(2), \dots, \psi_i(\mathbb{T})), \quad (36)$$

$$\zeta_d = [\zeta_d(1), \zeta_d(2), \dots, \zeta_d(\mathbb{T})]^T. \quad (37)$$

Reproducing Phase: After the weights in the forcing term are learnt, a new trajectory can now be generated by running (28) and (29). The learnt trajectory can be modulated according to the given scenario. Specifically, τ can be adjusted to speed up or slow down the trajectory execution, and g can be changed to set a new goal position while maintaining a similar transient movement to that position.

In this paper, the human expert demonstrates the trajectory in both the task space of the robot end effector and the null space of it in a bimanual and intuitive way via the mixed interface, as illustrated in Fig. 1 and Fig. 3. The demonstration teaches the robot to not only find the correct desired pose in task space but also shape the overall body to avoid collisions in a constrained environment.

VII. EXPERIMENT

Experiments were conducted on a vision-based robotic manipulation system to validate the proposed method, as shown in Fig. 8a. The overall system consisted of five modules: (i) a PC with Robot Operating System (ROS) and Ubuntu 18.04 LTS, in which the algorithm was implemented; (ii) a 7-DOF Franka robot with a two-fingered gripper, with ArUco markers attached to the gripper and the objects to aid perception (Fig. 8b); (iii) a Basler ace acA1440-220uc camera with 1440×1080 resolution, which was fixed in the workspace of the robot but not calibrated; (iv) an Omega 3, which is a haptic interface developed by Force Dimension; and (v) a HoloLens 2, which is a head-mounted AR device. Items (iv) and (v) comprised a mixed interface, which enabled the human expert to interact with the robot in both task space and null space in a bimanual way. The functions of the AR-haptic mixed interface are described in Fig. 9.

The AR device allowed the human expert to exert control efforts on the robot manipulator in two ways: by pulling the virtual robot closer and directly manipulating a specific joint (Fig. 8c) or by making the virtual robot overlap with the real robot and then using hand-ray and air-tap gestures to control the real robot remotely (Fig. 8d). These control efforts are represented as a vector (visualized in Fig. 8c and Fig. 8d) that is converted to a command velocity proportionally. Then, the control efforts \mathbf{d} are injected by projecting the velocity back to the joint space using the pseudo-inverse of the Jacobian matrix

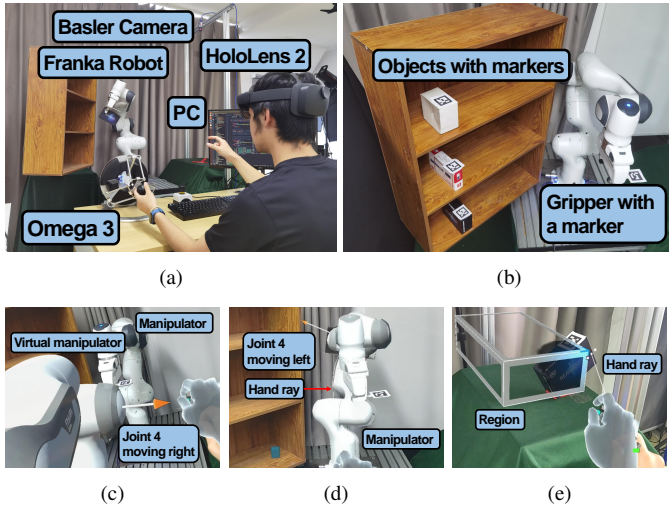


Fig. 8. The experimental setup: (a) The overall system consists of five modules: a haptic device, an AR device, a robot manipulator, a camera, and a PC. The expert is interacting with the robot in a bi-manual way. (b) The human expert performs the demonstration task in both the task space and the null space of the robot manipulator via the mixed interaction interface (i.e., the Omega 3 and HoloLens 2). (c) Direct dragging in the AR display. (d) Hand rays and air tapping in the AR display. (e) The human expert specifies the position and the size of the Cartesian-space region in the AR display.

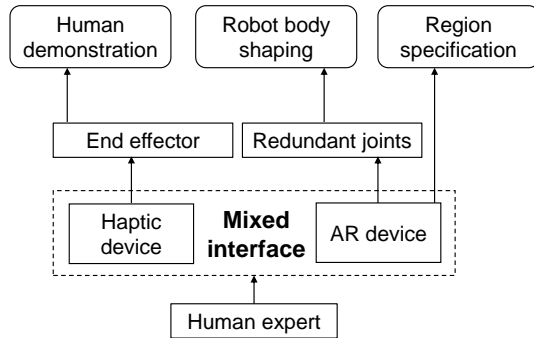


Fig. 9. The functions of the mixed interface, where the haptic device is used to control the robot end effector to follow human demonstration in task space, and the AR device is used to control redundant joints to follow the demonstration in joint space and also used to specify the Cartesian-space region.

of the selected joint. The method used to calculate d is the same as that used in [6]. In addition, the AR device allowed the human expert to conveniently specify the Cartesian-space region (Fig. 8e) by simply drawing a virtual region.

The following three experimental tasks were carried out to illustrate the performance of the proposed method. The purposes of experiments are detailed as follows.

- 1) *Placing Task* - to demonstrate how the cobot learned the desired trajectory in a complex unstructured environment.
- 2) *Grasping Task* - to validate the effectiveness of the global adaptive controller, in the presence of a large-scale transition, an uncalibrated camera, and joint limits.
- 3) *Collaboration Task* - to illustrate the entire pipeline, in which the robot transferred an object from an interactive environment (i.e., humans coexist) to an isolated environment. Such a task is commonly performed in many factories, e.g., during the transfer of hazardous chemicals.

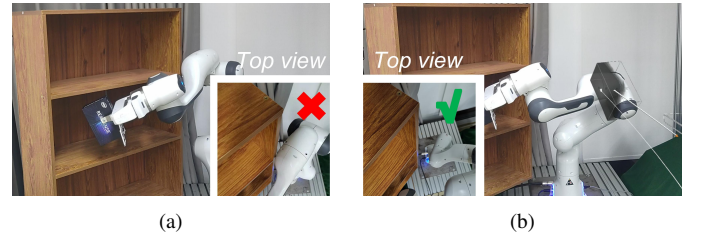


Fig. 10. Experiment 1 - Goal position inside the cabinet: (a) The 4th link collides with the cabinet due to inappropriate body shape; (b) The collision is avoided after the body shape is adjusted.

Note that the Franka robot accepts both torque input and velocity input. In this paper, the proposed control scheme is implemented at the kinematic level (i.e., the velocity input) as the robot moved at a relatively low velocity for human-robot collaboration; The control algorithm is developed based on Franka-interface and Frankap control stack [39].

A. Placing Task

In Experiment 1, the robot was already grasping an object and it was manipulated to place the object at a goal position on a shelf. The trajectory to the goal position was learnt from human demonstration via the DMP method and the mixed interface (i.e., Omega 3 and HoloLens 2).

During the demonstration, the translation of the robot end effector was defined by using Omega 3 and the orientation was defined simultaneously according to the translation as follows:

$$r_5 = \begin{cases} \frac{r_2}{Y_0} \times \beta_0, & |r_2| \leq Y_0, \\ -\beta_0, & r_2 < -Y_0, \\ \beta_0, & r_2 > Y_0, \end{cases} \quad (38a)$$

$$r_5 = \begin{cases} -\beta_0, & r_2 < -Y_0, \\ \beta_0, & r_2 > Y_0, \end{cases} \quad (38b)$$

$$r_5 = \begin{cases} \beta_0, & r_2 > Y_0, \end{cases} \quad (38c)$$

where $\mathbf{r} = [r_1, r_2, \dots, r_6]^T$ is the pose of the end effector in the base frame, $Y_0 = 0.6m$ and $\beta_0 = 90^\circ$ are the maximum reachable translation range in y-axis and the orientation range along the y-axis, respectively.

