

Contents lists available at [ScienceDirect](https://www.sciencedirect.com)

Journal of Manufacturing Systems

journal homepage: www.elsevier.com/locate/jmansys

Technical Paper



Advanced teleoperation and control system for industrial robots based on augmented virtuality and haptic feedback

Claudia González^a, J. Ernesto Solanes^{a,*}, Adolfo Muñoz^a, Luis Gracia^a, Vicent Girbés-Juan^b, Josep Tornero^a^a Instituto de Diseño y Fabricación, Universitat Politècnica de València, Camino de Vera s/n, 46022 Valencia, Spain^b Departament d'Enginyeria Electrònica, Universitat de València, Avda de la Universitat s/n, 46100 Burjassot, Spain

ARTICLE INFO

Keywords:

Augmented virtuality
Virtual reality
Haptic feedback
Polishing task

ABSTRACT

There are some industrial tasks that are still mainly performed manually by human workers due to their complexity, which is the case of surface treatment operations (such as sanding, deburring, finishing, grinding, polishing, etc.) used to repair defects. This work develops an advanced teleoperation and control system for industrial robots in order to assist the human operator to perform the mentioned tasks. On the one hand, the controlled robotic system provides strength and accuracy, holding the tool, keeping the right tool orientation and guaranteeing a smooth approach to the workpiece. On the other hand, the advanced teleoperation provides security and comfort to the user when performing the task. In particular, the proposed teleoperation uses augmented virtuality (i.e., a virtual world that includes non-modeled real-world data) and haptic feedback to provide the user an immersive virtual experience when remotely teleoperating the tool of the robot system to treat arbitrary regions of the workpiece surface. The method is illustrated with a car body surface treatment operation, although it can be easily extended to other surface treatment applications or even to other industrial tasks where the human operator may benefit from robotic assistance. The effectiveness of the proposed approach is shown with several experiments using a 6R robotic arm. Moreover, a comparison of the performance obtained manually by an expert and that obtained with the proposed method has also been conducted in order to show the suitability of the proposed approach.

1. Introduction

1.1. Motivation

Surface treatment tasks used to repair defects are complex processes. For instance, in the automotive industry, surface finishing requires the dexterity of human hands and an intelligent use of feedback, which human experts obtain by touching and looking at the defects on the workpiece surface during the treatment [1]. In addition, the procedure required to repair a defect may also vary during its repair due to the condition of the surface (e.g., defects on a car body surface recently painted [2]).

Although some automatic systems can be found in the literature dealing with surface treatment tasks in the automotive industry [3,4], in most of the factories surface treatment tasks to repair car body surface defects are still carried out manually by experts. The main reason for this is the difficulty of automating this process if the finishing tool does not

perform consistently or if there is no possibility to check the result [1].

However, manual operation gives rise to some issues, such as the expert criteria variability or the exclusion of workers with motor disabilities (e.g., those requiring wheelchairs). In addition, current solutions fail to meet the demanding time requirements of automotive production lines [5].

In order to mitigate the above issues, this work presents a novel solution based on a *remote* system for robot teleoperation [6] and human–robot interaction (HRI) [7,8]. On the one hand, the remote robot teleoperation will allow human workers to perform their tasks without being physically present on the production line. On the other hand, the remote HRI will allow to cooperatively solve the task at hand in synergy, i.e., the human user provides flexibility to adapt to complex operations, whereas the robot system provides precision and strength.

* Corresponding author.

E-mail address: esolanes@idf.upv.es (J.E. Solanes).<https://doi.org/10.1016/j.jmsy.2021.02.013>

Received 9 April 2020; Received in revised form 22 February 2021; Accepted 23 February 2021

Available online 13 March 2021

0278-6125/© 2021 The Society of Manufacturing Engineers. Published by Elsevier Ltd. All rights reserved.

1.2. Literature review

HRI has been a subject of numerous studies due to the possibilities that it offers to fulfill tasks and accomplish objectives that would not be possible only with autonomous machines. This type of collaboration has been widely implemented by using haptic devices. For instance, in [9] a mobile robot is commanded using a joystick with haptic feedback. A similar system is deployed in [10,11] to teleoperate different types of mobile robots. A multiple robot team interaction is developed in [12] using wearable haptic devices. In [13,14] robot arm interaction is used off-line for path planning and task learning, respectively.

The current emergence of mixed reality (MR) technology opens a new scenario for the improvement of many industrial processes [15]. In particular, MR has been used to improve all kinds of industrial processes, from maintenance services [16–19] to industrial robot programming [20]. MR fills the gap between reality and virtuality, where it can be considered two subcategories [21]: augmented reality (AR), i.e., merging virtual objects into the real world [22–24]; and augmented virtuality (AV), i.e., merging real-world objects into virtual worlds [25].

Teleoperation based on AV has been proposed by many researchers to give rise to feasible solutions to complex industrial applications. For instance, a robust adaptive control algorithm was proposed in [26] to deal with system uncertainties and to provide a smooth estimation of delayed reference signals in robot teleoperation applications. In [27], a remotely controlled welding scheme was developed to transfer human welder knowledge to a welding robot. A teleoperation scheme was proposed in [28] to perform inspection and maintenance tasks in oil and gas well-pads stations. In [29], a telemanipulation application was developed for surface taping. Furthermore, a teleoperation scheme was proposed in [30] to remotely command industrial robots for hazardous manufacturing environments. In particular, a collaborative robot was used by the worker to remotely teleoperate an industrial robot using the master-slave technique, while a model-driven display system based on cameras and an AR headset was also used by the worker to be aware of the production context in real-time.

For most industrial teleoperation tasks, the visual feedback needs to be accompanied by touch feedback. For example, this is the case of surface treatment tasks such as polishing or cleaning. Thus, haptic devices are commonly used for this purpose. For instance, a teleoperation interface based on AV and haptic feedback was developed in [31] for hole-cleaning workcells using robots. Moreover, a novel dual-arm teleoperation architecture with haptic and visual feedback was proposed in [32] to enhance the operator immersion in surface treatment tasks.

1.3. Proposal

Although HRI and MR have been widely explored and studied, as discussed above, this work presents a novel advanced teleoperation and control system for industrial robots that allows to get the best from humans and robots. That is, the flexibility of humans to adapt to complex operations and unforeseen situations, such as workpiece position errors (obviously, the responding time to adapt to these unforeseen situations depends on the worker's ability), and the precision and strength of robots. In addition, the teleoperation system developed in this work, which is based on a novel AV interface and haptic feedback, provides the user an immersive virtual experience when remotely commanding the robot system in order to properly perform the task with security and comfort.

With the proposed approach, the user is able to command the robot from a different workspace without missing information of the task environment. All the elements of the robot workplace are modeled in a VR environment, which allows the user to see the movement of the robot in real time in order to check how the robot responds to the user commands. Moreover, the haptic feedback allows the user to feel the robot dynamics and its interaction with the environment.

Moreover, for safety, non-modeled real-world data obtained with a

3D camera is included in the VR environment. This allows the user to visually verify if this data matches the virtual objects and, if that is not the case, the user may decide, for security reasons, to abort the robot teleoperation. For further safety, a force sensor mounted on the robot end-effector is also used to abort the robot operation when the measured values are abnormally high in order to protect both the robot and its environment from damage.

All these functionalities allow the user to have a full immersive experience whilst receiving information regarding the robot and its workspace.

Without loss of generality, a surface treatment operation is considered to illustrate the proposed approach, where the human operator and the robot system cooperatively perform the treatment on the surface of a known workpiece. In particular, the operator is responsible for remotely teleoperating the tool position, whereas the robot automatically ensures not only the perpendicularity between the workpiece surface and the robot tool but also a smooth approach to the workpiece. For this purpose, a high-level controller is developed to control the robot system, which includes three types of controllers within a task prioritization strategy.

1.4. Content of the article

Section 2 gives some theoretical basis used in this research. Then, Section 3 presents the proposed application, whereas the high-level controller of the robot is detailed in Section 4. Next, the feasibility of the proposal is proved in Section 5 with several experiments. Finally, Section 7 presents the conclusions of this work.

2. Theoretical basis

This section presents the theoretical basis used to develop the proposed approach in Sections 3 and 4. This background theory is related to the computer vision system, the robot kinematics, the task prioritization method and a non-conventional sliding mode control.

2.1. Computer vision system

Computer vision is widely used in industrial robot tasks since it provides flexibility and precision. The camera can be placed in the end-effector of the robot system (eye-in-hand configuration), e.g., see [33], or, alternatively, it can be placed in a structure to “observe” the whole workspace of the robot system (eye-to-hand configuration), e.g., see [34].

If a 2D camera is used, the depth can be estimated by processing the acquired data [35]. Alternatively, this estimation is not needed if a 3D camera is used [36], e.g., the Microsoft Kinect.

Moreover, a camera network can be used to have a complete view of the scene and to avoid the so-called dead zones or blind spots. For this purpose, the number and placement of the cameras have to be properly chosen. For instance, a systematic camera placement framework was proposed in [37] to overcome the problem of blind spots using the visual performance of the overall network (index of the total workspace covered) and the number of cameras used (index of the total cost of the solution).

In particular, a network of three 3D sensors (Kinect cameras) with eye-to-hand configuration is used in this work.

The process of “registration” is needed to obtain the transformation relating two views of the same workpiece [38]. This process is useful to obtain the position and orientation (i.e., the pose) of a workpiece by matching the *point cloud* acquired by the camera with the virtual model of the workpiece. In particular, this research uses the ICP method (Iterative Closest Points) [39] to obtain the pose of the workpiece since it is widely used due to its effectiveness and simplicity.

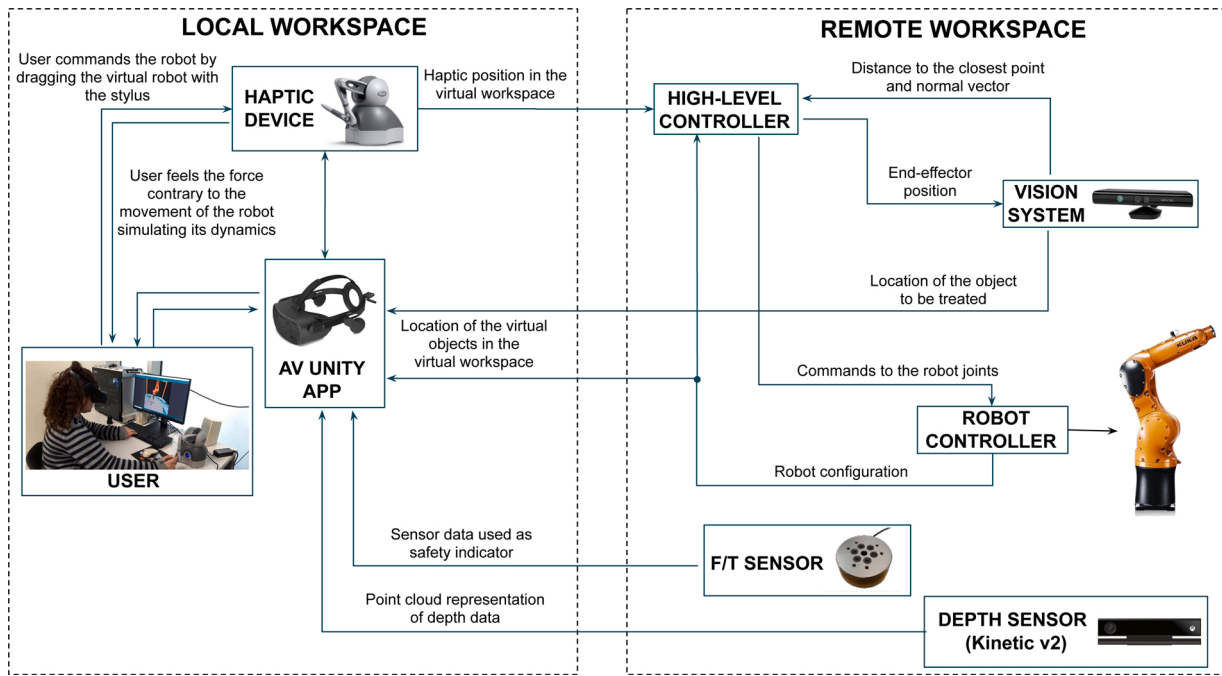


Fig. 1. Remote human-robot interaction using AV with data from depth sensors and haptics to feel the robot movement. A force sensor is included to protect both the workpiece and the robot tool.

