IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 3, 2020, accepted July 21, 2020, date of publication August 5, 2020, date of current version September 1, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3014501

Human-Robot Cooperation for Surface Repair Combining Automatic and Manual Modes

ALBERTO GARCÍA[1], VICENT GIRBÉS-JUAN[2], J. ERNESTO SOLANES[1], LUIS GRACIA[1], CARLOS PEREZ-VIDAL[3], AND JOSEP TORNERO[1]

[1]Instituto de Diseño y Fabricación, Universitat Politècnica de València, 46022 Valencia, Spain
[2]Departament d'Enginyeria Electrònica, Universitat de València, 46100 Burjassot, Spain
[3]Departamento de Ingeniería de Sistemas y Automática, Universidad Miguel Hernández, 03202 Elche, Spain

Corresponding author: J. Ernesto Solanes (esolanes@idf.upv.es)

This work was supported in part by the Spanish Government, and in part by the Generalitat Valenciana under Grant DPI2017-87656-C2-1-R and Grant ACIF/2019/007.

ABSTRACT This article develops a human-robot cooperation to carry out treatments such as sanding, polishing, etc. on the surface of a known rigid object. For this purpose, a vision system is considered to get the object location to ensure not only the perpendicularity of the robot tool to the object surface but also a smooth approach of the tool to the surface. In order to add flexibility, the proposal includes the simultaneous combination of automatic and manual modes of operation. Thus, the human user can guide the robot tool to treat arbitrary areas (manual mode) and, when the operator releases the tool, the robot goes into the automatic mode to treat prior established areas. The method uses a task prioritization framework and three types of controllers: an admittance controller for the tool guidance; a hybrid controller to modify the tool orientation and, in the automatic mode, the tool position; and a sliding mode controller to limit the velocity at which the tool approaches the object surface. The applicability and efficacy of the proposed method is demonstrated experimentally using a conventional 6R robot arm.

INDEX TERMS Cooperative sanding, robot cooperation, vision system.

I. INTRODUCTION

A. MOTIVATION

It is common knowledge that the manufacturing industry is becoming increasingly automated. This general tendency makes industrial production faster, more precise and more efficient, and it also creates the potential to replace human operators by machines in the most tedious, hard and dangerous tasks. Moreover, the new wave is not just seeking to replace the human operator by a robot, but rather allowing them to cooperate and collaborate in order to obtain the best from both, i.e., the adaptability of the human and the strength and accuracy of the robot, which opens new perspectives in the field of the manufacturing industry.

One of the key factors in this sector is quality control, i.e., the detection and correction of anomalies of the product. Specifically, quality control of surfaces is a critical process for the manufacturing sector since, at the very least, it affects the customer's perception of the product. However, this process remains one of the least automated ones in the manufacturing industry [1]. In particular, whereas there are industrial solutions for detecting surface defects using machine vision systems [2], the elimination of surface defects by surface treatment is still largely depending on manual labor and the personal skills of the human operator.

B. LITERATURE REVIEW

The complete or partial automation of surface treatment tasks, such as polishing, sanding or deburring, has been the main focus of a growing number of research contributions in the last years. For instance, some works addressed the automation of surface treatment using robotic systems [3], [4], whereas other works tackled some specific issues, such as detecting wether a polishing operation is complete or the polishing tool needs to be changed [5]. Furhtermore, [6] proposed a human-robot collaboration strategy so that the user guides the robot to perform the treatment on a specific area.

Note that, to properly apply the surface treatment, not just contact, but also perpendicularity between the tool and the surface must be kept [7].

The associate editor coordinating the review of this manuscript and approving it for publication was Bohui Wang.

Moreover, the use of machine vision systems in robotic applications with an industrial perspective has been extensively carried out, for it gives the robot controller feedback about its environment, providing accuracy and flexibility. On the one hand, some applications use eye-in-hand configurations (i.e., camera attached to the robot end-effector), as in [8], where computer vision is used to improve the accuracy of a welding operation, or in [9], where the tool of the robot is automatically changed thanks to an image-based visual servoing. On the other hand, other applications use eye-to-hand configurations, i.e., the vision system is attached to some rigid structure and observing the robot workspace, so as to localize pieces to perform operations such as pick and place [10], or to keep track of the environment of the robot for safety purposes [11].

The visual data can be obtained using 2D cameras, estimating depth by processing the images [8]–[10], or can be directly read from 3D sensors [11], such as Microsoft Kinect.

Registration is the process of getting the transformation needed to express two views of an object under the same coordinate system [12]. This is typically used to get the location (position and orientation) of an object by matching the 3D sensor data from the real object (point cloud) with a 3D virtual model, as considered in this work.