Note that the orientation defined in (38) is relatively coarse, which would be further adjusted with HoloLens 2; The demonstration via HoloLens 2 was thus used to define the motion of redundant joints in null space, not affecting the position of the robot end effector but shaping its orientation for fine-tuning.

The goal position on the shelf was shown in Fig. 10. It was insufficient to define only the position and orientation of the robot end effector, as an inappropriate body shape would result in the robot colliding with the cabinet (e.g., the collision of the 4th joint with the cabinet in Fig. 10). To solve this problem, the human expert used HoloLens 2 to define the motion of redundant joint and thus “pulled” the joint away from the cabinet. During the demonstration, the goal position was on the second shelf of the cabinet. The reproduction results shown in Fig. 11 indicated that the robot reached the goal position by following the learnt trajectory at a higher speed. The trajectories of the robot end effector during the demonstration and the reproduction in 3D space are plotted in Fig. 12, showing that the DMP-based learning method successfully captured the demonstration trajectory while removing several unnecessary jerks as well. The goal position of the learnt trajectory can also be set to positions on other shelves such

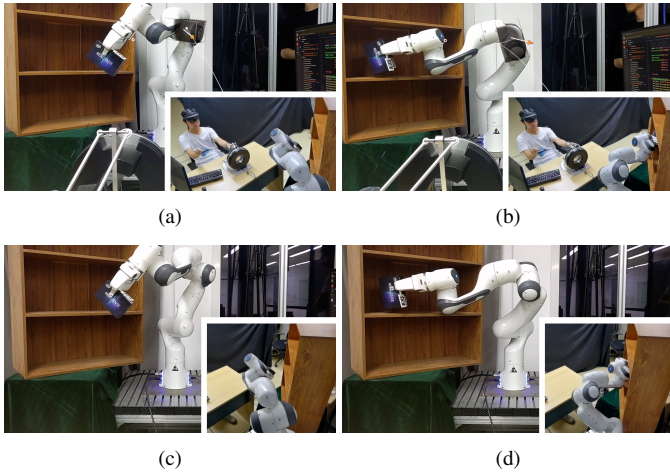


Fig. 11. Experiment 1 - Snapshots: the human expert performed the demonstration via the mixed interface, and the robot learnt and reproduced the trajectory. (a) *Learning phase* $t = 10.3\text{s}$: The human expert pulled the joint 4 of a virtual robot via HoloLens 2; (b) *learning phase* $t = 24.7\text{s}$: the real robot adjusted its body shape accordingly, without affecting its end effector; (c) *reproducing phase* $t = 6.8\text{s}$: the robot followed the learnt trajectory but at a higher velocity; (d) *reproducing phase* $t = 16.7\text{s}$: the robot placed the object at the goal position, without colliding with the cabinet.

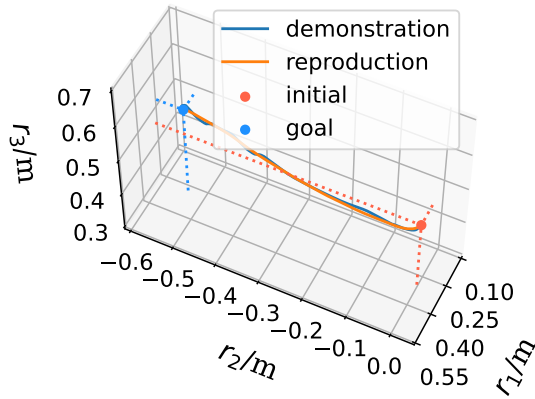


Fig. 12. Experiment 1 - Trajectories of the demonstration and reproduction in 3D space

that the grasped object is placed there accordingly. Note that the robot's movement was learnt and reproduced in joint space, where the corresponding joint angles were computed based on the analytical inverse kinematics method [40]. In addition, the angle of joint 7 was fixed as the redundant parameter, thus the reproduced trajectories with modified goals overlapped with each other (Fig. 13g).

From the joint motions illustrated in Fig. 13, the goal positions of the learnt trajectories were successfully changed to other ones (i.e., the positions on the lower or higher shelves). These new trajectories were implemented in the subsequent task (see Section VII.C) to place multiple objects at different shelves without colliding with the cabinet.

B. Grasping Task

In Experiment 2, the robot started from a remote initial position to grasp a target object, as shown in Fig. 14a. The proposed global adaptive controller defined by (19) and (24) was implemented to drive the robot to the object's position for

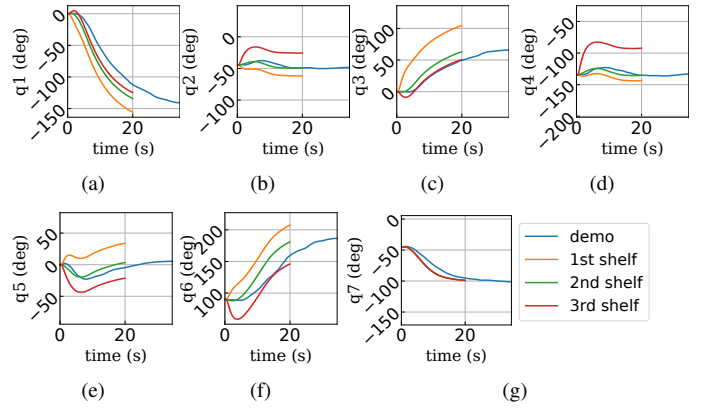


Fig. 13. Experiment 1 - Joint motions: the “demo” denotes the motion in the demonstration phase, and “1st, 2nd, 3rd” represent the motion for different goal positions in the reproducing phase. The goal positions were on three different shelves of the cabinet. The “2nd” shelf goal position was the same as the goal position in the demonstration. The trajectories were reproduced with $\tau = 1.5$, thus they terminated earlier than the demonstration, at $t = 20.0\text{s}$. (a) joint 1; (b) joint 2; (c) joint 3; (d) joint 4; (e) joint 5; (f) joint 6; (g) joint 7 (the reproduced trajectories overlapped with each other).

grasping, in the presence of joint limits, uncalibrated camera, and limited FOV.

Three regions were specified in the workspace of the robot: the joint-space region (7), the Cartesian-space region (10), (12), and the vision region (16). The joint-space region was introduced to prevent the robot entering joint limits, which could have resulted in an emergency stop. The vision region was specified to cover the position of the target object, such that the robot was able to employ the visual feedback for grasping. The Cartesian-space region was defined to dominate the remaining workspace of the robot, to ensure a smooth transition between different feedback; Specifically, the position region in Cartesian space was defined by the human expert via the AR interface (based on the approximate location of the FOV), and the quaternion region in Cartesian space was pre-defined. The combination of all the region feedback guaranteed the movement of the robot within the whole workspace.