2.2. Kinematics

The kinematics of the robot system can be expressed as:

$$p = l(q), \quad (1)$$

$$\dot{p} = \frac{\partial l(q)}{\partial q} \dot{q} = J\dot{q}, \quad (2)$$

$$\ddot{p} = J\ddot{q} + \dot{J}\dot{q}, \quad (3)$$

being $p = [x \ y \ z \ \alpha \ \beta \ \gamma]^T$ the pose of the robotic system, where α , β and γ represent the orientation angles (roll, pitch and yaw, respectively), $q = [q_1 \ \dots \ q_n]^T$ the configuration of the robotic system, J the Jacobian matrix and l the so-called kinematic function [40].

2.3. Task prioritization method

This method is useful to address a set of tasks with different priorities [41], where the error of the task equations has to be minimized. The recursive equations of this strategy are given below [42]:

$$A_i \bar{x} = b_i, \quad i = 1, \dots, M, \quad (4)$$

$$\bar{x}_i = \bar{x}_{i-1} + (A_i N_{i-1})^\dagger (b_i - A_i \bar{x}_{i-1}), \quad i = 1, \dots, M, \quad (5)$$

$$N_i = N_{i-1} (I - (A_i N_{i-1})^\dagger (A_i N_{i-1})), \quad i = 1, \dots, M, \quad (6)$$

being M the considered number of equalites or tasks, A_i and b_i the matrix and vector, respectively, for the i th task ($i = 1$ represents the highest priority), \bar{x}_M the solution that minimizes the error of the task equations, $N_0 = I$ the identity matrix, $\bar{x}_0 = \mathbf{0}$ the null vector and superscript \dagger the pseudoinverse of a matrix [43] (a threshold can be used to neglect the small singular values). Note that \bar{x} is the unknown vector to be computed, which corresponds to the commanded accelerations in this work, whereas \bar{x}_i denotes the solution to this unknown vector computed by the task prioritization algorithm taking into account the first i tasks.

2.4. Non-conventional sliding mode control

Sliding Mode Control (SMC) [44] is widely utilized in robot tasks since it has the inherent advantages of robustness and low computation cost, e.g., see [45,46], among others.

The modified SMC presented in [47] is used in this research to limit the approach speed of the robot tool to the workpiece. In particular, this approach is useful to satisfy *inequality constraints* as discussed next.

Consider the inequality constraints and state equation below:

$$\phi_{in,i}(x) \leq 0, \quad i = 1, \dots, N_{in}, \quad (7)$$

$$\dot{x} = f(x, d) + g(x)u, \quad (8)$$

being vector x the state, N_{in} the number of constraints, $\phi_{in,i}$ the i th constraint function, which is assumed to be differentiable, vector u the control action, vector d the disturbance, f the drift vector field and g a set of control vector fields.

Then, the fulfillment of the constraints (7) is guaranteed if the value of the control action u fulfills the following expression [47]:

$$v2dm(\text{pos}(\phi_{in})) L_g \phi_{in} u = -\text{pos}(\phi_{in}) u_{in}^+, \quad (9)$$

where $v2dm(\cdot)$ is a function that returns a diagonal matrix from a vector; $\text{pos}(\cdot)$ denotes the *positive function*, that is, if $x \leq 0$ then $\text{pos}(x) = 0$, otherwise $\text{pos}(x) = 1$; column vector ϕ_{in} is composed of the constraint functions $\phi_{in,i}$ of all the inequalities; $L_g \phi_{in}$ is a matrix containing the row vectors $L_g \phi_{in,i} = \frac{\partial \phi_{in,i}}{\partial x} g$ of all inequality constraints; and u_{in}^+ represents the switching gain, which is a high enough positive scalar.

See [47] for further details about the above SMC.

3. Proposed application

3.1. Overview

The application developed in this work consists of two workspaces: the local workspace in which the AV headset is used and the remote workspace in which the robotic system operates, as shown in Fig. 1.

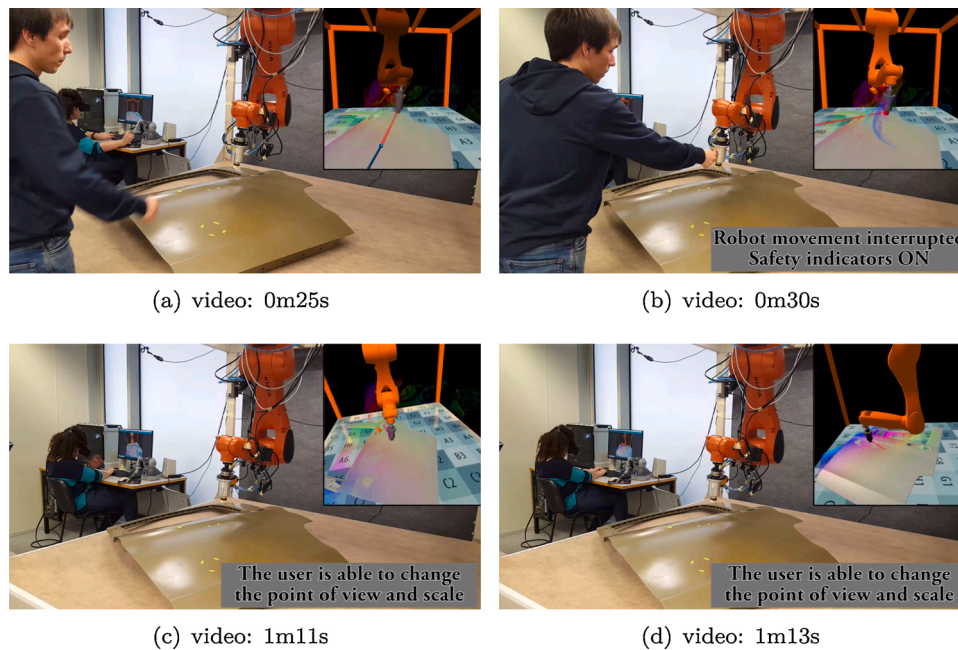


Fig. 2. Frames of the video showing the functionalities of the proposed AV-based user interface. See video at <https://media.upv.es/player/?id=5edfec30-1e12-11eb-9463-339a8e543e47>. (For interpretation of the references to color in this figure citation, the reader is referred to the web version of this article.)

In the *local workspace*, the user is able to visualize the robot and its environment by wearing the VR headset. For this purpose, the real world objects (robot, object to be treated, table, etc.) have been modeled and included in the virtual world. The location of these virtual objects in the virtual world is updated according to that of the corresponding real world objects, which is obtained online from sensor measurements. In particular, in the proposed application, the robot configuration is obtained reading the joint values from the robot controller, whereas the accurate location of the object to be treated is obtained using a vision system based on three 3D cameras (Kinect v1 sensors) focused on the object area.

In addition, for safety, another 3D camera (Kinect v2 depth sensor) has been used to capture non-modeled data of the robot environment in the form of a point cloud, which is integrated within the VR environment.¹ This allows the user to visually verify if this non-modeled real world data matches the virtual objects. If that is not the case (e.g., the virtual object corresponding to the object to be treated has not been accurately placed in the virtual world, or the arm of a human operator has been detected within the robot workspace), the user may decide, for security reasons, to abort the remote robot teleoperation.

For further security, an F/T (Force–Torque) sensor mounted on the robot end-effector is also utilized as a safety indicator to stop the motion of the robot system when the sensor measurements are abnormally large, which can be due to collisions or abrupt movements performed by the user. Note that, depending on the specific tool attached to the sensor, a compensation of the values provided by the F/T sensor is required to obtain values near to zero when the tool is still and out of contact with the environment. Note also that the characteristics of the selected F/T sensor will depend on the specific treatment application, which means not only its physical size but also its static and dynamic properties, i.e., resolution, sensitivity, time response, etc.

Moreover, a haptic device consisting of a stylus is used to allow the user dragging the virtual model of the robot through the virtual workspace, thus resulting in the movement of the real robot. In this sense, the

¹ The points of the cloud are represented in the VR environment using a color map instead of their real RGB color due to the preferences expressed by the users during the tests conducted in this work.

force feedback from the haptic device has two main functionalities. Firstly, the haptic feedback allows the user to properly teleoperate the robot according to its real movement. This is accomplished essentially by means of a “spring effect” between the haptic pointer and the actual position of the robot tool.² Thus, the larger the error, the larger the force feedback, making the user move slower the haptic stylus. Secondly, the haptic feedback allows the user to “feel” the presence of virtual objects, e.g., the modeled workpiece, and act accordingly. In particular, the haptic device blocks the further movement of the stylus in the normal direction at the contact point on the virtual object.

In the *remote workspace*, the high level controller of the robot receives the position command from the VR application, which corresponds to the haptic position in the virtual workspace that is given by the movement performed by user with the stylus. The controller also receives from the vision system mentioned above the separation distance between the tool of the robot system and the closest point of the workpiece together with the normal vector at this point of the workpiece surface. Thus, according to these values, the high-level controller computes a proper tool pose (position and orientation) and commands the corresponding joint values to the robot controller. The high-level controller for the robotic system is detailed in Section 4.

3.2. Augmented virtuality-based user interface

The Unity 3D Engine was used to implement the user interface. As mentioned above, a 3D model of the robot and its environment has been replicated in the application, see Fig. 2. The user is able to command the robot motion by means of the haptic stylus, i.e., the robot tracks the stylus position. However, for safety reasons, a required condition to start the robot motion is that the stylus position in the virtual environment must be close to the position of end-effector of the virtual robot. In particular, the state indicator in the robot end-effector turns from blue to

² Note that a small high frequency component will be introduced in the haptic feedback due to the SMC used to move the robot tool. However, this small high frequency component will be filtered by the haptic device itself and, hence, the user will not experience any chattering effect when guiding the robot.

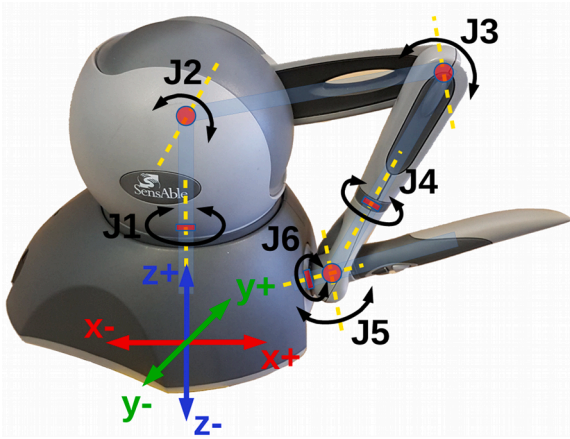


Fig. 3. Phantom haptic device with 6 revolute joints used to remotely teleoperate the robot. The red circles represent hinges, i.e., joints that rotate around a common normal axis of two consecutive links; whereas the red rectangles represent axial rotations, i.e., joints where both links are aligned and the rotation is around the common axis.

green when the stylus position is close enough to start the robot motion, which can be done after pressing the “deadman” switch located in the haptic device. Fig. 2a shows the robot model once the deadman switch has been pressed, which changes into a darker tone to indicate that the robot is being commanded by the movements made by the user with the stylus.