Regarding the control strategy, Sliding Mode Control (SMC) provides the key advantages of computational efficiency and robustness [13], so it has been extensively used for robotic applications. For instance, in [14] SMC is combined with fuzzy control theory to control a robotic shoulder, whereas in [15] an adaptive backstepping based SMC is used to control a robot manipulator with position commands.

C. PROPOSAL

As mentioned above, previous works [3] developed the automation of surface treatment tasks using a robot system with force feedback in order to control both the contact and the orthogonality between the robot tool and the treated surface. In contrast, the approach proposed in this study considers a computer vision system to get the position and orientation of the object to be treated in order to ensure not only the orthogonality between the tool and the surface but also to guarantee a smooth approach of the tool to the surface.

Although the object deformations, changes in its stiffness or even sudden movements are not so easy to detect with a computer vision system as they are with force-torque data, for the scope of this work the advantages of a computer vision system outweigh its drawbacks, as long as the treatment task is assumed to be performed on a known rigid object. In particular, a major advantage of using a vision system instead of a force-torque sensor is that it allows to control also the phase where there is no contact between the tool and the treated object, e.g., to get a smooth approach to the object surface as mentioned above.

In order to add flexibility to the application, the proposal also includes the simultaneous combination of automatic and manual modes of operation. In particular, the human user can guide the robot tool to treat arbitrary points or areas (manual mode) and, when the operator releases the tool, the robot goes into the automatic mode to treat prior established points or areas.

The control strategy used in this work includes a task prioritization framework and three types of controllers, as discussed below.

Firstly, for the manual mode of operation, an admittance/impedance controller is considered in order to allow the user to guide the tool using a force-torque sensor placed at the end-effector of the robot. This type of control is a common way of using force-torque data to control the movement of robot manipulators [16].

Secondly, a hybrid controller is used to ensure the orthogonality between the robot tool and the treated surface, so that it remains aligned with the normal vector of the closest point of the surface. This normal vector is obtained from the matched virtual model of the treated object, i.e., the desired orientation is known beforehand, and regardless of the distance from the robot tool to the surface. Furthermore, for the automatic mode of operation, another hybrid controller is used to modify the tool position in order to perform the treatment on the prior established points or areas.

Thirdly, a non-conventional SMC is also developed to limit the velocity at which the tool approaches the object surface, tending to zero at the moment of contact, thus avoiding excessive pressure on the surface while applying the treatment as well. This non-conventional SMC is only active when the tool is advancing towards the surface at high speed, and it remains inactive both when the tool moves away from the surface or when it moves to the surface at low speed. Hence, the operator can freely and quickly move the tool away from the object surface. The speed at which the tool approaches the surface is also obtained from the data of the matched virtual model.

D. STRUCTURE OF THE PAPER

The content of the paper is as follows. Section II introduces some background theory used in this work. Next, Section III develops the proposed controller for the target application, while its implementation is presented in Section IV. Then, the applicability and efficacy of the method is shown in Section V with real experimentation. Lastly, some conclusions are drawn in Section VI.

II. BACKGOUND THEORY

A. ROBOT KINEMATIC EQUATIONS

The robot kinematics and its derivatives (velocity and acceleration) are given by:

$$\mathbf{p} = \mathbf{l}(\mathbf{q}) \tag{1}$$

$$\dot{\mathbf{p}} = \frac{\partial \mathbf{l}(\mathbf{q})}{\partial \mathbf{q}}\dot{\mathbf{q}} = \mathbf{J}\dot{\mathbf{q}} \tag{2}$$

$$\ddot{\mathbf{p}} = \mathbf{J}\ddot{\mathbf{q}} + \dot{\mathbf{J}}\dot{\mathbf{q}}, \tag{3}$$

where vector $\mathbf{p} = \begin{bmatrix} x & y & z & \alpha & \beta & \gamma \end{bmatrix}^{\mathrm{T}}$ is the robot pose (orientation is given by roll α, pitch β and yaw γ angles), vector

$\mathbf{q} = \begin{bmatrix} q_1 & \cdots & q_n \end{bmatrix}^{\mathrm{T}}$ is the robot configuration, n is the number of robot joints, vector $\mathbf{l}(\mathbf{q})$ is the robot kinematic function and matrix $\mathbf{J}$ is the robot Jacobian [17].

B. LOW-LEVEL CONTROLLER

This study assumes that the low-level controller developed by the robot manufacturer is able to achieve a particular joint acceleration $\ddot{\mathbf{q}}$ from the commanded acceleration $\ddot{\mathbf{q}}_c$ with negligible (fast) dynamics. Notice that this low-level controller should take into account the robot dynamic model.