The control parameters are listed in Table I. The initial image Jacobian $\hat{J}_s(\mathbf{r})_{t=0}$ was set randomly as in (39), and the initial weight $\hat{\mathbf{W}}_{t=t_0}$ was set according to $\hat{J}_s(\mathbf{r})_{t=0}$ by (40), where t_0 refers to the moment when the marker entered the FOV. In addition, the human expert did not exert additional control efforts on the robot in this experiment, and hence $\mathbf{d} = \mathbf{0}$.

$$\hat{J}_s(\mathbf{r})_{t=0} = \begin{bmatrix} -1500 & 0 & 0 & 0 & -200 & 60 \\ 0 & 2400 & 170 & -200 & 0 & 180 \end{bmatrix}, \quad (39)$$

$$\hat{\mathbf{W}}(i, j)_{t=t_0} = \frac{\text{vec}(\hat{J}_s(\mathbf{r})_{t=0})(i)}{\sum_{k=1}^{n_k} \theta(\mathbf{r}_{t=t_0})}, \quad (40)$$

$$i = 1, \dots, 2m; j = 1, \dots, n_k.$$

The experimental results are shown in Fig. 14. At the beginning, the human expert specified the size and the position of the Cartesian-space region (10) via the mixed interface, to ensure that it was within the FOV (see Fig. 14a); Then, the robot employed the Cartesian-space feedback to transit from

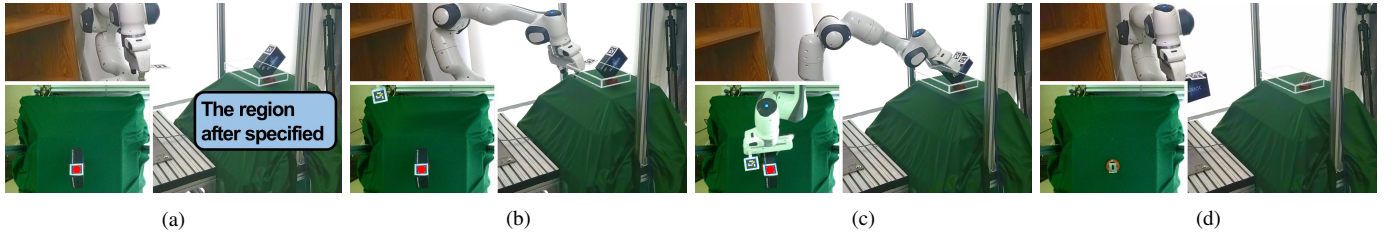


Fig. 14. Experiment 2 - Snapshots: The robot started from a remote initial position and moved to the target object to grasp it, in the presence of joint limits, uncalibrated camera, and limited FOV. The Cartesian-space region was defined by the expert via the AR interface and represented as a transparent cube with gray edges. (a) $t = 0.0$ s: the initial configuration; (b) $t = 5.8$ s: after the robot entered the Cartesian-space region, the marker appeared in the FOV; (c) $t = 9.2$ s: the robot employed the visual feedback to adjust its pose to aim at the target object; (d) $t = 15.0$ s: the robot grasped the object then returned to the home pose.

TABLE I
CONTROL PARAMETERS IN EXPERIMENT 2

Joint-Space Region (8)	k_q, k_r	10, 1
	f	$(q_i - q_{imin/max})^2 - 0.1^2 \leq 0$
	f_r	$(q_i - q_{imin/max})^2 - 0.3^2 \leq 0$
Position Region in Cartesian Space (10)(11)	r_c (center)	by expert
	k_{c1}, k_{c2}, k_{c3}	$[4e - 4, 4e - 4, 4e - 5]^T$
	c_1, c_2, c_3 (size)	by expert
Orientation Region in Cartesian Space (12)(13)	α_o	15
	p_g	$[-0.28, 0.63, 0.66, 0.28]^T$
	k_o	1
Vision Region (16)(17)	x_d (target object)	by detection
	b_1, b_2	$[1440, 1080]$
	k_v	0.3
Human Control Input (19)	c_d	3
	c_i	$[0.05, 0.35]^T + [0.15j, 0.15j]^T$ for $i, j = 0, 1, 2$
Adaptive NN (22),(23),(24)	σ	0.1
	L	$0.25I_9$
	$\hat{W}_{t=t_0}$	calculated according to $\hat{J}_s(\mathbf{r})_{t=0}$

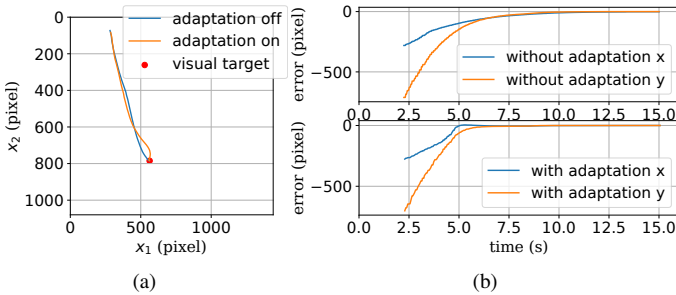


Fig. 15. Experiment 2 - Results: (a) The path of the robot end effector in vision space; (b) The position errors in vision space.

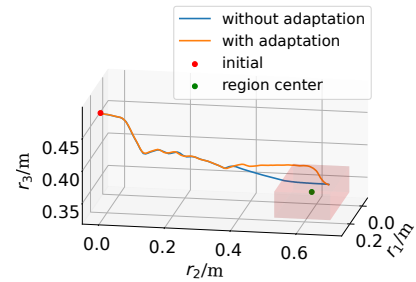


Fig. 16. Experiment 2 - The path of the robot end effector in 3D space after entering the FOV

outside to inside the FOV; Subsequently, the vision feedback became available (see Fig. 14b) and was used to drive the robot end effector to aim at the target object (see Fig. 14c); Finally, the robot grasped the object and moved back to its home position to complete the task (see Fig. 14d).

In this experiment, the camera was not calibrated beforehand and hence the exact information about the image Jacobian matrix was unknown. Thus, the adaptive NN was implemented to estimate the image Jacobian matrix via the online update law (24). The results with NN adaptation (where $L \neq 0$) or without NN adaptation (by setting $L = 0$) are

shown in Fig. 15. While both drove the robot to move to the desired position in vision space (see Fig. 15a), the control input with NN adaption achieved a faster convergence. The trajectory of the robot end effector in 3D space was shown in Fig. 16, proving the smooth transition of the robot among different regions.

C. Collaboration Task

In Experiment 3, the robot conducted both grasping and placing tasks. The robot transferred an object from an interactive environment to an isolated environment while also

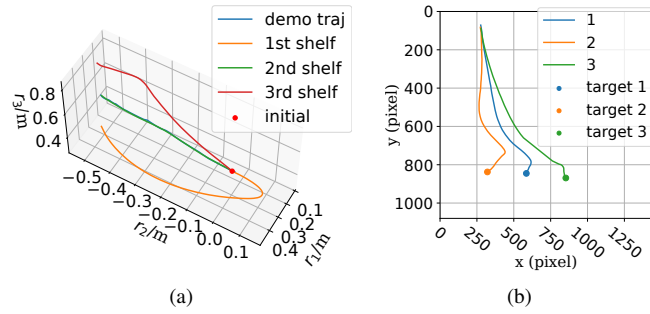


Fig. 17. Experiment 3 - Results: (a) The trajectories demonstrated by human experts and reproduced with DMP. (b) The path of the ArUco marker (i.e., the feature of target object) in vision space. The initial position was at $x_2 > 0$ as the marker was occluded at the beginning.