While the robot is in motion, if the F/T sensor detects a collision, the state indicator in the end-effector of the robot turns red, as shown in Fig. 2b, and the robot stops its movement. When this occurs, the user is allowed to move the robot in order to retrieve the end-effector from the collision situation, but prevents the user from moving the robot forward into the direction in which the collision was met. The movement after a collision is indicated by the orange color in the state indicator. Once the forces measured by the F/T sensor are again under the threshold value, the state indicator turns blue or green depending on the distance from the stylus to the end-effector position.

For this application, it has also been implemented a zoom and positioning system that allows the user to accommodate its position in the virtual environment using the keyboard of the computer in order to facilitate performing the task. Once the zoom function has been activated, the user can change the point of view and scale of the virtual world, see Fig. 2c and d. Furthermore, the user is also able to reduce the moving velocity of the robot.

3.3. Haptic feedback

A Phantom Omni haptic device [48] is used to remotely teleoperate the robot. From a kinematic point of view, this device is equivalent to a small robot arm with six revolute joints with the standard configuration: hip-shoulder-elbow and in-line wrist, see Fig. 3. Although this device allows six degrees of freedom, i.e., three linear coordinates (Cartesian position) plus three Euler angles, only the Cartesian position is going to be employed by the user to remotely teleoperate the robot, since the tool orientation is automatically controlled by the high-level controller of the robotic system, see Section 4.

The haptic feedback mentioned in Section 3.1 has been implemented considering the well-known impedance/admittance control [49]. This kind of controller is typically used to relate force-torque data and the robot motion. In this work, this control is first used to compute the force feedback for the haptic device (only the first three joints can be actuated, which are used to achieve the desired linear force vector in the center of the device wrist) from the Cartesian error e_h between the stylus position in the virtual workspace and the position of the end-effector of the virtual robot. In particular, the following equation is used to compute the force that the user “feels” from the haptic device:

$$F_h = K_h e_h + D_h \dot{e}_h + A_h \ddot{e}_h, \tag{10}$$

where F_h is a 3-dimensional column vector containing the linear forces to be achieved by the haptic device and diagonal matrices K_h , D_h and A_h represent the parameters of the impedance controller. The computed value F_h is used only when the deadman switch is pressed and the robot is in motion, otherwise there is no haptic feedback felt by the user.

4. High-level controller of the robot

Without loss of generality, a surface treatment operation is considered to illustrate the application developed in Section 3. In this operation, the human operator and the robot system cooperatively perform the treatment on the workpiece surface. Hence, the operator remotely teleoperates the tool position, whereas the robot automatically ensures not only the perpendicularity between the workpiece surface and the robot tool but also a smooth approach to the workpiece.

Fig. 4 depicts the control scheme of the high-level controller for the robotic system in order to carry out the mentioned surface treatment operation. In particular, three prioritized levels are considered to simultaneously accomplish several tasks. The first level, i.e., the highest priority level, is utilized to ensure that the tool is perpendicular to the workpiece surface. The second level is included to limit the approach speed of the robot tool to the workpiece. Lastly, the third level, i.e., the lowest priority level, is utilized to conduct the surface treatment on a

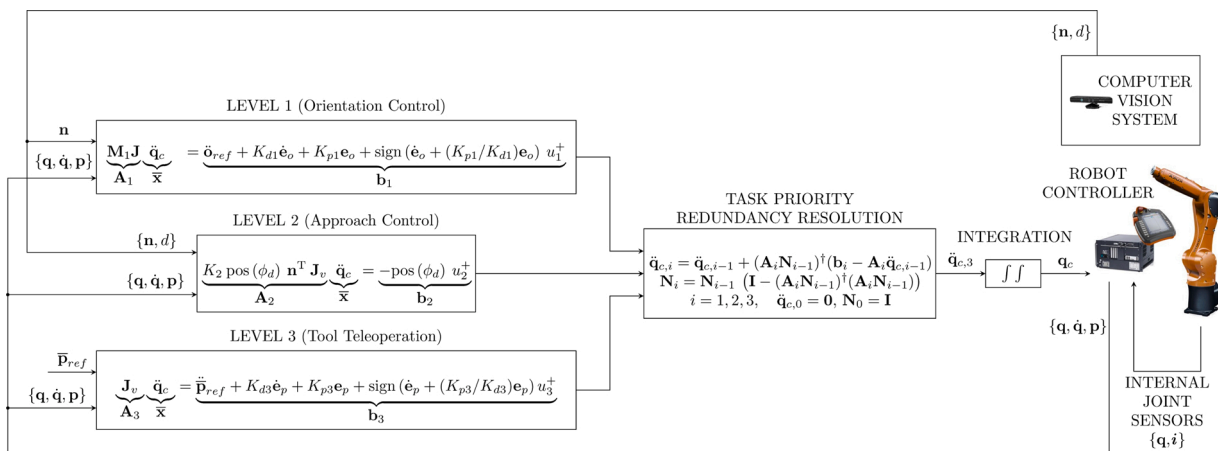


Fig. 4. Control scheme of the high-level controller of the robot system.

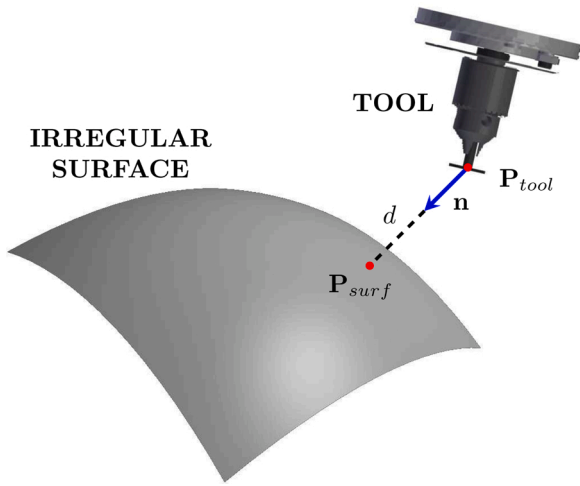


Fig. 5. Graphical illustration of the information obtained from the machine vision system.

particular part of the workpiece by means of tool guidance, i.e., the user remotely commands the tool position using the haptic device.

The following input information is considered for these levels: the robot pose p and the robot configuration and its derivative $\{q, \dot{q}\}$, which are obtained from the robot controller; the reference \bar{p}_{ref} for the tool position $\bar{p} = [x \ y \ z]^T$, which is remotely provided by the user with the haptic device; and the data $\{d, n\}$ obtained using machine vision, where d represents the length of the vector from the robot tool, see Fig. 5, to the nearest point of the workpiece, whereas n denotes the unit vector of the mentioned vector (note that n is normal to the workpiece surface as long as it is smooth at the nearest point to the tool).

The equation $A_i \bar{x} = b_i$ (4) for each priority level is obtained below, where \bar{x} corresponds to the commanded acceleration \ddot{q}_c for the robot system. The errors of these equations are minimized using (5) and (6), as shown in Fig. 4. Thus, the acceleration command $\ddot{q}_{c,3}$ is double integrated to get the robot configuration command q_c . Finally, the low-level robot controller defines an inner control loop to track the commanded values q_c taking into account the internal measurements of the joint angles q and joint currents i [50,51]. Note that this inner loop has already been developed and provided by the robot manufacturer. Thus, d_c stands for the inaccuracy of this inner loop, which is assumed to be

Table 1

Code of the algorithm.

Executed every sampling time T_s	
1	$[q, \dot{q}, n, d] = \text{GetRobotStateAndVisionData};$
2	$o_{ref} = \text{OrientationOfVector}(n);$
3	$p = l(q);$ // Eq. (1)
4	$\dot{p} = J\dot{q};$ // Eq. (2)
5	$\phi_d = \epsilon_d - d + K_2 n^T J_v \dot{q};$ // Eqs. (12),(13)
6	$\dot{o}_{ref} = (o_{ref} - o_{ref,prev})/T_s;$ // Derivative
7	$\ddot{o}_{ref} = (\dot{o}_{ref} - \dot{o}_{ref,prev})/T_s;$ // Derivative
8	$\bar{p}_{ref} = (\bar{p}_{ref} - \bar{p}_{ref,prev})/T_s;$ // Derivative
9	$\ddot{\bar{p}}_{ref} = (\dot{\bar{p}}_{ref} - \dot{\bar{p}}_{ref,prev})/T_s;$ // Derivative
10	$e_o = (o_{ref} - M_1 p);$ // Orientation error
11	$\dot{e}_o = (\dot{o}_{ref} - M_1 \dot{p});$ // Derivative of orientation error
12	$e_p = (\bar{p}_{ref} - \bar{p});$ // Position error
13	$\dot{e}_p = (\dot{\bar{p}}_{ref} - \dot{\bar{p}});$ // Derivative of position error
14	$A_1 = M_1 J;$ // Eq. (11)
15	$b_1 = \ddot{o}_{ref} + K_{d1} \dot{e}_o + K_{p1} e_o + \text{sign}(\dot{e}_o + (K_{p1}/K_{d1})e_o) u_1^+;$ // Eq. (11)
16	$A_2 = K_2 \text{pos}(\phi_d) n^T J_v;$ // Eq. (15)
17	$b_2 = -\text{pos}(\phi_d) u_2^+;$ // Eq. (15)
18	$A_3 = J_v;$ // Eq. (17)
19	$b_3 = \ddot{\bar{p}}_{ref} + K_{d3} \dot{e}_p + K_{p3} e_p + \text{sign}(\dot{e}_p + (K_{p3}/K_{d3})e_p) u_3^+;$ // Eq. (17)
20	$\ddot{q}_{c,1} = A_1^\dagger b_1;$ // Eq. (5), $i = 1$
21	$N_1 = I - A_1^\dagger A_1;$ // Eq. (6), $i = 1$
22	$\ddot{q}_{c,2} = \ddot{q}_{c,1} + (A_2 N_1)^\dagger (b_2 - A_2 \ddot{q}_{c,1});$ // Eq. (5), $i = 2$
23	$N_2 = N_1 (I - (A_2 N_1)^\dagger (A_2 N_1));$ // Eq. (6), $i = 2$
24	$\ddot{q}_{c,3} = \ddot{q}_{c,2} + (A_3 N_2)^\dagger (b_3 - A_3 \ddot{q}_{c,2});$ // Eq. (5), $i = 3$
25	$\dot{q}_c = \dot{q}_{c,prev} + \ddot{q}_{c,3} T_s;$ // Integration
26	$q_c = q_{c,prev} + \dot{q}_c T_s;$ // Integration
27	SendToJointControllers($\ddot{q}_{c,3}$);
28	$o_{ref,prev} = o_{ref};$ // For next iteration
29	$\dot{o}_{ref,prev} = \dot{o}_{ref};$ // For next iteration
30	$\bar{p}_{ref,prev} = \bar{p}_{ref};$ // For next iteration
31	$\dot{\bar{p}}_{ref,prev} = \dot{\bar{p}}_{ref};$ // For next iteration