C. COMPUTER VISION SYSTEM

In this work, three 3D sensors (Kinect cameras) are used, so that the interference problem of 3D sensors working in the same environment can be easily overcome [11], whereas they allow to cover a large area without placing them far from the object to be treated.

The registration used in this work to locate the object to be treated is based on the so-called Iterative Closest Points (ICP) [12], which is one of the most extensively used methodologies for its simplicity and effectiveness. In this sense, two conditions are needed to ensure a good performance of the ICP algorithm: the initial pose of the 3D virtual model and the point cloud have to be relatively similar, so that the algorithm does not get stuck in local minima; and the point cloud has to be adequately filtered, so it does not have outliers. In particular, both conditions can be easily met in the context of this work.

D. TASK PRIORITIZATION FRAMEWORK

This framework allows to simultaneously tackle several goals or tasks with different priority levels [18]. Hence, lower-priority tasks are accomplished by utilizing the remaining degrees of freedom, i.e., those not used by higher-priority tasks. In case that an exact solution is not possible, the tasks errors are minimized. The formulation for this method is as follows [19]:

$$\mathbf{A}_i\bar{\mathbf{x}} = \mathbf{b}_i, \tag{4}$$

$$\bar{\mathbf{x}}_i = \bar{\mathbf{x}}_{i-1} + (\mathbf{A}_i\mathbf{N}_{i-1})^{\dagger}(\mathbf{b}_i - \mathbf{A}_i\bar{\mathbf{x}}_{i-1}), \tag{5}$$

$$\mathbf{N}_i = \mathbf{N}_{i-1}(\mathbf{I} - (\mathbf{A}_i\mathbf{N}_{i-1})^{\dagger}(\mathbf{A}_i\mathbf{N}_{i-1})), \tag{6}$$

where M tasks or equalities are considered in (4), which are given by matrix $\mathbf{A}_i$ and vector $\mathbf{b}_i$ ($i = 1$ and $i = M$ denote the highest and lowest priority, respectively), and the solution $\bar{\mathbf{x}}_M$ is computed with the recursive formulation given by (5) and (6) in order to minimize the tasks errors, where $\bar{\mathbf{x}}_0 = \mathbf{0}$ (zero column vector), $\mathbf{N}_0 = \mathbf{I}$ (identity matrix) and superscript † represents the Moore-Penrose pseudoinverse [20].

E. NON-CONVENTIONAL SLIDING MODE CONTROL

This work uses the non-conventional SMC described in [3] to limit the velocity at which the robot tool approaches the object surface. This control allows to fulfill *inequality constraints* as detailed in Theorem 1.

Theorem 1: Consider the state equation given by:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{d}) + \mathbf{g}(\mathbf{x})\,\mathbf{u}, \tag{7}$$

where $\mathbf{x}(t)$ *is the state vector,* $\mathbf{d}(t)$ *is the disturbance vector,* $\mathbf{u}(t)$ *is the control input vector,* $\mathbf{f}$ *is the drift vector field and* $\mathbf{g} = [\mathbf{g}_1 \ \ldots \ \mathbf{g}_{n_u}]$*, is a set of control vector fields.*

Additionally, it is considered that $\mathbf{x}$ *is subject to* inequality *constraints* $\phi_{in,i}(\mathbf{x}) \leq 0$, $i = 1, \ldots, N_{in}$*, where* $\phi_{in,i}(\mathbf{x})$ *is the i-th constraint function. Therefore, the space compatible with these constraints is given by:*

$$\Phi_{in} = \left\{\mathbf{x} \mid \phi_{in,i}(\mathbf{x}) \leq 0\right\}, \tag{8}$$

with $i = 1, \ldots, N_{in}$.

Thus, considering that functions $\phi_{in,i}$ *are differentiable, the control action* $\mathbf{u}$ *that satisfies the control equation below ensures that the system converges to* Φ_{in} *in finite time and remains there henceforth:*

$$v2dm\left(pos\left(\boldsymbol{\phi}_{in}\right)\right)\mathbf{L_g}\boldsymbol{\phi}_{in}\mathbf{u} = -\mathbf{W}_{in}\,pos\left(\boldsymbol{\phi}_{in}\right)\,u_{in}^{+} \tag{9}$$

$$u_{in}^{+} > \sum_{i=1}^{N_a}\left(\max(L_f\phi_{in,i}^{a}, 0)\right)/diag_{\min}(\mathbf{W}_{in}), \tag{10}$$