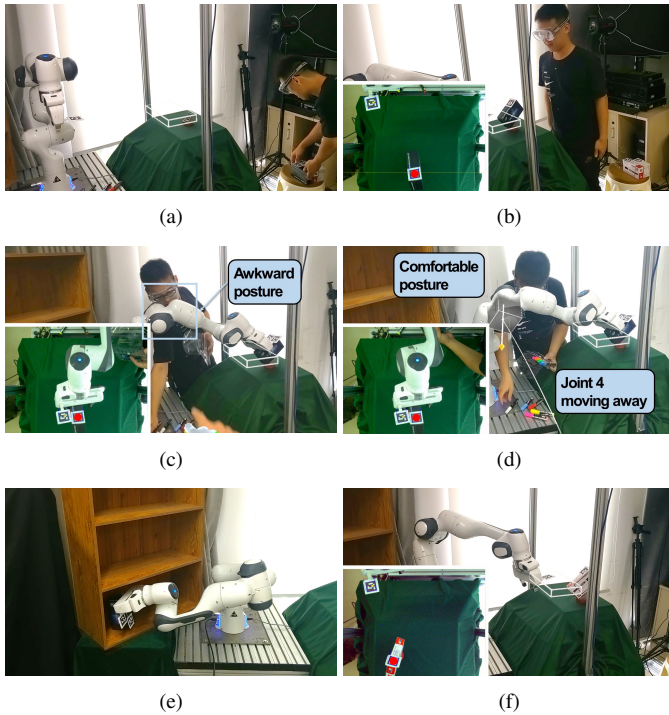


Fig. 18. Experiment 3 - Snapshots: (a) $t = 0.0$ s: the initial configuration; (b) $t = 3.4$ s: the worker placed the object on the workbench and the robot moved to grasp the object; (c) $t = 15.8$ s: the worker had to collect markers from the table in an awkward position due to obstruction by the robot; (d) $t = 26.3$ s: the expert used the mixed interface to drag the 4th joint of the robot away from the worker without affecting the main task of the robot end effector, enabling the worker to adopt a comfortable position; (e) $t = 68.4$ s: the robot transferred the object and placed it on a shelf; (f) $t = 80.1$ s: the worker placed another object at a different position on the workbench, and the robot moved to grasp and then place the object again.

collaborating with the human expert in the interactive environment. Such a mixed scenario is common in factories. For example, a worker hands over objects to a robot, which then transfers the objects to an unmanned laboratory where potentially hazardous experiments are conducted.

To fulfil the requirements, both the proposed learning scheme and the global adaptive controller were implemented in the robot and activated in different environment (i.e., the isolated and interactive environments, as illustrated in Fig. 1). The whole task of grasping-placing was performed three times in succession, i.e., three objects (with random initial positions within the FOV) were transferred to different shelves in the

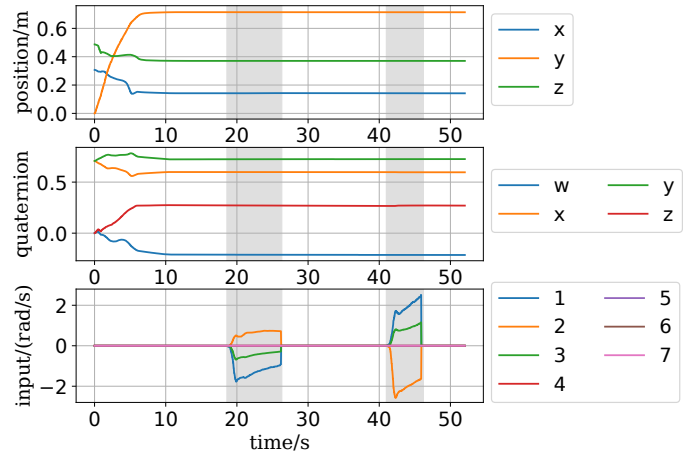


Fig. 19. Experiment 3 - The translation (top) and orientation (middle) of the robot end effector, and the human control efforts (bottom). The shaded area denotes the period when the human expert was exerting control efforts.

cabinet (see Fig. 8b).

For grasping task, the positions of target objects were detected in the vision space and then set as the desired positions for the controller, as shown in Fig. 17b. In addition, the desired orientation could be approximated by referring to the tangential direction of each trajectory in the neighborhood of the target object. Every time a new object was to be grasped, the image Jacobian matrix was varied but well estimated with the adaptive NN (24).

For placing task, the robot followed the learnt trajectories in Experiment 1. The trajectories had different goals corresponding to multiple desired positions on different shelves. The trajectories also consisted of similar paths towards the goal positions, which avoids collisions between the robot body and the shelves whenever a new object was being placed. The experimental results are shown in Fig. 17, which confirms that the consecutive grasping-placing was successfully realized.

Snapshots of the experiment are shown in Fig. 18. At $t = 0 \sim 10$ s, the robot moved to the desired pose for grasping. Meanwhile, the worker approached the robot and attempted to collect the scattered markers. As the markers were under the robot, the worker had to conduct the task in an awkward position. At $t = 19 \sim 26$ s and $t = 42 \sim 46$ s, the human expert observed the situations and dragged the 4th joint of the virtual robot forward and backward to adjust the body shape of the real robot, which guaranteed the worker's safety and comfort. The control efforts d exerted by the expert are shown in Fig. 19, which were projected into the null space; thus, they did not affect the main task (translation and orientation) of the robot end effector. Hence, the collaboration between the human expert and the robot was efficient. During the process, human expertise of the unstructured environment powerfully complemented the robot's large-scale transition ability, which was implemented with the vision-based global adaptive controller.

All the experimental results can be found at <https://youtu.be/JCZwo0fCbeg>. Specifically, the uploaded video also shows both the robot and the worker can perform tasks simultaneously in the shared space, without affecting each other.

VIII. CONCLUSIONS

This paper develops a new framework for human-robot collaboration, where the main novelty is its complementary feature. This enables a human expert and a robot to collaborate in a more efficient way. Specifically, a new vision-based adaptive controller is proposed for the robot to ensure the global convergence of the end effector, in the presence of joint limits, uncalibrated camera, and limited FOV; A mixed AR-haptic interface is developed to allow the expert to perform demonstration in both task space and redundant joint space and perform collaboration to deal with unforeseen changes (e.g., suddenly appearing human walkers), without affecting the main task. Therefore, the proposed framework enables the robot to safely interact with other co-existing workers, in parallel to its ongoing works, and it also provides a natural and intuitive way for the expert to deliver his/her knowledge and smart decision. The global stability of closed-loop system is rigorously proved with Lyapunov methods, and the performance of the proposed scheme is validated in a series of transferring tasks in the hybrid environment (i.e., interactive and isolated). Future works will be devoted to the marker-free perception and the field application in factories.

APPENDIX

A. Stability Analysis

Multiplying both sides of (27) by $\mathbf{N}(\mathbf{q})$ and noting that $\mathbf{N}(\mathbf{q})\mathbf{J}^+(\mathbf{q}) = \mathbf{0}$ and $\mathbf{N}^2(\mathbf{q}) = \mathbf{N}(\mathbf{q})$, we have

$$\mathbf{N}(\mathbf{q})\dot{\mathbf{q}} = \mathbf{N}(\mathbf{q})c_d^{-1}(\mathbf{d} - \boldsymbol{\xi}_q), \quad (41)$$

that is,

$$\mathbf{N}(\mathbf{q})(c_d\dot{\mathbf{q}} - \mathbf{d} - \boldsymbol{\xi}_q) = \mathbf{0}, \quad (42)$$

which maps the desired damping model $c_d\dot{\mathbf{q}} = \mathbf{d} - \boldsymbol{\xi}_q$ into the null space of the Jacobian matrix, such that both the expert's control efforts \mathbf{d} and the joint-space regional feedback $\boldsymbol{\xi}_q$ work without affecting the robot end effector.