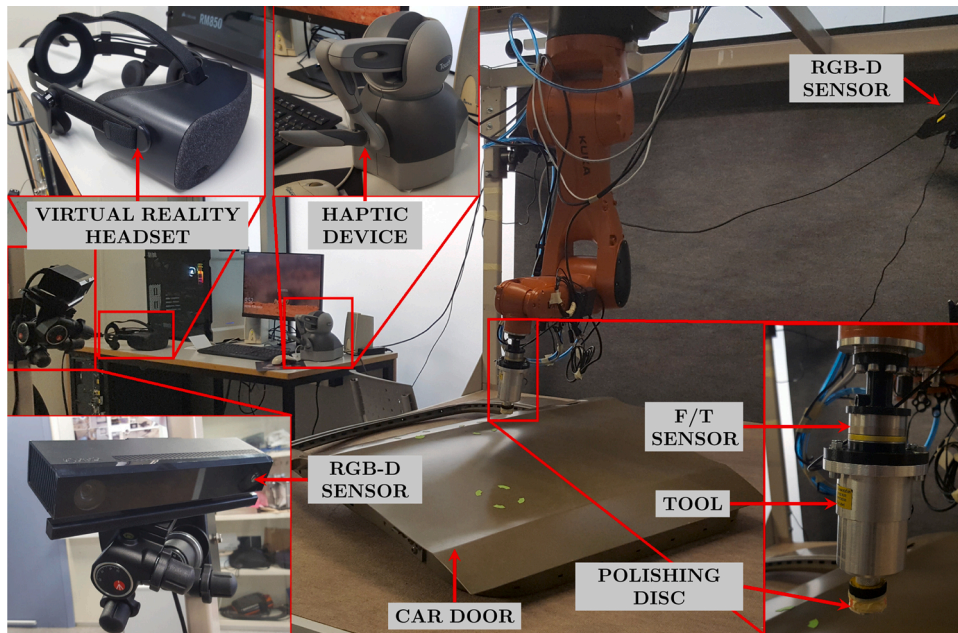


Fig. 6. Experimental platform used for the real experimentation: a 6R robot arm with a F/T sensor, four RGB-D cameras, a polishing tool, a haptic device, a VR headset and a car door.

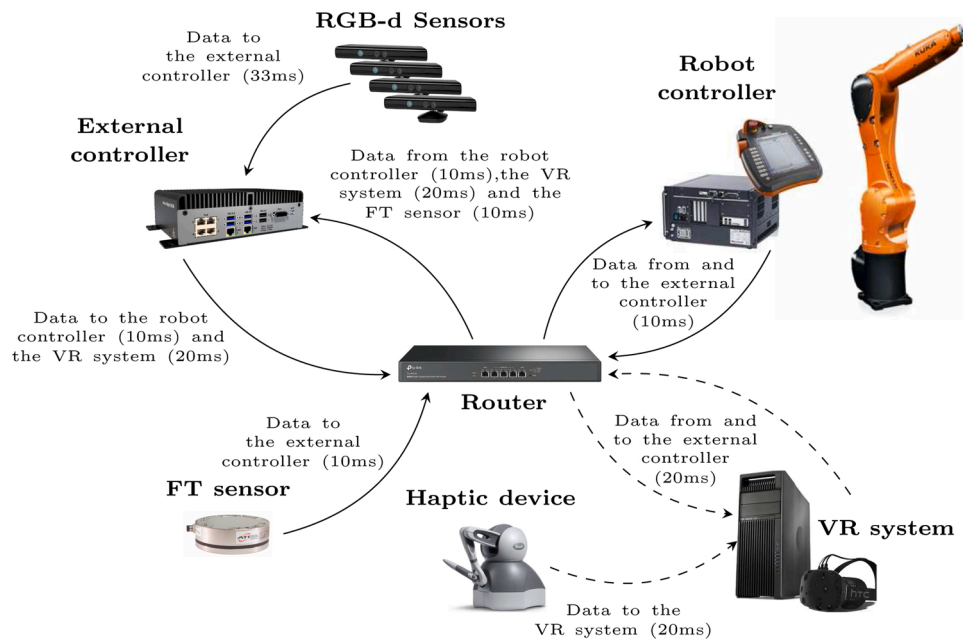


Fig. 7. Hardware communications. The dashed and solid lines are used to represent wireless and wired communications, respectively.

bounded. However, note that the value of d_c does not need to be computed nor estimated since the proposed SMC-based controller, which is detailed below, is inherently robust against this error.

4.1. Level 1: Orientation control

A key requirement for surface treatment operations is that the robot tool has to be orthogonal to the workpiece surface, that is, the Z-axis of the robot tool (see Fig. 5) must point in the direction of n . Thus, the reference for the tool orientation is vector n , which can be easily transformed [52] to roll and pitch reference values, i.e., α_{ref} and β_{ref} . It is worth noting that there is no requirement for the yaw angle and, hence, it could be used for other purposes.

Thus, the control equation for Level 1 results in:

$$M_1 J \ddot{q}_c = \ddot{o}_{ref} + K_{d1} \dot{e}_o + K_{p1} e_o + \text{sign}(\dot{e}_o + (K_{p1}/K_{d1})e_o) u_1^+ \rightarrow A_1 \ddot{q}_c = b_1, \quad (11)$$

where matrix $M_1 = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$ is used to affect only α and β angles (i.e., roll and pitch) of the pose vector p ; vector $o_{ref} = [\alpha_{ref} \ \beta_{ref}]^T$ represents the reference orientation; vector $e_o = [\alpha \ \beta]^T$ denotes the error of roll and pitch angles; K_{p1} and K_{d1} are the correction gains for the roll and pitch angles and their derivatives, respectively (note that $\dot{\alpha}$ and $\dot{\beta}$ can be readily obtain from the robot



Fig. 8. Frames of the video of the first experiment.

equations in (2)); u_1^\dagger represents a switching gain; and b_1 and A_1 denote the vector and matrix for the control equation of Level 1.

It is worth noting that (11) represents a hybrid controller, where the last switching term is used to cancel out the last term in (3) and, hence, the computation of the Jacobian derivative is avoided. Note that this hybrid controller in sort uses conventional SMC, whose proof of convergence can be found in [44].

4.2. Level 2: Approach control

In order to limit the approach speed of the robot tool to the workpiece, the following constraint is used:

$$\phi_d = \epsilon_d - d - K_2 \dot{d} \leq 0, \quad (12)$$

where ϵ_d represents a security margin between the tip of the tool and the

workpiece surface and K_2 is a free design parameter that establishes the maximum approach speed allowed depending on the separation distance between the tool and the workpiece surface. Hence, the maximum allowed speed tends to zero as the mentioned separation distance tends to zero.

Considering that the motion of the workpiece (in case it is not static) is significantly slower than the motion of the robot system, the derivative of the distance d in (12) is readily obtained from the robot velocity as detailed below:

$$\dot{d} = (\partial d / \partial q)^T \dot{q} = ((\partial \bar{p} / \partial q)^T (\partial d / \partial \bar{p}))^T \dot{q} = (J_v^T (-n))^T \dot{q} = -n^T J_v \dot{q}, \quad (13)$$

where matrix J_v represents the top 3×3 submatrix of the Jacobian J .

In order to use the SMC detailed in Section 2.4 to satisfy the approach constraint in (12), the following second-order dynamical system (8) is considered:

$$\dot{x} = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix} x + d + \begin{bmatrix} 0 \\ I \end{bmatrix} u, \quad (14)$$

where $x = [q^T \ \dot{q}^T]^T$, $u = \ddot{q}_c$ and $d = d_c$.

From Eqs. (9) and (12), the control equation for Level 2 results in:

$$\begin{aligned} \text{pos}(\phi_d) L_g \phi_d \ddot{q}_c &= -\text{pos}(\phi_d) u_2^+, \\ &\rightarrow A_2 \ddot{q}_c = b_2, \end{aligned} \quad (15)$$

where u_2^+ represents switching gain of the SMC, b_2 and A_2 denote the vector and matrix for the control equation of Level 2 and, according to (12)–(14), matrix $L_g \phi_d$ is given by:

$$L_g \phi_d = (\partial \phi_d / \partial x)^T g = (\partial \phi_d / \partial \dot{q})^T = -K_2 (\partial \dot{d} / \partial \dot{q})^T = K_2 n^T J_v. \quad (16)$$

4.3. Level 3: Tool teleoperation

This level is included to conduct the surface treatment on a particular part of the workpiece by means of tool guidance, i.e., the user remotely commands the tool position using the haptic device. In particular, and similarly to (11), the controller used to track the reference Cartesian position \bar{p}_{ref} is as follows:

$$J_v \ddot{q}_c = \ddot{\bar{p}}_{ref} + K_{d3} \dot{e}_p + K_{p3} e_p + \text{sign}(\dot{e}_p + (K_{p3}/K_{d3}) e_p) u_3^+ \rightarrow A_3 \ddot{q}_c = b_3 \quad (17)$$

where $e_p = \bar{p}_{ref} - \bar{p}$ denotes the tool pose error; K_{p3} is the correction gain for the pose error; K_{d3} is the correction gain for the pose error derivative; the tool linear velocity \bar{p} is obtained from the robot kinematic equation in (2); u_3^+ represents a switching gain; and b_3 and A_3 denote the vector and matrix for the control equation of Level 3.

Note that the reference \bar{p}_{ref} for the tool position is generated by the human operator using the haptic device. In this process, the operator has to visually spot in the real image of the workpiece surface the points that require the treatment and teleoperate the robot to these points. In order to assist the operator, some marks can be added in the real or virtual workpiece [53] to help the user locating these points. For this purpose, some automatic system could be used to obtain the points that require the treatment, such as [54,55,5] in the automotive industry. However, whether this automatically generated data is available or not, the human operator has to check the points and is ultimately responsible for teleoperating the robot to one position or another. That is, the operator does not command the robot “blindly” according to automatic data, since this data may not be complete nor perfectly accurate. Note that this is also the procedure currently used to manually repair paint defects in the automotive industry [56,57].

4.4. Implementation of the control algorithm

Table 1 shows the pseudo-code of the high-level controller developed in this work for the robot system. Note that it is assumed that this high-level controller is implemented in a device external to the robot controller, such as a Programmable Logic Controller (PLC) or an industrial workstation. However, if needed, the proposed approach could also be implemented directly in the robot controller.

Each line of the code is explained as follows. The first line of code, i. e., LC1, is used to update the readings from the sensors. LC2 is used to compute the orientation angles of n , which is provided by the computer vision. LC3 and LC4 are used to compute the robot pose and its derivative from the robot kinematics. LC5 is used to compute the constraint function of the inequality in Level 2. LC6 to LC9 are used to compute the time derivative of several signals. LC10 to LC13 are used to compute the orientation and pose errors and their derivatives. LC14 and LC15 are used to compute the matrix and vector, respectively, for the control equation of Level 1. LC16 and LC17 are used to compute the matrix and vector, respectively, for the control equation of Level 2. LC18 and LC19

are used to compute the matrix and vector, respectively, for the control equation of Level 3. LC20 to LC24 are used to compute the solution of the commanded accelerations that minimizes the errors of the control equations of the three levels. LC25 and LC26 are used to compute the commanded positions by integrating twice the commanded accelerations. LC27 is used to send the commanded values to the robot controller. LC28 to LC31 are used to update the previous value of several signals in order to use them in the next iteration.