where function $v2dm(\cdot)$ *converts a vector into a diagonal matrix, function* $pos(\cdot)$ *represents the* positive function *(i.e.,* $pos(x)$ *is equal to* 0 *if* $x \leq 0$ *and equal to* 1 *if* $x > 0$*),* $\boldsymbol{\phi}_{in}$ *is a column vector with all the inequality constraint functions* $\phi_{in,i}$*, matrix* $\mathbf{L_g}\boldsymbol{\phi}_{in}$ *contains the row vectors* $\mathbf{L_g}\phi_{in,i}$ *of all inequality constraints, the scalar* $L_f\phi_{in,i} = \frac{\partial\phi_{in,i}^{T}}{\partial\mathbf{x}}\mathbf{f}$ *represents the Lie derivative in the direction of* $\mathbf{f}$*, the row vector* $\mathbf{L_g}\phi_{in,i} = \frac{\partial\phi_{in,i}^{T}}{\partial\mathbf{x}}\mathbf{g}$ *represents the Lie derivative in the direction of* $\mathbf{g}$*, scalar* u_{in}^{+} *is the so-called switching gain,* $\mathbf{W}_{in}$ *is a weighting diagonal matrix and right superscript a denotes the active constraints (that is,* $\phi_{in,i}^{a} \geq 0$*), whose number is equal to* N_a.

For the proof of the above theorem, and further details about the non-conventional SMC, see [3].

III. PROPOSED APPROACH

A. CONTROL SCHEME

The control scheme of the proposed approach is shown in Fig. 1, where three tasks with different priority levels are depicted. Firstly, the highest priority task (Level 1) is used to maintain the orthogonality between the robot tool and the object surface by means of a hybrid controller. Secondly, the medium-priority task (Level 2) is used to limit the velocity at which the tool approaches the object surface by means of the non-conventional SMC described in Section II-E. Finally, the task with the lowest priority (Level 3) is employed to treat a specific point of the object surface with the robot tool, either using an admittance controller with force feedback (manual mode) or a hybrid controller to track a reference Cartesian position (automatic mode).

The input to the tasks are: the robot angles and velocities $\{\mathbf{q}, \dot{\mathbf{q}}\}$ and the pose vector of the tool $\mathbf{p}$ (all three levels), which are provided by the low-level controller of the