Note that \mathbf{d} and $\boldsymbol{\xi}_q$ do not usually work at the same time:

- When the robot leaves the joint-space region and hence stays away from joint limits or singularity, $\boldsymbol{\xi}_q = \mathbf{0}$, (42) becomes $\mathbf{N}(\mathbf{q})(c_d\dot{\mathbf{q}}) = \mathbf{N}(\mathbf{q})\mathbf{d}$, such that the motion of redundant joints is solely determined by the expert.
- When the robot is inside the joint-space region and the expert does not input control efforts, $\mathbf{d} = \mathbf{0}$, (42) becomes $\mathbf{N}(\mathbf{q})(c_d\dot{\mathbf{q}}) = \mathbf{N}(\mathbf{q})\boldsymbol{\xi}_q$. Although it is possible that $\boldsymbol{\xi}_q \neq \mathbf{0}$ but $\mathbf{N}(\mathbf{q})\boldsymbol{\xi}_q = \mathbf{0}$, those cases are very rare; Because when the robot is not exactly located at the singular configuration, $\text{rank}(\mathbf{N}(\mathbf{q})) = 1$, $\mathbf{N}(\mathbf{q}) = \mathbf{a}\mathbf{b}^T$, where $\mathbf{a}, \mathbf{b} \in \mathbb{R}^7$ are vectors; That is, the aforementioned cases occur only when $\boldsymbol{\xi}_q$ is exactly orthogonal to \mathbf{b} , which is very rare in actual implementation; Hence, the robot will not stay inside the joint-space region (i.e., $\boldsymbol{\xi}_q \neq \mathbf{0}$ and $\dot{\mathbf{q}} = \mathbf{0}$) and will leave it by the end, i.e., $\boldsymbol{\xi}_q = \mathbf{0}$.

Hence, the control objective in null space is realized.

Multiplying both sides of (27) by $\mathbf{J}(\mathbf{q})$ and noting that $\mathbf{J}(\mathbf{q})\dot{\mathbf{q}} = \dot{\mathbf{r}}$ and $\mathbf{J}(\mathbf{q})\mathbf{N}(\mathbf{q}) = \mathbf{0}$, it is obtained that

$$\dot{\mathbf{r}} = -\hat{\mathbf{J}}_s^T(\mathbf{r})\boldsymbol{\xi}_x - \boldsymbol{\xi}_r. \quad (43)$$

To prove the stability of the closed-loop system in task space, a Lyapunov-like candidate is proposed as

$$V = P_v(\mathbf{x}) + P_c(\mathbf{r}) + \text{tr}(\tilde{\mathbf{W}}\mathbf{L}^{-1}\tilde{\mathbf{W}}^T), \quad (44)$$

where $\tilde{\mathbf{W}} = \mathbf{W} - \hat{\mathbf{W}}$ is the approximation error.

Differentiating (44) with respect to time yields

$$\begin{aligned} \dot{V} &= \dot{\mathbf{x}}^T \frac{\partial P_v(\mathbf{x})}{\partial \mathbf{x}} + \dot{\mathbf{r}}^T \frac{\partial P_c(\mathbf{r})}{\partial \mathbf{r}} - \text{tr}(\tilde{\mathbf{W}}\mathbf{L}^{-1}\dot{\tilde{\mathbf{W}}}^T) \\ &= \dot{\mathbf{x}}^T \boldsymbol{\xi}_x + \dot{\mathbf{r}}^T \boldsymbol{\xi}_r - \text{tr}(\tilde{\mathbf{W}}\mathbf{L}^{-1}\dot{\tilde{\mathbf{W}}}^T) \\ &= \dot{\mathbf{r}}^T (\mathbf{J}_s^T(\mathbf{r})\boldsymbol{\xi}_x + \boldsymbol{\xi}_r) - \text{tr}(\tilde{\mathbf{W}}\mathbf{L}^{-1}\dot{\tilde{\mathbf{W}}}^T). \end{aligned} \quad (45)$$

Substituting (43) into (45), it is obtained that

$$\begin{aligned} \dot{V} &= -(\hat{\mathbf{J}}_s^T(\mathbf{r})\boldsymbol{\xi}_x + \boldsymbol{\xi}_r)^T \times (\mathbf{J}_s^T(\mathbf{r})\boldsymbol{\xi}_x + \boldsymbol{\xi}_r) \\ &\quad - \text{tr}(\tilde{\mathbf{W}}\mathbf{L}^{-1}\dot{\tilde{\mathbf{W}}}^T) \\ &= -(\hat{\mathbf{J}}_s^T(\mathbf{r})\boldsymbol{\xi}_x + \boldsymbol{\xi}_r)^T \times ((\hat{\mathbf{J}}_s(\mathbf{r}) + \tilde{\mathbf{J}}_s(\mathbf{r}))^T \boldsymbol{\xi}_x + \boldsymbol{\xi}_r) \\ &\quad - \text{tr}(\tilde{\mathbf{W}}\mathbf{L}^{-1}\dot{\tilde{\mathbf{W}}}^T) \\ &= -(\hat{\mathbf{J}}_s^T(\mathbf{r})\boldsymbol{\xi}_x + \boldsymbol{\xi}_r)^T \times (\hat{\mathbf{J}}_s^T(\mathbf{r})\boldsymbol{\xi}_x + \boldsymbol{\xi}_r) \\ &\quad - (\hat{\mathbf{J}}_s^T(\mathbf{r})\boldsymbol{\xi}_x + \boldsymbol{\xi}_r)^T \tilde{\mathbf{J}}_s^T(\mathbf{r})\boldsymbol{\xi}_x - \text{tr}(\tilde{\mathbf{W}}\mathbf{L}^{-1}\dot{\tilde{\mathbf{W}}}^T), \end{aligned} \quad (46)$$

where $\tilde{\mathbf{J}}_s^T(\mathbf{r}) \triangleq \mathbf{J}_s^T(\mathbf{r}) - \hat{\mathbf{J}}_s^T(\mathbf{r})$. Making use of (23) and (26), it is clear that

$$\begin{aligned} & - (\hat{\mathbf{J}}_s^T(\mathbf{r})\boldsymbol{\xi}_x + \boldsymbol{\xi}_r)^T \tilde{\mathbf{J}}_s^T(\mathbf{r})\boldsymbol{\xi}_x \\ &= -\text{tr}[\tilde{\mathbf{J}}_s^T(\mathbf{r})\boldsymbol{\xi}_x(\hat{\mathbf{J}}_s^T(\mathbf{r})\boldsymbol{\xi}_x + \boldsymbol{\xi}_r)^T] \\ &= -\text{tr}[\boldsymbol{\xi}_x' \text{vec}(\tilde{\mathbf{J}}_s^T(\mathbf{r}))(\hat{\mathbf{J}}_s^T(\mathbf{r})\boldsymbol{\xi}_x + \boldsymbol{\xi}_r)^T] \\ &= -\text{tr}[\boldsymbol{\xi}_x' \tilde{\mathbf{W}}\boldsymbol{\theta}(\mathbf{r})(\hat{\mathbf{J}}_s^T(\mathbf{r})\boldsymbol{\xi}_x + \boldsymbol{\xi}_r)^T]. \end{aligned} \quad (47)$$

Substituting the update law (24) into the last term of (46), it is obtained that

$$\begin{aligned} & -\text{tr}(\tilde{\mathbf{W}}\mathbf{L}^{-1}\dot{\tilde{\mathbf{W}}}^T) \\ &= \text{tr}[\tilde{\mathbf{W}}\boldsymbol{\theta}(\mathbf{r})(\hat{\mathbf{J}}_s^T(\mathbf{r})\boldsymbol{\xi}_x + \boldsymbol{\xi}_r)^T \boldsymbol{\xi}_x'] \\ &= \text{tr}[\boldsymbol{\xi}_x' \tilde{\mathbf{W}}\boldsymbol{\theta}(\mathbf{r})(\hat{\mathbf{J}}_s^T(\mathbf{r})\boldsymbol{\xi}_x + \boldsymbol{\xi}_r)^T]. \end{aligned} \quad (48)$$