Note that the derivatives \dot{o}_{ref} , \ddot{o}_{ref} , $\dot{\bar{p}}_{ref}$ and $\ddot{\bar{p}}_{ref}$ are computed in the code of Table 1 using numerical differentiation. However, depending on

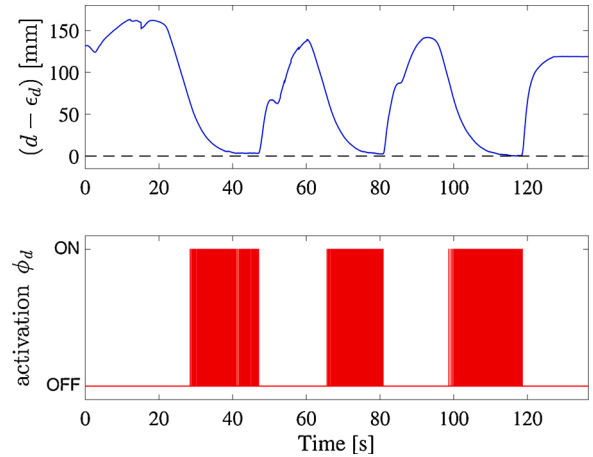


Fig. 9. Graphs for the approach constraint in the first experiment. Top: distance of separation between the tool of the robotic arm and the workpiece surface as a function of time. Bottom: activation of the approach constraint.

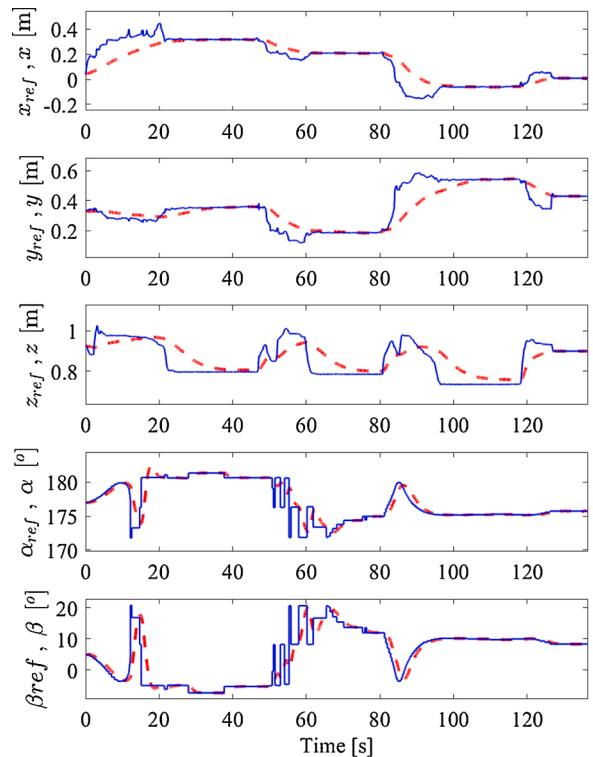


Fig. 10. Tool pose and reference pose in the first experiment. From top to bottom: linear coordinates x , y and z , and roll and pitch angles. The solid-blue lines denote the actual values, whereas the dashed-red lines denote the reference values supplied by the user (coordinates x , y and z) and machine vision system (roll and pitch angles). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

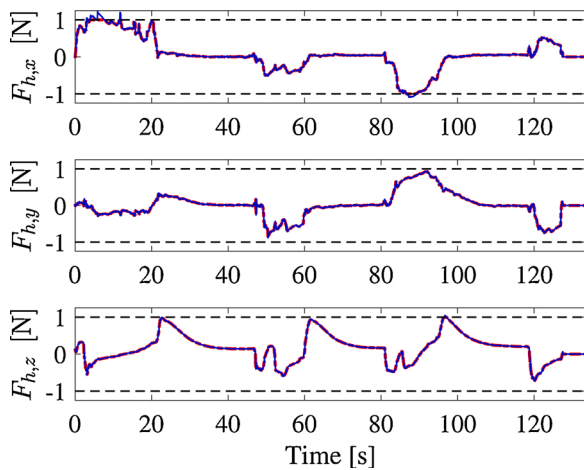


Fig. 11. Force feedback of the haptic device in the first experiment. The solid-blue lines denote the computed values for the linear forces, whereas the dashed-red lines denote the values sent to the haptic device taking into account the maximum allowed values, which are depicted with horizontal dashed-black lines. Note that both values are so similar that they can be hardly distinguished. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the application, this approach could lead to excessive noise in the signals. In order to avoid this issue, the sampling period T_s of the control algorithm should be chosen large enough in order to avoid noisy signals.³ For instance, in the specific case of the experimentation in Section 5, a synchronous sampling period of 20 ms gave rise to negligible noise in numerical differentiation.

The computation of one iteration of the algorithm in Table 1 (compiled C code) takes around 0.015 ms for the case in Section 5.

5. Experiments

5.1. Experimental platform

The experimental platform used in this work is shown in Fig. 6, which is composed of: a 6R robot arm (Kuka KR6 Agilus); a tool consisting of a spot repair polisher (Mirka AROS-B 150NV) placed in the robot end-effector using a self-developed adapter; a cylinder of $29 \times 29 \times 23$ mm used as polishing disc; a 45 mm F/T steel sensor with DataBox V1.0 electronics (see [58]), which is attached between the end-effector of the robot arm and the polisher; four RGB-D cameras (three Microsoft Kinect v1 sensors and one Microsoft Kinect v2 sensor); a workpiece consisting of a car door; a Phantom Omni haptic device (3D Systems Touch, the maximum possible value for the force feedback is ± 1 N in each linear axis); and a VR Headset (HP Reverb VR – Pro Edition).

In the remote workspace (see Fig. 1), an external computer was used to implement the robot high-level controller detailed in Section 4.4. Moreover, the robot controller, F/T sensor and external computer communicated by means of an Ethernet switch. In addition, the RGB-D cameras were connected to serial ports of the external computer.

In the local workspace (see Fig. 1), the proposed AV-based user interface detailed in Section 3.2) was implemented in another external workstation using the software Unity 2019.3.0.b14 version.

Fig. 7 shows the periods of the communications between the different devices. Specifically, the communication periods are: 10 ms

³ Note that if a small sampling period is chosen, a low-pass filter would be required to remove the noise from the derivatives. However, the bandwidth of this approach is approximately equivalent to use a larger sampling period with no filtering.

between the robot controller and the external controller; 20 ms between the VR system and the external controller; 10 ms between the F/T sensor and the external controller; 20 ms between the haptic device and the VR system; and 33 ms between the camera and the external controller.

These communication periods yield the following maximum delays: it takes at most 50 ms ($20 + 20 + 10$) to reflect in the real robotic system the movement performed by the user in the haptic device; it takes at most 30 ms ($10 + 20$) to reflect in the user interface the actual movement of the robot system; and the reflections of the measurements of the camera and F/T sensor in the user interface take at most 53 ms ($33 + 20$) and 30 ms ($10 + 20$), respectively. Note that all these delays are negligible and almost imperceptible to the user during the teleoperation task.

The maximum workpiece position error given by the used 3D camera network was around 1 mm, which is acceptable for the proposed application. However, more accurate depth sensors could be used for applications requiring more precision.

5.2. Values of the parameters

The values used for the control algorithm parameters are given below.

- (i) Sampling period: $T_s = 0.01$ s.
- (ii) Impedance control parameters (Section 3.3): $K_h = 4.5$ I, $D_h = 0.1$ I and $A_h = 0.02$ I.
- iii) Parameters of Level 1 (Section 4.1): $K_{p1} = 1.5$, $K_{d1} = 1.8$ and $u_1^+ = 0.01$.
- iv) Parameters of Level 2 (Section 4.2): $\epsilon_d = 2$ mm, $K_2 = 2.5$ and $u_2^+ = 0.45$.
- v) Parameters of Level 3 (Section 4.3): $K_{p3} = 2$, $K_{d3} = 4.2$ and $u_3^+ = 0.01$.

The above control parameter values were experimentally set in a previous stage to obtain a proper performance of the setup shown in Fig. 6 for the surface treatment task (note that the control parameter values remain constant for all the experiments below). For instance, the (position) correction gains K_{p1} and K_{p3} were set large enough to obtain a fast, but stable, response of the robot system and, subsequently, the (speed) correction gains K_{d1} and K_{d3} were set large enough to avoid large overshoot values.

5.3. Results

Two experiments were conducted to validate the performance of the proposed approach. The objective of the first experiment was to perform the surface treatment with the robot tool at three points (e.g., polishing tasks) of a car door surface. It is assumed that the three *points* are given by some automatic defect detection system [55–57], depicted in the virtual car door surface and used by the operator as a visual reference to teleoperate the robot. The objective of the second experiment was to perform a profiling task (e.g., fine grinding) on the car door surface. As before, it is assumed that the *path* associated with the profiling task is given, depicted in the virtual car body surface and used by the operator as a visual reference to teleoperate the robot.

The video recording of the first experiment is available at <https://media.upv.es/player/?id=84e85220-1e10-11eb-9463-339a8e543e47>, where it can be appreciated how the user utilizes the AV-based interface and the haptic device to remotely teleoperate the tool of the robot system to the three goal points in order to conduct the treatment operation. Moreover, the orthogonality and smooth approach of the robot tool to the workpiece are ensured by the proposed high-level controller of the robot (see Section 4), which receives in real-time the required data from the vision sensors, i.e., the normal vector and distance to the surface of the car door.

Several frames of this video recording are shown in Fig. 8: in the

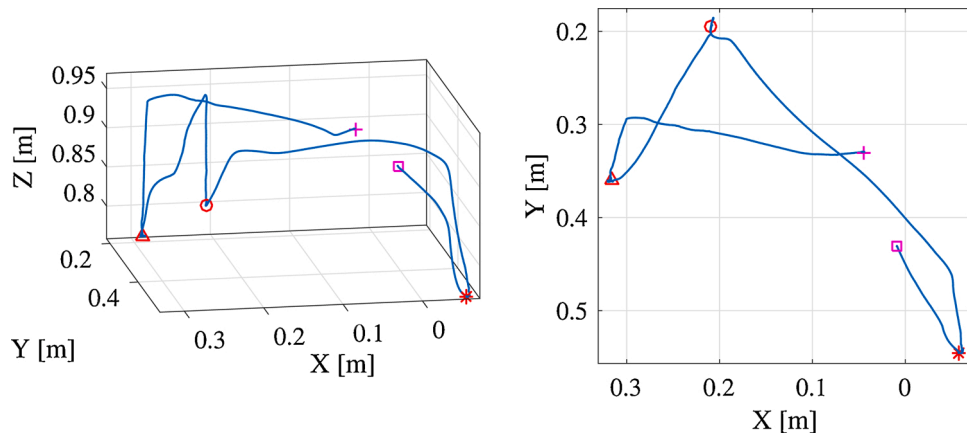


Fig. 12. Trajectory followed by the robot tool in the first experiment. Cross and square symbols denote the initial and final positions, respectively, whereas triangle, circle and star symbols denote the positions of the first, second and third goal points, respectively.

interval 32–42 s, see Fig. 8a and b, the user teleoperates the tool to the first goal point while its vertical speed to the workpiece (i.e., the car door) is progressively reduced; at around 50 s, see Fig. 8c, the robot arm stops its movement, keeping the security distance with the workpiece surface and with the tool orientation pointing to the surface normal in order to perform the treatment; at around 58 s, see Fig. 8d, the user retrieves the robot tool from the first goal point once the surface treatment (e.g., polishing task) has finished; and in the interval 1 m03 s–1 m33 s, see Fig. 8e to h, the user treats another region near a style-line of the workpiece, which is also performed properly regardless of the sharpness around this part of the workpiece.