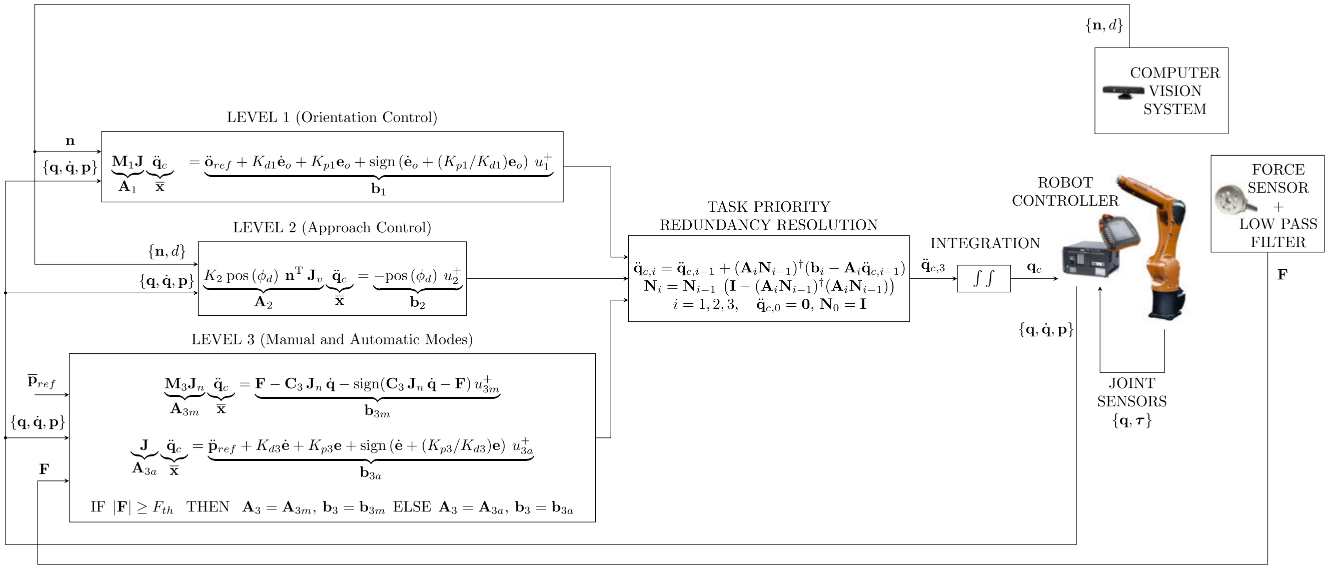

$\{\mathbf{n}, d\}$
COMPUTER VISION SYSTEM
LEVEL 1 (Orientation Control)
$\mathbf{n}$
$\{\mathbf{q}, \dot{\mathbf{q}}, \mathbf{p}\}$
$\underbrace{\mathbf{M}_1\mathbf{J}}_{\mathbf{A}_1}\underbrace{\ddot{\mathbf{q}}_c}_{\overline{\mathbf{x}}} = \underbrace{\ddot{\mathbf{o}}_{ref} + K_{d1}\dot{\mathbf{e}}_o + K_{p1}\mathbf{e}_o + \mathrm{sign}\,(\dot{\mathbf{e}}_o + (K_{p1}/K_{d1})\mathbf{e}_o)\ u_1^+}_{\mathbf{b}_1}$
LEVEL 2 (Approach Control)
$\{\mathbf{n}, d\}$
$\{\mathbf{q}, \dot{\mathbf{q}}, \mathbf{p}\}$
$\underbrace{K_2\,\mathrm{pos}\,(\phi_d)\ \mathbf{n}^{\mathrm{T}}\,\mathbf{J}_v}_{\mathbf{A}_2}\underbrace{\ddot{\mathbf{q}}_c}_{\overline{\mathbf{x}}} = \underbrace{-\mathrm{pos}\,(\phi_d)\ u_2^+}_{\mathbf{b}_2}$
LEVEL 3 (Manual and Automatic Modes)
$\overline{\mathbf{p}}_{ref}$
$\{\mathbf{q}, \dot{\mathbf{q}}, \mathbf{p}\}$
$\mathbf{F}$
$\underbrace{\mathbf{M}_3\mathbf{J}_n}_{\mathbf{A}_{3m}}\underbrace{\ddot{\mathbf{q}}_c}_{\overline{\mathbf{x}}} = \underbrace{\mathbf{F} - \mathbf{C}_3\,\mathbf{J}_n\,\dot{\mathbf{q}} - \mathrm{sign}(\mathbf{C}_3\,\mathbf{J}_n\,\dot{\mathbf{q}} - \mathbf{F})\,u_{3m}^+}_{\mathbf{b}_{3m}}$
$\underbrace{\mathbf{J}}_{\mathbf{A}_{3a}}\underbrace{\ddot{\mathbf{q}}_c}_{\overline{\mathbf{x}}} = \underbrace{\ddot{\mathbf{p}}_{ref} + K_{d3}\dot{\mathbf{e}} + K_{p3}\mathbf{e} + \mathrm{sign}\,(\dot{\mathbf{e}} + (K_{p3}/K_{d3})\mathbf{e})\ u_{3a}^+}_{\mathbf{b}_{3a}}$
IF $|\mathbf{F}| \geq F_{th}$ THEN $\mathbf{A}_3 = \mathbf{A}_{3m}$, $\mathbf{b}_3 = \mathbf{b}_{3m}$ ELSE $\mathbf{A}_3 = \mathbf{A}_{3a}$, $\mathbf{b}_3 = \mathbf{b}_{3a}$
TASK PRIORITY REDUNDANCY RESOLUTION
$\ddot{\mathbf{q}}_{c,i} = \ddot{\mathbf{q}}_{c,i-1} + (\mathbf{A}_i\mathbf{N}_{i-1})^\dagger(\mathbf{b}_i - \mathbf{A}_i\ddot{\mathbf{q}}_{c,i-1})$
$\mathbf{N}_i = \mathbf{N}_{i-1}\ \left(\mathbf{I} - (\mathbf{A}_i\mathbf{N}_{i-1})^\dagger(\mathbf{A}_i\mathbf{N}_{i-1})\right)$
$i = 1, 2, 3, \quad \ddot{\mathbf{q}}_{c,0} = \mathbf{0},\ \mathbf{N}_0 = \mathbf{I}$
INTEGRATION
$\ddot{\mathbf{q}}_{c,3}$
$\int\int$
$\mathbf{q}_c$
ROBOT CONTROLLER
$\{\mathbf{q}, \dot{\mathbf{q}}, \mathbf{p}\}$
JOINT SENSORS $\{\mathbf{q}, \boldsymbol{\tau}\}$
FORCE SENSOR + LOW PASS FILTER
$\mathbf{F}$

FIGURE 1. Control scheme of the proposal.