With (47) and (48), the last two terms in (46) can be cancelled such that

$$\dot{V} = -(\hat{\mathbf{J}}_s^T(\mathbf{r})\boldsymbol{\xi}_x + \boldsymbol{\xi}_r)^T \times (\hat{\mathbf{J}}_s^T(\mathbf{r})\boldsymbol{\xi}_x + \boldsymbol{\xi}_r) \leq 0. \quad (49)$$

Since $V > 0$ and $\dot{V} \leq 0$, V is bounded, and the closed-loop system is stable. The boundedness of V ensures the boundedness of $P_v(\mathbf{x})$, $P_c(\mathbf{r})$, and $\tilde{\mathbf{W}}$. Hence, all the regional feedback vectors $\boldsymbol{\xi}_x$, $\boldsymbol{\xi}_r$ are bounded. From (43), it can be seen that $\dot{\mathbf{r}}$ is bounded, which also ensures the boundedness of $\dot{\mathbf{x}}$ and $\dot{\mathbf{q}}$. Hence, the term $(\hat{\mathbf{J}}_s^T(\mathbf{r})\boldsymbol{\xi}_x + \boldsymbol{\xi}_r)$ is uniformly continuous. From (49), it follows that $(\hat{\mathbf{J}}_s^T(\mathbf{r})\boldsymbol{\xi}_x + \boldsymbol{\xi}_r) \in L_2(0, +\infty)$. Therefore, we have $(\hat{\mathbf{J}}_s^T(\mathbf{r})\boldsymbol{\xi}_x + \boldsymbol{\xi}_r) \rightarrow \mathbf{0}$.

When the regional feedback vector $\boldsymbol{\xi}_q$ keeps the robot away from the joint limits, $\boldsymbol{\xi}_q = \mathbf{0}$. Then, the regional feedback vector $\boldsymbol{\xi}_r$ drives the robot to move from outside to inside the FOV. After it enters the FOV, the regional feedback vector $\boldsymbol{\xi}_x$ is activated and it reduces to zero only when the robot has reached the desired position (to grasp the target object). Hence, the convergence of $(\hat{\mathbf{J}}_s^T(\mathbf{r})\boldsymbol{\xi}_x + \boldsymbol{\xi}_r) \rightarrow \mathbf{0}$ actually implies that the grasping task is realized, in the presence of limited FOV and uncalibrated camera.

B. Orientation Region

As the analytical Jacobian is derived by differentiating the forward kinematic equations, ξ_r can be obtained through pure differential operations. Representing the Cartesian configuration \mathbf{r} as $[\mathbf{r}_t^T \ \mathbf{r}_o^T]^T$, which consists of the translation part $\mathbf{r}_t = [x \ y \ z]^T$ and the orientation part $\mathbf{r}_o = [r_{ox} \ r_{oy} \ r_{oz}]^T = [\phi \hat{n}_x \ \phi \hat{n}_y \ \phi \hat{n}_z]^T = \phi \hat{\mathbf{n}}$. Note that \mathbf{r}_o is given in rotation vector form, which represents the axis of rotation $\hat{\mathbf{n}} = [\hat{n}_x \ \hat{n}_y \ \hat{n}_z]^T$ and the angle of rotation ϕ . Then, the regional feedback vector in Cartesian space from (15) is rewritten as

$$\xi_r \triangleq \begin{bmatrix} \frac{\partial P_t(\mathbf{r})}{\partial \mathbf{r}_t} \\ \frac{\partial P_o(\mathbf{r})}{\partial \mathbf{r}_o} \end{bmatrix}. \quad (50)$$

Applying the chain rule and we have

$$\frac{\partial P_o(\mathbf{r})}{\partial \mathbf{r}_o^T} = \frac{\partial P_o(\mathbf{r})}{\partial \mathbf{p}^T} \cdot \frac{\partial \mathbf{p}}{\partial \mathbf{r}_o^T}. \quad (51)$$

Assuming that the unit quaternions $\mathbf{p} = (v_o, \mathbf{u}_o)$, $\mathbf{p}_g^{-1} = (v_g, \mathbf{u}_g)$, and their product $\mathbf{p} * \mathbf{p}_g^{-1} = (v_e, \mathbf{u}_e)$ in the orientation region function (12). According to the quaternion product rule, we can compute the real part of the orientation error as

$$v_e = v_o \cdot v_g - \mathbf{u}_o^T \mathbf{u}_g. \quad (52)$$

Considering the constraint $v_o^2 + \|\mathbf{u}_o\|^2 \equiv 1$ for the unit quaternion $\mathbf{p} = (v_o, \mathbf{u}_o)$, where $\mathbf{u}_o = (u_{ox}, u_{oy}, u_{oz})$, the partial derivative of v_e with respect to \mathbf{p} is derived as

$$\begin{aligned} \frac{\partial v_e}{\partial v_o} &= v_g + v_o \left(\frac{u_{gx}}{u_{ox}} + \frac{u_{gy}}{u_{oy}} + \frac{u_{gz}}{u_{oz}} \right) \\ \frac{\partial v_e}{\partial u_{ox}} &= -u_{gx} + u_{ox} \left(-\frac{v_g}{v_o} + \frac{u_{gy}}{u_{oy}} + \frac{u_{gz}}{u_{oz}} \right) \\ \frac{\partial v_e}{\partial u_{oy}} &= -u_{gy} + u_{oy} \left(-\frac{v_g}{v_o} + \frac{u_{gx}}{u_{ox}} + \frac{u_{gz}}{u_{oz}} \right) \\ \frac{\partial v_e}{\partial u_{oz}} &= -u_{gz} + u_{oz} \left(-\frac{v_g}{v_o} + \frac{u_{gx}}{u_{ox}} + \frac{u_{gy}}{u_{oy}} \right). \end{aligned} \quad (53)$$

To obtain a more specific expression of (12), the unit quaternion logarithmic is defined as

$$\log(\mathbf{p} * \mathbf{p}_g^{-1}) = \begin{cases} \arccos(v_e) \frac{\mathbf{u}_e}{\|\mathbf{u}_e\|}, & \mathbf{u}_e \neq 0, \\ [0, 0, 0]^T, & \text{otherwise.} \end{cases} \quad (54a)$$

Substituting (54) into (12), it is obtained that

$$f_o(\mathbf{r}) = \alpha_o \cdot \arccos(v_e) \cdot \mathbb{I}_{\|\mathbf{u}_e\| > 0} - 1. \quad (55)$$

where $\mathbb{I}_{\|\mathbf{u}_e\| > 0}$ is the indicator function introduced to prevent dividing by zero in (54).

$$\mathbb{I}_{\|\mathbf{u}_e\| > 0} = \begin{cases} 1, & \|\mathbf{u}_e\| > 0, \\ 0, & \text{otherwise.} \end{cases} \quad (56a)$$