Next, several graphs are presented to show the quantitative performance of the first experiment. In particular, Fig. 9 presents the performance of the tool approach to the workpiece: it can be seen in the top graph that the separation distance between the tool and the workpiece is always above the safety margin, which is due to the approach control in Level 2, see the constraint activation in the second graph.

Fig. 10 displays the tool pose and the reference pose in the first experiment (the yaw angle is omitted since it has no reference value and, hence, it is kept still). Note that both roll and pitch values follow closely the reference values supplied by the machine vision system, which is due to the orientation control performed in Level 1. With regard to the linear coordinates, both x and y values also follow closely the reference values supplied by the user, which is due to the tracking control performed in Level 3. Furthermore, the behavior for the z component is explained as follows. It can be appreciated that the dynamics of the downward movement is slower than that of the upward movement. This is due to the fact that the downward movement is given not only by the reference tracking control in Level 3 but also by the approach control in Level 2, which limits the approach speed to the workpiece. Moreover, it can be observed that, when the tool is in contact with the workpiece (intervals 38–47, 76–81 and 112–118 s in the graph), there is a small steady-state error between the actual z value and the reference value z_{ref} provided by the user. This is due to the security margin included in the boundary of the constraint in Level 2 and, hence, the tool does not go down further despite that the reference value provided by the user is lower. Note also that, when the tool is in contact with the workpiece, the reference value provided by the user could not go down further due to the blocking of the haptic device in the downward direction due to its interaction with the modeled virtual workpiece.

Fig. 11 shows the force feedback felt by the user through the haptic device, which is contrary to the direction of the user movements with the haptic stylus. Note that the values of the force feedback have been saturated taking into account the maximum values allowed in the haptic device. It is interesting to remark the three peaks of the force signal in the Z -axis, see the bottom plot. These peaks are due to the control actions

in Level 2, which slow down the downward movements (negative Z -axis) of the robot with respect to those teleoperated by the user with the stylus in order to ensure a smooth approach to the three goal points on the car door surface.

Fig. 12 shows the trajectory followed by the robot tool due to the remote teleoperation in Level 3. In particular, note that the robot tool covers a large area during this experiment to achieve the three goal points: around 0.4 m in X - and Y -axes and 0.25 m in Z -axis.

The control signals in the first experiment are presented in Fig. 13, where the commanded accelerations computed by each control level are shown. Note that all three levels contribute to the commanded accelerations.

For the second experiment, a reference path composed of two lines marked on the car door surface is considered (note that both lines are curved according to the shape of the door surface). Thus, the user has to

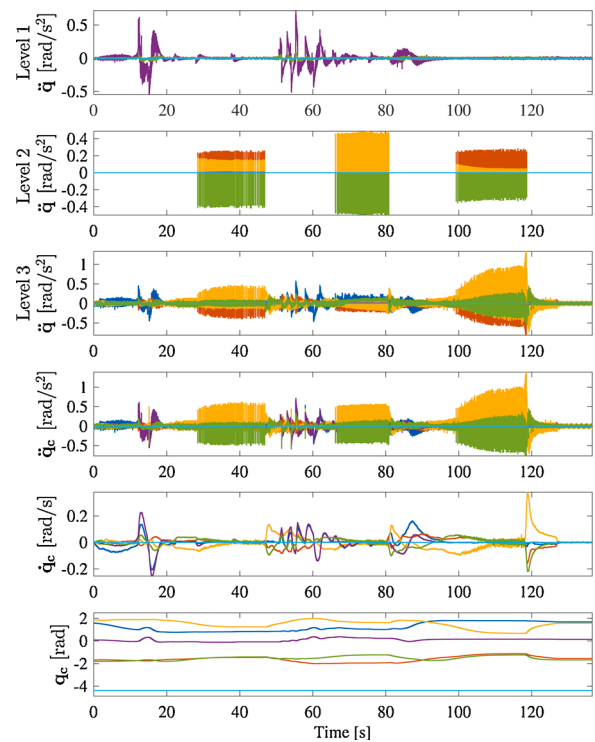


Fig. 13. Control signals in the first experiment. From top to bottom: commanded accelerations computed by each control level; joint accelerations, velocities and positions to be sent to the robot controller.

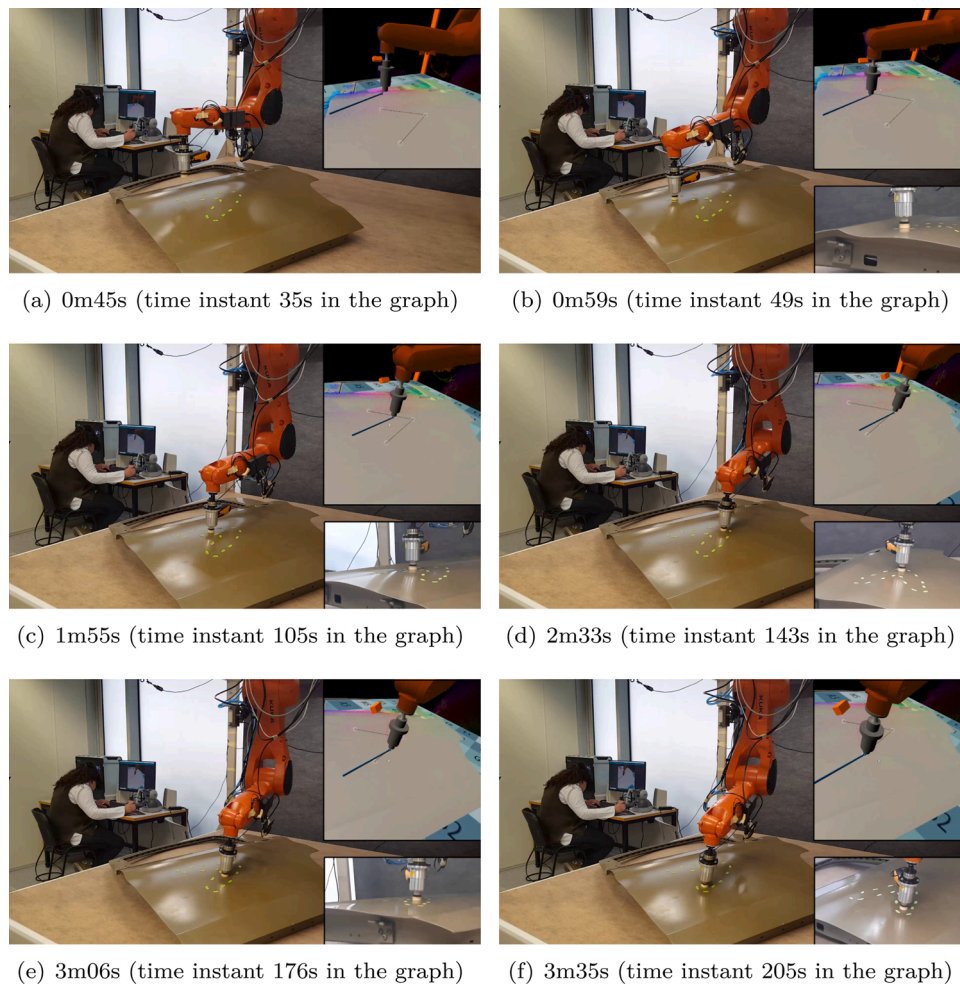


Fig. 14. Frames of the video of the second experiment.

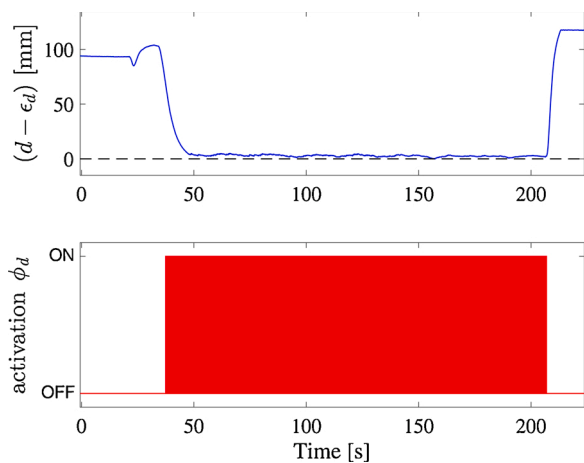


Fig. 15. Graphs for the approach constraint in the second experiment. Top: distance of separation between the tool of the robotic arm and the workpiece surface as a function of time. Bottom: activation of the approach constraint.

teleoperate the robot tool along this reference path, which is marked on both real and virtual doors. The video for this experiment can be played at <https://media.upv.es/player/?id=d8ca06f0-1e14-11eb-9463-339a8e543e47>, where it can be appreciated how the user utilizes the AV-based interface and the haptic device to remotely teleoperate the tool of the robot system in order to conduct the treatment operation following the

reference path marked on the car door surface.

As in the previous experiment, the orthogonality and smooth approach of the robot tool to the workpiece are ensured by the proposed high-level controller of the robot, which receives in real-time the required information from the vision sensors, i.e., the normal vector and distance to the surface of the car door.

Several frames of this video recording are shown in Fig. 14: in the interval 45–59 s, see Fig. 14a and b, the user teleoperates the tool to the starting point of the reference path while its vertical speed to the workpiece (i.e., the car door) is progressively reduced; and in the interval 1 m55 s–3 m35 s, see Fig. 14c to f, the user teleoperates the robot following the reference path, i.e., the two lines, while the robot maintains the tool orientation perpendicular to the surface and keeps the security distance.

Fig. 15 presents the performance of the tool approach to the workpiece when performing the continuous treatment on its surface. As before, it can be seen in the top graph that the separation distance between the tool and the workpiece is always above the safety margin, which is due to the approach control in Level 2, see the constraint activation in the bottom graph.