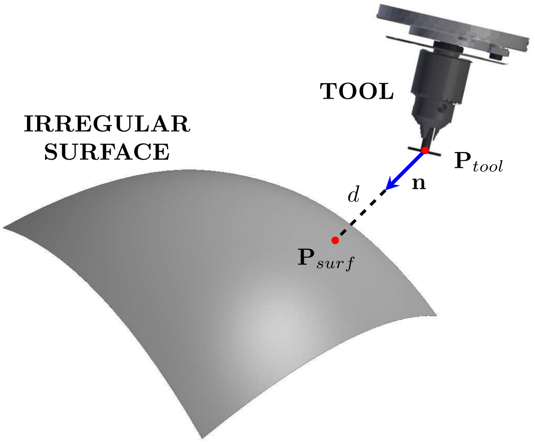


FIGURE 2. Representation of the data provided by the computer vision system.

robot; the vector $\mathbf{F}$ (Level 3) of forces and torques measured by the *guidance sensor* placed at the end-effector of the robot (it is assumed that the sensor electronics has filtered the measured signals); the reference Cartesian position $\overline{\mathbf{p}}_{ref} = \left[x_{ref}\ y_{ref}\ z_{ref}\right]^{\mathrm{T}}$ (Level 3) of the tool; and the distance d (Level 2) and the normal vector $\mathbf{n}$ (Level 1 and Level 2) obtained from the computer vision system, see Fig. 2. In particular, d is the distance from the tool tip $\mathbf{P}_{tool}$ to the closest point of the object surface, namely $\mathbf{P}_{surf}$. Whereas $\mathbf{n}$ is the vector from the tool tip to point $\mathbf{P}_{surf}$, which corresponds to the normal vector of the surface at point $\mathbf{P}_{surf}$ as long as it is defined (i.e., as long as the object surface is smooth at $\mathbf{P}_{surf}$).

Each task gives an equation $\mathbf{A}_i\overline{\mathbf{x}} = \mathbf{b}_i$ (4) whose error has to be minimized, where the unknown $\overline{\mathbf{x}}$ is the commanded acceleration $\ddot{\mathbf{q}}_c$. Hence, Eqs. (5) and (6) are employed to compute the acceleration $\ddot{\mathbf{q}}_{c,3}$, which in turn is integrated twice to obtain the angles $\mathbf{q}_c$ that are commanded to the robot. Lastly, the low-level controller of the robot closes a loop to track these angles considering the joint sensor measurements, i.e., the current values $\mathbf{q}$ and the torques $\boldsymbol{\tau}$. Hence, the inaccuracy of this low-level control loop is bounded and represented by $\mathbf{d}_c$.

B. LEVEL 1: ORIENTATION CONTROL TO ENSURE PERPENDICULARITY TO THE SURFACE

As commented previously, a key constraint in surface treatment tasks consists in keeping the tool perpendicular to the surface, i.e., the tool Z-axis must match vector $\mathbf{n}$, which is the vector from the tool tip $\mathbf{P}_{tool}$ to the closest point of the object surface $\mathbf{P}_{surf}$, as shown in Fig. 2. Therefore, vector $\mathbf{n}$ represents the reference orientation for the robot tool and can be readily converted [17] to roll α_{ref} and pitch β_{ref} reference values. Note that the yaw angle is not constrained in the application and can be used for other purposes, e.g., it can be guided by the user as considered in Section III-D1.

Therefore, the acceleration equality for this level is given by:

$$\mathbf{M}_1\mathbf{J}\ddot{\mathbf{q}}_c = \ddot{\mathbf{o}}_{ref} + K_{d1}\dot{\mathbf{e}}_o + K_{p1}\mathbf{e}_o + \mathrm{sign}\left(\dot{\mathbf{e}}_o + (K_{p1}/K_{d1})\mathbf{e}_o\right) u_1^+ \\ \rightarrow \mathbf{A}_1\ddot{\mathbf{q}}_c = \mathbf{b}_1, \qquad (11)$$

where matrix $\mathbf{M}_1 = \begin{bmatrix} 0\ 0\ 0\ 1\ 0\ 0 \\ 0\ 0\ 0\ 0\ 1\ 0 \end{bmatrix}$ is used to affect only the angular coordinates *roll* and *pitch* of the robot tool; vector $\mathbf{o}_{ref} = \left[\alpha_{ref}\ \beta_{ref}\right]^{\mathrm{T}}$ represents the reference orientation; vector $\mathbf{e}_o = \mathbf{o}_{ref} - \left[\alpha\ \beta\right]^{\mathrm{T}}$ is the tool orientation error; K_{p1} and K_{d1} are the gains to correct tool orientation error and its derivative, respectively; $\dot{\alpha}$ and $\dot{\beta}$ are computed from the first-order kinematics in (2); u_1^+ is the gain for the last switching term; and $\mathbf{A}_1$ and $\mathbf{b}_1$ are the matrix and vector for the first task in (4).