Computing the partial derivative of (55) with respect to quaternion \mathbf{p} and we have

$$\begin{aligned} \frac{\partial f_o(\mathbf{r})}{\partial \mathbf{p}^T} &= -\frac{\alpha_o}{\sqrt{1 - (v_e)^2}} \cdot \frac{\partial v_e}{\partial \mathbf{p}^T} \cdot \mathbb{I}_{\|\mathbf{u}_e\| > 0} \\ &= -\frac{\alpha_o}{\|\mathbf{u}_e\|} \cdot \frac{\partial v_e}{\partial \mathbf{p}^T} \cdot \mathbb{I}_{\|\mathbf{u}_e\| > 0}, \end{aligned} \quad (57)$$

where $\frac{\partial v_e}{\partial \mathbf{p}^T} = \left(\frac{\partial v_e}{\partial v_o}, \frac{\partial v_e}{\partial u_{ox}}, \frac{\partial v_e}{\partial u_{oy}}, \frac{\partial v_e}{\partial u_{oz}} \right)$ is derived from (53). Then according to (13), we have

$$\begin{aligned} \frac{\partial P_o(\mathbf{r})}{\partial \mathbf{p}^T} &= k_o \max(0, f_o(\mathbf{r})) \cdot \frac{\partial f_o(\mathbf{r})}{\partial \mathbf{p}^T} \\ &= -\frac{\alpha_o k_o}{\|\mathbf{u}_e\|} \cdot \max(0, f_o(\mathbf{r})) \cdot \frac{\partial v_e}{\partial \mathbf{p}^T} \cdot \mathbb{I}_{\|\mathbf{u}_e\| > 0}. \end{aligned} \quad (58)$$

The conversion from the unit quaternion to the rotation vector form which represents the same orientation is defined as

$$\mathbf{p} = \left(\underbrace{p_w}_{v_e}, \underbrace{(p_x, p_y, p_z)}_{\mathbf{u}_e} \right) = \left(\cos \frac{\phi}{2}, \sin \frac{\phi}{2} \cdot \underbrace{(\hat{n}_x, \hat{n}_y, \hat{n}_z)}_{\hat{\mathbf{n}}} \right), \quad (59)$$

where $\phi = \|\mathbf{r}_o\|$ and $\hat{\mathbf{n}} = \frac{\mathbf{r}_o}{\phi}$. By collecting first partial derivatives of the quaternion \mathbf{p} with respect to the rotation vector \mathbf{r}_o , we obtain the Jacobian matrix as $\mathbf{J}_{rot} = [\mathbf{J}_{rot,ij}] = \left[\frac{\partial p_i}{\partial r_{oj}} \right] \in \mathbb{R}^{4 \times 3}$. The Jacobian matrix \mathbf{J}_{rot} is defined as

$$\mathbf{J}_{rot} \triangleq \begin{bmatrix} A_x & A_y & A_z \\ B_{xyz} & C_{xy} & C_{xz} \\ C_{yx} & B_{yz} & C_{yz} \\ C_{zx} & C_{zy} & B_{zxy} \end{bmatrix} = \frac{\partial \mathbf{p}}{\partial \mathbf{r}_o^T}. \quad (60)$$

There are three types of elements in the matrix above, which are represented as

$$\begin{cases} A_i = -\frac{r_{oi}}{2\|\mathbf{r}_o\|} \sin \frac{\|\mathbf{r}_o\|}{2} \\ B_{ijk} = \frac{r_{oi}^2}{2\|\mathbf{r}_o\|^2} \cos \frac{\|\mathbf{r}_o\|}{2} + \frac{r_{oj}^2 + r_{ok}^2}{\|\mathbf{r}_o\|^3} \sin \frac{\|\mathbf{r}_o\|}{2}, \\ C_{ij} = \frac{r_{oi}r_{oj}}{2\|\mathbf{r}_o\|^2} \cos \frac{\|\mathbf{r}_o\|}{2} - \frac{r_{oi}r_{oj}}{\|\mathbf{r}_o\|^3} \sin \frac{\|\mathbf{r}_o\|}{2} \end{cases}, \quad (61)$$

where the subscripts are selected as follows

$$\begin{cases} A_i, & i \in \{x, y, z\} \\ B_{ijk}, & ijk \in \{xyz, yzx, zxy\} \\ C_{ij}, & ij \in \{xy, yx, xz, zx, yz, zy\} \end{cases}. \quad (62)$$

With (58), (60) and (61), we can obtain the regional feedback vector through (51). Note that if other orientation representations are chosen as \mathbf{r}_o , such as the Euler angles, ξ_r can be derived in a similar manner as (50) and (51).

Nevertheless, when the geometric Jacobian is used in lieu of the analytic Jacobian, the differential operations mentioned above become invalid. Referring to the translation part of the differential potential energy

$$\frac{\partial P_t(\mathbf{r})}{\partial \mathbf{r}_t} = 2\mathbf{k}_c \odot \max(0, f_c(\mathbf{r}_t)) \odot (\Delta \mathbf{r}_t \odot \mathbf{c}^2), \quad (63)$$

where $\Delta \mathbf{r}_t = \mathbf{r}_t - \mathbf{r}_c$, $\mathbf{c} = [c_1 \ c_2 \ c_3]^T$, and $\mathbf{k} = [k_{c1} \ k_{c2} \ k_{c3}]^T \in \mathbb{R}^3$ are column vectors, and \odot , \oslash represent the element-wise multiplication and division, respectively. We formally define the orientation part of differential potential energy as

$$\frac{\partial P_o(\mathbf{r})}{\partial \mathbf{r}_o} = \alpha_o k_o \max(0, f_o(\mathbf{r}_o)) \mathbf{r}_e. \quad (64)$$

In (64), \mathbf{r}_e denotes the rotation vector form of orientation error $\mathbf{p} * \mathbf{p}_g^{-1}$ and resembles the term $\Delta \mathbf{r}_t$ in (63), which are, respectively, equivalent to the angular and linear velocity required to align the desired frame with the end effector frame in unit time. The differences between (63) and (64) are as follows:

- The region function is a vector \mathbf{f}_c in (63), while it is a scalar f_o in (64). This is due to the coupling effect of orientation representation.
- In (64), the scaling factor $\alpha_o k_o$ is consistent for each dimension. However, in (63), the vectorized term $2\mathbf{k}_c \odot \mathbf{c}^2$ may lead to different convergence speeds in different dimensions of \mathbf{r}_t . Thus, \mathbf{k}_c must be chosen carefully.

Generally, calculating ξ_r through (64) is less computation-intensive than through (51). In addition, the introduction of geometric Jacobian prevents the occurrence of representation singularities.