Fig. 16 shows the tool pose and the reference pose in the second experiment (the yaw angle is omitted since it has no reference value and, hence, it is kept still). As before, both roll and pitch values follow closely the reference values supplied by the machine vision system (in fact, both values are so similar that they can be hardly distinguished), which is due to the orientation control performed in Level 1. With regard to the linear coordinates, both x and y values also follow closely the reference values supplied by the user, which is due to the tracking control performed in

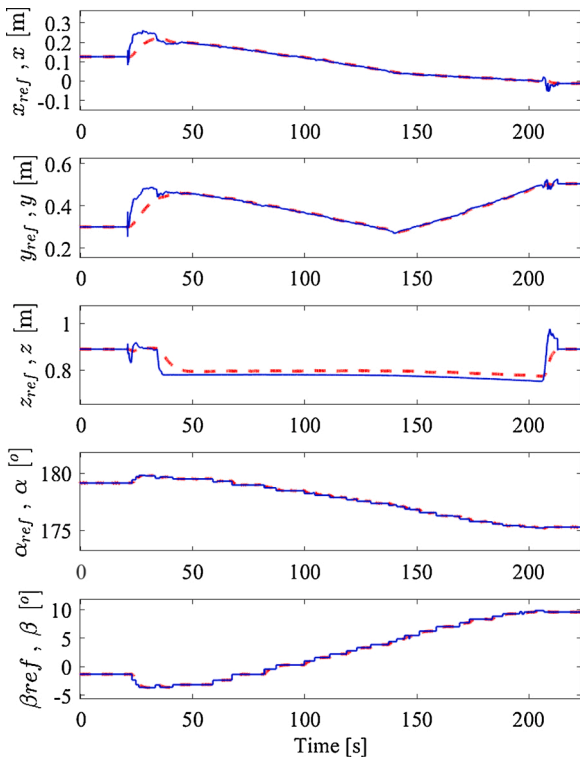
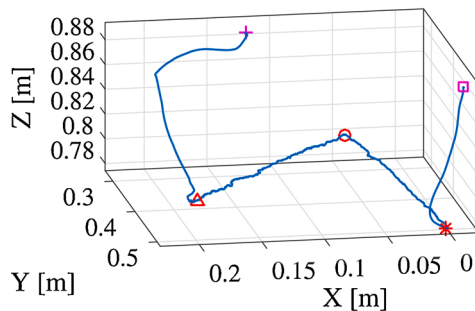


Fig. 16. Tool pose and reference pose in the second experiment. From top to bottom: linear coordinates x , y and z , and roll and pitch angles. The solid-blue lines denote the actual values, whereas the dashed-red lines denote the reference values supplied by the user (coordinates x , y and z) and machine vision system (roll and pitch angles). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Level 3. Moreover, the behavior for the z component is very similar to that explained in the first experiment, see the comments above.

Finally, Fig. 17 shows the trajectory followed by the robot tool due to the remote teleoperation in Level 3. In particular, it can be appreciated that the trajectory described by the robot tool on the object surface corresponds closely to the virtual reference path, i.e., the two lines on the workpiece surface. In fact, the maximum deviation of the path followed by the robot tool on the workpiece surface compared to the reference path was around 6 mm, with a standard deviation of about 2.2 mm. Hence, given that the diameter of the polishing disk is 29 mm, this error is within the allowed tolerance. Note that these error values of



the profiling task include all the potential sources of error: communication delays, accuracy of the workpiece location, robot control, teleoperation system, operator's ability to teleoperate the robot, etc. Therefore, it can be concluded that the *position accuracy* of proposed approach is good enough for the task at hand.

6. Discussion

In the two experiments shown in the previous section, the robot tool exerted a certain force on the workpiece. On the one hand, if this force were too large, the workpiece surface would be damaged. On the other hand, if the force were too small, the treatment would not be performed properly.

Therefore, in order to establish appropriate force values, i.e., maximum and minimum force values for the treatment task, the authors



Fig. 18. Setup used for measuring the forces exerted by the expert on the workpiece surface while performing the surface treatment task: an F/T sensor, a polishing tool and a sample surface with defects.

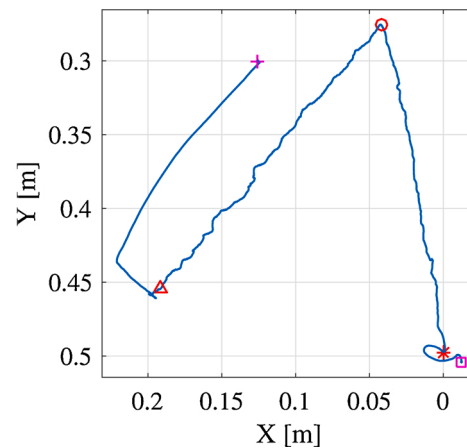


Fig. 17. Trajectory followed by the tool of the robotic system in the second experiment. Cross and square symbols denote the initial and final positions, respectively, whereas triangle, circle and star symbols denote the positions of the initial, medium and final positions, respectively, of the continuous trajectory followed on the workpiece surface.

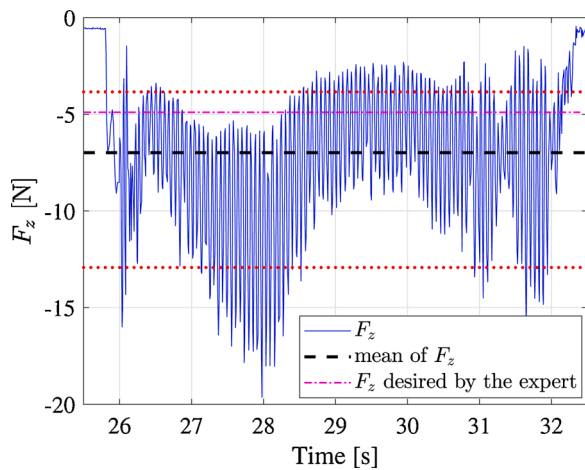


Fig. 19. Force exerted by the expert on the workpiece surface: the solid-blue line corresponds to the force values; the dashed-black line corresponds to the mean value of the force; the dash-dotted-magenta line corresponds to the optimal value of the force according to the expert experience; and the dotted-red lines denote the boundary of the approximate force range. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

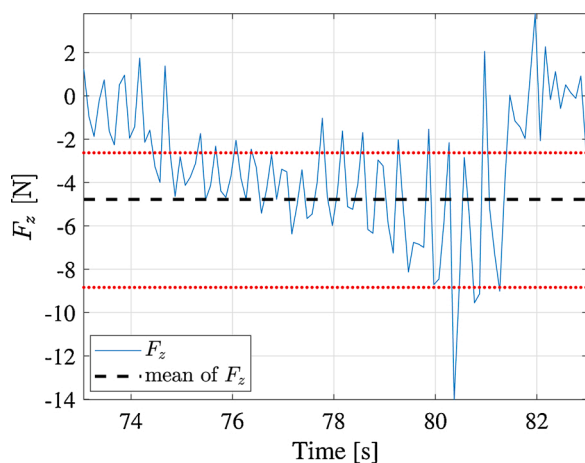


Fig. 20. Force exerted by the robot tool on the surface in the first experiment: the solid-blue line corresponds to the force values; the dashed-black line corresponds to the mean value of the force; and the dashed-red lines denote the boundary of the approximate force range. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

of this work contacted an expert in car surface repair tasks with extensive experience in several well-known companies in the automotive industry. Fig. 18 shows the setup considered for measuring the forces exerted by the expert with the tool on the workpiece while performing the surface treatment task, which consisted of the same industrial polishing tool, polishing disc and F/T sensor detailed in Section 5.1 together with a sample surface attached to the F/T sensor.

Fig. 19 shows the result of the expert performance with the industrial polishing tool during the treatment task. Note that the mean value of the force exerted on the workpiece surface was around 7 N and the maximum value was around 17 N. Note that, during the task, the expert sought to exert a force to about 5 N, which was the optimal value according to his experience. In addition, it is worthy to note that the size of the approximate force range was around 9 N.

Fig. 20 shows the result of the force exerted by the robot tool on the workpiece surface in the first experiment above, concretely for the

surface treatment of the second goal point, see Fig. 8g. The mean value of the force exerted on the workpiece surface was around 4.7 N, which is very similar to the optimal value indicated by the expert for this surface treatment task. Moreover, the maximum force value was around 14 N, which is smaller than that exerted by the expert. Furthermore, the size of the approximate force range was around 6 N, which is also smaller than that produced by the expert.

Therefore, it can be concluded that the proposed method generates appropriate force values for the surface treatment task, i.e., the *force accuracy* of proposed approach is good enough for the task at hand.

It is worth noting that, although the proposed application uses robotic assistance, the task execution is ultimately under the worker's control, which implies that it is susceptible to unconscious manipulation. Note that this situation is also present in the original manual approach, where no robotic assistance is considered.

However, in order to facilitate the robot teleoperation to the worker, the proposed AV-based user interface (see Section 3.2) allows the worker to zoom in on a specific area of the workpiece surface in order to get better visibility of this area. Moreover, the movement of the robot is also scaled according to the zoom performed, which allows the worker to teleoperate the robot more accurately along this area. Note that this accuracy ultimately depends on the hardware used, i.e., the haptic device, the robot arm, etc. Therefore, depending on the accuracy required for the specific task at hand, the devices of the application have to be properly selected from the market [59,60] to meet the required specifications.

This work assumed that the worker cannot be removed from the task due to its complexity, as is the case in repair of defects of car body surfaces. In fact, nowadays, there is no automatic system in the automotive industry able to perform this task successfully. Therefore, this work proposes for this task the cooperation between human and robot, i.e., the robot system assists the worker during the task in order to improve security, comfort and productivity.

However, it is important to remark that, in case that the human worker is dispensable (e.g., there are no uncertainties in the repair task), the proposed approach does not exclude a completely automatic operation. In fact, this completely automatic approach can be easily obtained by replacing the teleoperation performed by the user in the local workspace, see Fig. 1, with a trajectory generator that provides the tool position reference \bar{p}_{ref} for Level 3 of the robot high-level controller, see Fig. 4 and Section 4. Even more, the proposed approach could simultaneously combine worker's teleoperation and automatic operation. That is, the worker can teleoperate the robot tool to treat arbitrary areas and, when the teleoperation becomes inactive, the robot goes into an automatic mode to treat prior established areas. Details omitted for brevity.

Even though the proposed advanced robot teleoperation approach has been applied to car body surface polishing tasks, it is worthy to remark that it can be applied to other surface treatment applications. For instance, the process of taping (i.e., covering objects with masking tapes) has been recently addressed with a robot teleoperation approach in [29]. Also, a robot teleoperation solution was proposed in [31] for the small hole-cleaning process under object positioning uncertainties. Furthermore, robot teleoperation has also been proposed in [28] for performing maintenance tasks in the oil and gas well-pads stations. All these applications have in common the complexity of being fully automatized due to uncertainties, unstructured environments, etc. Thus, the synergistic effect obtained with the proposed method in the specific application at hand could also be used to improve the performance of other robot teleoperation applications, like those mentioned above.

7. Conclusions

An advanced teleoperation and control system for industrial robots was developed in this work in order to assist the human operator to conduct industrial operations such as finishing, sanding, deburring,

grinding, etc. on the surface of a workpiece. For this purpose, augmented virtuality (i.e., a virtual world that includes non-modeled real-world data) and haptic feedback were used to provide the user an immersive virtual experience when remotely teleoperating the robot system in order to properly perform the task.

The main advantages of the proposal are twofold. First, the proposed remote teleoperation is useful to preserve the physical integrity of human workers or to prevent the exclusion of persons with motor disabilities from the labor market. Second, the proposed robot control application has a positive synergistic effect: the human user provides flexibility to adapt to complex operations; whereas the controlled robot system provides precision and strength. Although the method was illustrated with a surface treatment operation, it can be easily extended to other industrial tasks where the human operator may benefit from robotic assistance.

The effectiveness and feasibility of the proposed approach was shown with several experiments using a 6R robotic arm. In addition, a comparison of the performance obtained manually by an expert and that obtained with the proposed method was also conducted to show the suitability of the proposed approach.

Declaration of Competing Interest

The authors report no declarations of interest.

Acknowledgements

This research was funded by the Spanish Government and the Generalitat Valenciana under Grants PID2020-117421RB-C21 and GV/2021/005.

Conflict of interest: None declared.