Note that (11) is a hybrid control due to the last switching term, which is introduced to compensate the term $\dot{\mathbf{J}}\dot{\mathbf{q}}$, see (3).

TABLE 1. Algorithm of the proposed method (executed at T_s seconds).

```
$[\mathbf{q}, \dot{\mathbf{q}}, \mathbf{F}, \mathbf{n}, d]$=GetRobot&Forces&VisionData;
$\mathbf{o}_{ref}$ =OrientationOfVector($\mathbf{n}$);
$\mathbf{p} = \mathbf{l}(\mathbf{q})$ ;                                   // Eq. (1)
$\dot{\mathbf{p}} = \mathbf{J}\dot{\mathbf{q}}$ ;                                    // Eq. (2)
$\phi_d = \epsilon_d - d + K_2\,\mathbf{n}^{\mathrm{T}}\,\mathbf{J}_v\,\dot{\mathbf{q}}$ ;   // Eqs. (12),(13)
$\dot{\mathbf{o}}_{ref} = (\mathbf{o}_{ref} - \mathbf{o}_{ref,prev})/T_s$ ;       // Derivative
$\ddot{\mathbf{o}}_{ref} = (\dot{\mathbf{o}}_{ref} - \dot{\mathbf{o}}_{ref,prev})/T_s$ ;       // Derivative
$\dot{\mathbf{p}}_{ref} = (\mathbf{p}_{ref} - \mathbf{p}_{ref,prev})/T_s$ ;       // Derivative
$\ddot{\mathbf{p}}_{ref} = (\dot{\mathbf{p}}_{ref} - \dot{\mathbf{p}}_{ref,prev})/T_s$ ;       // Derivative
$\mathbf{e}_o = (\mathbf{o}_{ref} - \mathbf{M}_1\mathbf{p})$ ;      // Orientation error
$\dot{\mathbf{e}}_o = (\dot{\mathbf{o}}_{ref} - \mathbf{M}_1\dot{\mathbf{p}})$ ;          // Derivative of orientation error
$\mathbf{e} = (\mathbf{p}_{ref} - \mathbf{p})$ ;                       // Pose error
$\dot{\mathbf{e}} = (\dot{\mathbf{p}}_{ref} - \dot{\mathbf{p}})$ ; // Derivative of pose error
$\mathbf{A}_1 = \mathbf{M}_1\mathbf{J}$ ;                                // Eq. (11)
$\mathbf{b}_1 = \ddot{\mathbf{o}}_{ref} + K_{d1}\dot{\mathbf{e}}_o + K_{p1}\mathbf{e}_o + \mathrm{sign}\,(\dot{\mathbf{e}}_o + (K_{p1}/K_{d1})\mathbf{e}_o)\,u_1^+$ ;   // Eq. (11)
$\mathbf{A}_2 = K_2\,\mathrm{pos}\,(\phi_d)\,\mathbf{n}^{\mathrm{T}}\,\mathbf{J}_v$ ;                // Eq. (15)
$\mathbf{b}_2 = -\mathrm{pos}\,(\phi_d)\;u_2^+$ ;                    // Eq. (15)
if $\mathbf{F} \geq F_{th}$ then
    $\mathbf{A}_3 = \mathbf{M}_3\,\mathbf{J}_n$ ;                           // Eq. (19)
    $\mathbf{b}_3 = \mathbf{F} - \mathbf{C}_3\,\mathbf{J}_n\,\dot{\mathbf{q}} - \mathrm{sign}(\mathbf{C}_3\,\mathbf{J}_n\,\dot{\mathbf{q}} - \mathbf{F})\,u_{3m}^+$ ; // Eq. (19)
else
    $\mathbf{A}_3 = \mathbf{J}$ ;                               // Eq. (20)
    $\mathbf{b}_3 = \ddot{\mathbf{p}}_{ref} + K_{d3}\dot{\mathbf{e}} + K_{p3}\mathbf{e} + \mathrm{sign}\,(\dot{\mathbf{e}} + (K_{p3}/K_{d3})\mathbf{e})\,u_{3a}^+$ ;        // Eq. (20)
end
$\ddot{\mathbf{q}}_{c,1} = \mathbf{A}_1^{\dagger}\mathbf{b}_1$ ;                      // Eq. (5), $i = 1$
$\mathbf{N}_1 = \mathbf{I} - \mathbf{A}_1^{\dagger}\mathbf{A}_1$ ;                  // Eq. (6), $i = 1$
$\ddot{\mathbf{q}}_{c,2} = \ddot{\mathbf{q}}_{c,1} + (\mathbf{A}_2\mathbf{N}_1)^{\dagger}(\mathbf{b}_2 - \mathbf{A}_2\ddot{\mathbf{q}}_{c,1})$ ; // Eq. (5), $i = 2$
$\mathbf{N}_2 = \mathbf{N}_1(\mathbf{I} - (\mathbf{A}_2\mathbf{N}_1)^{\dagger}(\mathbf{A}_2\mathbf{N}_1))$ ;     // Eq. (6), $i = 2$
$\ddot{\mathbf{q}}_{c,3} = \ddot{\mathbf{q}}_{c,2} + (\mathbf{A}_3\mathbf{N}_2)^{\dagger}(\mathbf{b}_3 - \mathbf{A}_3\ddot{\mathbf{q}}_{c,2})$ ; // Eq. (5), $i = 3$
SendToJointControllers($\ddot{\mathbf{q}}_{c,3}$);
$\mathbf{o}_{ref,prev} = \mathbf{o}_{ref}$ ;          // For next iteration
$\dot{\mathbf{o}}_{ref,prev} = \dot{\mathbf{o}}_{ref}$ ;          // For next iteration
$\mathbf{p}_{ref,prev} = \mathbf{p}_{ref}$ ;          // For next iteration
$\dot{\mathbf{p}}_{ref,prev} = \dot{\mathbf{p}}_{ref}$ ;          // For next iteration
```