REFERENCES

- [1] A. Weiss, A.-K. Wortmeier, and B. Kubicek, "Cobots in industry 4.0: A roadmap for future practice studies on human-robot collaboration," *IEEE Transactions on Human-Machine Systems*, vol. 51, no. 4, pp. 335–345, 2021.
- [2] X. Li, G. Chi, S. Vidas, and C. C. Cheah, "Human-guided robotic co-manipulation: Two illustrative scenarios," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 5, pp. 1751–1763, 2016.
- [3] A. Hentout, M. Aouache, A. Maoudj, and I. Akli, "Human-robot interaction in industrial collaborative robotics: a literature review of the decade 2008–2017," *Advanced Robotics*, vol. 33, no. 15-16, pp. 764–799, 2019.
- [4] J. Iqbal, Z. H. Khan, and A. Khalid, "Prospects of robotics in food industry," *Food Science and Technology*, vol. 37, pp. 159–165, 2017.
- [5] A. Sanderson, "Intelligent robotic recycling of flat panel displays," Master's thesis, University of Waterloo, 2019.
- [6] X. Yan, C. Chen, and X. Li, "Adaptive vision-based control of redundant robots with null-space interaction for human-robot collaboration," in *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2022.
- [7] S. Chivaverini, "Kinematically redundant manipulators," *Handbook of Robotics*, pp. 245–268, 2008.
- [8] C. Gosselin and L.-T. Schreiber, "Kinematically redundant spatial parallel mechanisms for singularity avoidance and large orientational workspace," *IEEE Transactions on Robotics*, vol. 32, no. 2, pp. 286–300, 2016.
- [9] M. G. Carmichael, D. Liu, and K. J. Waldron, "A framework for singularity-robust manipulator control during physical human-robot interaction," *The International Journal of Robotics Research*, vol. 36, no. 5-7, pp. 861–876, 2017.
- [10] G. Chesi, K. Hashimoto, D. Prattichizzo, and A. Vicino, "Keeping features in the field of view in eye-in-hand visual servoing: A switching approach," *IEEE Transactions on Robotics*, vol. 20, no. 5, pp. 908–914, 2004.
- [11] N. García-Aracil, E. Malis, R. Aracil-Santonja, and C. Pérez-Vidal, "Continuous visual servoing despite the changes of visibility in image features," *IEEE Transactions on Robotics*, vol. 21, no. 6, pp. 1214–1220, 2005.
- [12] N. R. Gans, G. Hu, K. Nagarajan, and W. E. Dixon, "Keeping multiple moving targets in the field of view of a mobile camera," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 822–828, 2011.
- [13] C. P. Bechlioulis, S. Heshmati-Alamdari, G. C. Karras, and K. J. Kyriakopoulos, "Robust image-based visual servoing with prescribed performance under field of view constraints," *IEEE Transactions on Robotics*, vol. 35, no. 4, pp. 1063–1070, 2019.
- [14] T. Li, J. Yu, Q. Qiu, and C. Zhao, "Hybrid uncalibrated visual servoing control of harvesting robots with rgb-d cameras," *IEEE Transactions on Industrial Electronics*, 2022.
- [15] X. Liang, H. Wang, Y.-H. Liu, Z. Liu, B. You, Z. Jing, and W. Chen, "Purely image-based pose stabilization of nonholonomic mobile robots with a truly uncalibrated overhead camera," *IEEE Transactions on Robotics*, vol. 36, no. 3, pp. 724–742, 2020.
- [16] X. Li and C. C. Cheah, "Global task-space adaptive control of robot," *Automatica*, vol. 49, no. 1, pp. 58–69, 2013.
- [17] S. Hjørth and D. Chrysostomou, "Human-robot collaboration in industrial environments: A literature review on non-destructive disassembly," *Robotics and Computer-Integrated Manufacturing*, vol. 73, p. 102208, 2022.
- [18] A. K. Inkulu, M. R. Bahubalendruni, A. Dara, and K. SankaranarayanaSamy, "Challenges and opportunities in human robot collaboration context of industry 4.0-a state of the art review," *Industrial Robot: the international journal of robotics research and application*, 2021.
- [19] A. C. Simões, A. Pinto, J. Santos, S. Pinheiro, and D. Romero, "Designing human-robot collaboration (hrc) workspaces in industrial settings: A systematic literature review," *Journal of Manufacturing Systems*, vol. 62, pp. 28–43, 2022.
- [20] A. Kanazawa, J. Kinugawa, and K. Kosuge, "Motion planning for human-robot collaboration using an objective-switching strategy," *IEEE Transactions on Human-Machine Systems*, vol. 51, no. 6, pp. 590–600, 2021.
- [21] A. Palleschi, M. Hamad, S. Abdolshah, M. Garabini, S. Haddadin, and L. Pallottino, "Fast and safe trajectory planning: Solving the cobot performance/safety trade-off in human-robot shared environments," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5445–5452, 2021.
- [22] E. Noohi, M. Žefran, and J. L. Patton, "A model for human-human collaborative object manipulation and its application to human-robot interaction," *IEEE transactions on robotics*, vol. 32, no. 4, pp. 880–896, 2016.
- [23] W. Wang, R. Li, Z. M. Diekel, Y. Chen, Z. Zhang, and Y. Jia, "Controlling object hand-over in human-robot collaboration via natural wearable sensing," *IEEE Transactions on Human-Machine Systems*, vol. 49, no. 1, pp. 59–71, 2018.
- [24] I. El Makrini, K. Merckaert, D. Lefeber, and B. Vanderborght, "Design of a collaborative architecture for human-robot assembly tasks," in *IROS*, pp. 1624–1629, 2017.
- [25] G. Hoffman, "Evaluating fluency in human-robot collaboration," *IEEE Transactions on Human-Machine Systems*, vol. 49, no. 3, pp. 209–218, 2019.
- [26] R. Bogue, "Europe continues to lead the way in the collaborative robot business," *Industrial Robot: An International Journal*, 2016.
- [27] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and autonomous systems*, vol. 57, no. 5, pp. 469–483, 2009.
- [28] H. Ravichandar, A. S. Polydoros, S. Chernova, and A. Billard, "Recent advances in robot learning from demonstration," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, pp. 297–330, 2020.
- [29] M. Saveriano, F. J. Abu-Dakka, A. Kramberger, and L. Peternel, "Dynamic movement primitives in robotics: A tutorial survey," *arXiv preprint arXiv:2102.03861*, 2021.
- [30] A. Ude, B. Nemeč, T. Petrič, and J. Morimoto, "Orientation in cartesian space dynamic movement primitives," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2997–3004, IEEE, 2014.
- [31] A. Gams, B. Nemeč, A. J. Ijspeert, and A. Ude, "Coupling movement primitives: Interaction with the environment and bimanual tasks," *IEEE Transactions on Robotics*, vol. 30, no. 4, pp. 816–830, 2014.
- [32] Y. Zhou, J. Gao, and T. Asfour, "Learning via-point movement primitives with inter- and extrapolation capabilities," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4301–4308, IEEE, 2019.
- [33] H. Beik-Mohammadi, M. Kerzel, B. Pleintinger, T. Hulin, P. Reisich, A. Schmidt, A. Pereira, S. Werlter, and N. Y. Lii, "Model mediated teleoperation with a hand-arm exoskeleton in long time delays using reinforcement learning," in *2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pp. 713–720, IEEE, 2020.
- [34] L. Peternel, T. Petrič, and J. Babič, "Robotic assembly solution by human-in-the-loop teaching method based on real-time stiffness modulation," *Autonomous Robots*, vol. 42, no. 1, pp. 1–17, 2018.
- [35] C. C. Cheah and X. Li, *Task-space sensory feedback control of robot manipulators*, vol. 73. Springer, 2015.
- [36] H. Sadeghian, L. Villani, M. Keshmiri, and B. Siciliano, "Task-space control of robot manipulators with null-space compliance," *IEEE Transactions on Robotics*, vol. 30, no. 2, pp. 493–506, 2013.
- [37] J.-J. E. Slotine, W. Li, et al., *Applied nonlinear control*, vol. 199. Prentice hall Englewood Cliffs, NJ, 1991.
- [38] Microsoft, "Microsoft hololens2." <https://www.microsoft.com/en-us/hololens/hardware>, 2020. Accessed: 2022-06-01.

- [39] K. Zhang, M. Sharma, J. Liang, and O. Kroemer, "A modular robotic arm control stack for research: Franka-interface and frankapy," *arXiv preprint arXiv:2011.02398*, 2020.
- [40] Y. He and S. Liu, "Analytical inverse kinematics for Franka Emika Panda – a geometrical solver for 7-DOF manipulators with unconventional design," in *2021 9th International Conference on Control, Mechatronics and Automation (ICCMA2021)*, IEEE, Nov. 2021.