References

- Hoglund T, Alander J, Mantere T. A survey of telerobotic surface finishing. *Open Eng* 2018;8:156–61.
- Tornero J, Armesto L, Mora MC, Montés N, Herráez A, Asensio J. Detección de defectos en carrocerías de vehículos basado en visión artificial: Dise no e implantación. *Rev Iberoam Autom Inform Ind* 2012;9:93–104.
- Roboticom. Robotic systems for polishing complex forms. 2019 [accessed 27 October 2020], <https://www.roboticom.it/>.
- Ferrobotics. Grinding, polishing, deburring with ferrobotics. 2020. [https://www.ferrobotics.com/en/technologie-produkte/anwendungen/automotive/schleifen/](https://www.ferrobotics.com/en/technologie-produkte/anwendungen/automotive/schleifen/https://www.ferrobotics.com/en/technologie-produkte/anwendungen/automotive/schleifen/) [accessed 27 October 2020].
- Kieselbach KK, Nöthen M, Heuer H. Development of a visual inspection system and the corresponding algorithm for the detection and subsequent classification of paint defects on car bodies in the automotive industry. *J Coat Technol Res* 2019;16:1033–42.
- Melchiorri C. Robot teleoperation. London: Springer; 2013. p. 1–14.
- Sheridan TB. Human-robot interaction: status and challenges. *Human Factors* 2016;58:525–32. PMID: 27098262.
- Oliff H, Liu Y, Kumar M, Williams M, Ryan M. Reinforcement learning for facilitating human-robot-interaction in manufacturing. *J Manuf Syst* 2020;56:326–40.
- Ong KW, Seet G, Sim SK. An implementation of seamless human-robot interaction for telerobotics. *Int J Adv Robot Syst* 2008;5:18.
- Lui KY, Cho H, Ha C, Lee D. First-person view semi-autonomous teleoperation of cooperative wheeled mobile robots with visuo-haptic feedback. *Int J Robot Res* 2017;36:840–60.
- Liu L, Zhang Y, Liu G, Xu W. Variable motion mapping to enhance stiffness discrimination and identification in robot hand teleoperation. *Robot Comput Integr Manuf* 2018;51:202–8.
- Musić S, Salvietti G, Dohmann PBG, Chinello F, Prattichizzo D, Hirche S. Human-robot team interaction through wearable haptics for cooperative manipulation. *IEEE Trans Haptics* 2019;19:50–62.
- He X, Chen Y. Haptic-aided robot path planning based on virtual tele-operation. *Robot Comput Integr Manuf* 2009;25:792–803.
- Lin H-I. Design of an intelligent robotic precise assembly system for rapid teaching and admittance control. *Robot Comput Integr Manuf* 2020;64:101946.
- Bekele MK, Pierdicca R, Frontoni E, Malinverni ES, Gain J. A survey of augmented, virtual, and mixed reality for cultural heritage. *J Comput Cult Herit* 2018;11. 7: 1–36.
- Vignali G, Bertolini M, Bottani E, Donato LD, Ferraro A, Longo F. Design and testing of an augmented reality solution to enhance operator safety in the food industry. *Int J Food Eng* 2017;14:1–16.
- Masoni R, Ferrise F, Bordegoni M, Gattullo M, Uva AE, Fiorentino M, et al. Supporting remote maintenance in industry 4.0 through augmented reality. *Proc Manuf* 2017;11:1296–302. 27th international conference on flexible automation and intelligent manufacturing, FAIM2017, 27–30 June 2017, Modena, Italy.
- Uva AE, Gattullo M, Manghisi VM, Spagnolo D, Cascella GL, Fiorentino M. Evaluating the effectiveness of spatial augmented reality in smart manufacturing: a solution for manual working stations. *Int J Adv Manuf Technol* 2018;94:509–21.
- Gattullo M, Scurati GW, Fiorentino M, Uva AE, Ferrise F, Bordegoni M. Towards augmented reality manuals for industry 4.0: a methodology. *Robot Comput-Integr Manuf* 2019;56:276–86.
- Neves J, Serrario D, Pires JN. Application of mixed reality in robot manipulator programming. *Ind Robot: Int J Robot Res Appl* 2018;45:784–93.
- Milgram P, Takemura H, Utsumi A, Kishino F. Augmented reality: a class of displays on the reality-virtuality continuum. *Telemanipulator and telepresence technologies*, vol. 2351 1995:282–92.
- Lai Z-H, Tao W, Leu MC, Yin Z. Smart augmented reality instructional system for mechanical assembly towards worker-centered intelligent manufacturing. *J Manuf Syst* 2020;55:69–81.
- Baroroh DK, Chu C-H, Wang L. Systematic literature review on augmented reality in smart manufacturing: collaboration between human and computational intelligence. *J Manuf Syst* 2020. <https://doi.org/10.1016/j.jmsy.2020.10.017> (In press).
- Mawson VJ, Hughes BR. The development of modelling tools to improve energy efficiency in manufacturing processes and systems. *J Manuf Syst* 2019;51:95–105.
- Guo Z, Zhou D, Zhou Q, Zhang X, Geng J, Zeng S, et al. Applications of virtual reality in maintenance during the industrial product lifecycle: a systematic review. *J Manuf Syst* 2020;56:525–38.
- Kebria PM, Khosravi A, Nahavandi S, Shi P, Alizadehsani R. Robust adaptive control scheme for teleoperation systems with delay and uncertainties. *IEEE Trans Cybern* 2020;50:3243–53.
- Liu Y, Zhang Y. Toward welding robot with human knowledge: a remotely-controlled approach. *IEEE Trans Autom Sci Eng* 2015;12:769–74.
- Caiza G, Garcia CA, Naranjo JE, Garcia MV. Flexible robotic teleoperation architecture for intelligent oil fields. *Heliyon* 2020;6:e03833.
- Yuan Q, Weng C-Y, Suárez-Ruiz F, Chen I-M. Flexible telemanipulation based handy robot teaching on tape masking with complex geometry. *Robot Comput-Integr Manuf* 2020;66:101990.
- Liu H, Wang L. Remote human-robot collaboration: a cyber-physical system application for hazard manufacturing environment. *J Manuf Syst* 2020;54:24–34.
- Owan P, Garbini J, Devasia S. Faster confined space manufacturing teleoperation through dynamic autonomy with task dynamics imitation learning. *IEEE Robot Autom Lett* 2020;5:2357–64.
- Girbes-Juan V, Schettino V, Demiris Y, Tornero J. Haptic and visual feedback assistance for dual-arm robot teleoperation in surface conditioning tasks. *IEEE Trans Haptics* 2020. <https://doi.org/10.1109/TOH.2020.3004388> (in press).
- Cui L, Wang H, Liang X, Wang J, Chen W. Visual servoing of a flexible aerial refueling boom with an eye-in-hand camera. *IEEE Trans Syst Man Cybern: Syst* 2020. <https://doi.org/10.1109/TSMC.2019.2957992> (in press).
- Taryudi, Wang M-S. Eye to hand calibration using anis for stereo vision-based object manipulation system. *Microsyst Technol* 2018;24:305–17.
- Lai S-H, Fu C-W, Chang S. A generalized depth estimation algorithm with a single image. *IEEE Trans Pattern Anal Mach Intell* 1992;14:405–11.
- Kent D, Saldanha C, Chernova S. Leveraging depth data in remote robot teleoperation interfaces for general object manipulation. *Int J Robot Res* 2020;39:39–53.
- Kim J, Ham Y, Chung Y, Chi S. Systematic camera placement framework for operation-level visual monitoring on construction jobsites. *J Constr Eng Manag* 2019;145. 04019019-1-14.
- Gong Y, Seibel EJ. Feature-based three-dimensional registration for repetitive geometry in machine vision. *J Inform Technol Softw Eng* 2016;6. 184-1–5.
- He Y, Liang B, Yang J, Li S, He J. An iterative closest points algorithm for registration of 3D laser scanner point clouds with geometric features. *Sensors* 2017; 17. 1862-1–16.
- Angeles J. Fundamentals of robotic mechanical systems: theory, methods, and algorithms. 3rd ed. New York, NJ: Springer-Verlag; 2007.
- Lee W, Lee Y, Park G, Hong S, Kang Y. A whole-body rescue motion control with task-priority strategy for a rescue robot. *Auton Robots* 2017;41:243–58.
- Flacco F, De Luca A, Khatib O. Prioritized multi-task motion control of redundant robots under hard joint constraints. *IEEE/RSJ international conference on intelligent robots and systems* 2012:3970–7.
- Golub G, Kahan W. Calculating the singular values and pseudo-inverse of a matrix. *J Soc Ind Appl Math Ser B: Numer Anal* 1965;2:205–24.
- Utkin V, Guldner J, Shi J. Sliding mode control in electro-mechanical systems. 2nd ed. London: Taylor & Francis; 2009.
- Baek J, Jin M, Han S. A new adaptive sliding-mode control scheme for application to robot manipulators. *IEEE Trans Ind Electron* 2016;63:3628–37.
- Lee J, Chang PH, Jin M. Adaptive integral sliding mode control with time-delay estimation for robot manipulators. *IEEE Trans Ind Electron* 2017;64:6796–804.
- Gracia L, Solanes JE, Muñoz-Benavent P, Valls Miro J, Perez-Vidal C, Tornero J. Human-robot collaboration for surface treatment tasks. *Interact Stud* 2019;20: 148–84.
- Teklemariam HG, Das AK. A case study of phantom omni force feedback device for virtual product design. *Int J Interact Des Manuf* 2017;11:881–92.
- Li Z, Huang B, Ye Z, Deng M, Yang C. Physical human-robot interaction of a robotic exoskeleton by admittance control. *IEEE Trans Ind Electron* 2018;65:9614–24.

- [50] Kuka. Technical specifications of KUKA products. 2021 [accessed 22 January 2021], <https://www.kuka.com/en-be/services/downloads?terms=Language:en:1&q=>.
- [51] Li S, Ghasemi A, Xie W-F, Gao Y. Sliding mode control (SMC) of image-based visual servoing for a 6dof manipulator, recent developments in sliding mode control theory and applications. *IntechOpen* 2018;16:844–55.
- [52] Siciliano B, Khatib OE. *Handbook of robotics*. London, UK: Springer-Verlag; 2008.
- [53] Muñoz A, Mahiques X, Solanes JE, Martí A, Gracia L, Tornero J. Mixed reality-based user interface for quality control inspection of car body surfaces. *J Manuf Syst* 2019;53:75–92.
- [54] Micro-Epsilon. Fully automatic surface inspection of painted car bodies. 2018. <https://www.micro-epsilon.com/measurement-systems/Paint-Inspection/karosserie/>.
- [55] Stavropoulos P, Papacharalampopoulos A, Petridis D. A vision-based system for real-time defect detection: a rubber compound part case study. *Proc CIRP* 2020;93:1230–5.
- [56] Molina J, Solanes JE, Arnal L, Tornero J. On the detection of defects on specular car body surfaces. *Robot Comput Integr Manuf* 2017;48:263–78.
- [57] Arnal L, Solanes JE, Molina J, Tornero J. Detecting dings and dents on specular car body surfaces based on optical flow. *J Manuf Syst* 2017;45:306–21.
- [58] Fondazione Istituto Italiano di Tecnologia. Force/torque sensors. 2015 [accessed 4 April 2020], <https://www.iit.it/technology-transfer-docs/509-new-technology-taser-forcatorque-sensor-z-1/file>.
- [59] FAULHABER. Faulhaber Haptic Device. 2021 [accessed 22 January 2021], <https://www.faulhaber.com/en/markets/factory-automation-robotics/haptic-device/>.
- [60] DirectInd. Direct Industry: high-precision robots. 2021 [accessed 22 January 2021], <https://www.directindustry.com/industrial-manufacturer/high-precision-robot-118858.html>.