This hybrid control has two main advantages: the derivative of the robot Jacobian is not required; and, due to the other continuous terms in the control action, the gain u_1^+ can be relatively small, reducing the chattering effects.

C. LEVEL 2: NON-CONVENTIONAL SMC TO CONTROL THE APPROACH TO THE SURFACE

The following inequality constraint is considered to progressively limit the velocity of approach of the tool tip to the object surface:

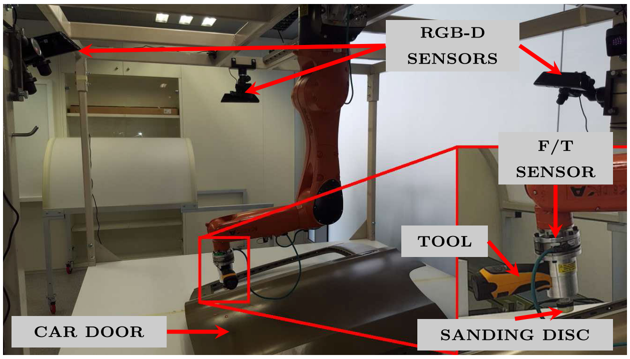


FIGURE 3. Setup used for the experiments: a 6R industrial manipulator with a force sensor, a network of three RGB-D sensors, a sanding tool and a car door.

$$\phi_d = \epsilon_d - d - K_2\dot{d} \leq 0, \tag{12}$$

where ϵ_d is a tolerance which provides a safety margin between the tool tip and the surface during the surface treatment task and parameter K_2 can be freely designed to establish the maximum allowed rate of approach to the object surface. Thus, the maximum rate of reduction of the distance given by the inequality (12) tends to zero as the tool reaches the surface.

Assuming that the object to be treated remains static or moves slowly compared to the robot motion, the time derivative of d for the last term in (12) can be computed from the joint velocities as follows:

$$\begin{aligned}\dot{d} &= (\partial d/\partial \mathbf{q})^{\mathrm{T}}\,\dot{\mathbf{q}} = \left((\partial \boldsymbol{P}_{tool}/\partial \mathbf{q})^{\mathrm{T}}\,(\partial d/\partial \mathbf{P}_{tool})\right)^{\mathrm{T}}\dot{\mathbf{q}}\\ &= \left(\mathbf{J}_v^{\mathrm{T}}(-\mathbf{n})\right)^{\mathrm{T}}\dot{\mathbf{q}} = -\mathbf{n}^{\mathrm{T}}\mathbf{J}_v\dot{\mathbf{q}},\end{aligned} \tag{13}$$

where matrix $\mathbf{J}_v$ contains the first three rows of the robot Jacobian $\mathbf{J}$.

The non-conventional SMC presented in Section II-E is considered to satisfy the constraint in (12). For this purpose, a dynamical system in the form of Eq. (7) is considered with $\mathbf{x} = \left[\mathbf{q}^{\mathrm{T}}\;\dot{\mathbf{q}}^{\mathrm{T}}\right]^{\mathrm{T}}$, $\mathbf{d} = \mathbf{d}_c$ and $\mathbf{u} = \ddot{\mathbf{q}}_c$. Thus, the model of the robot system is a double integrator given by:

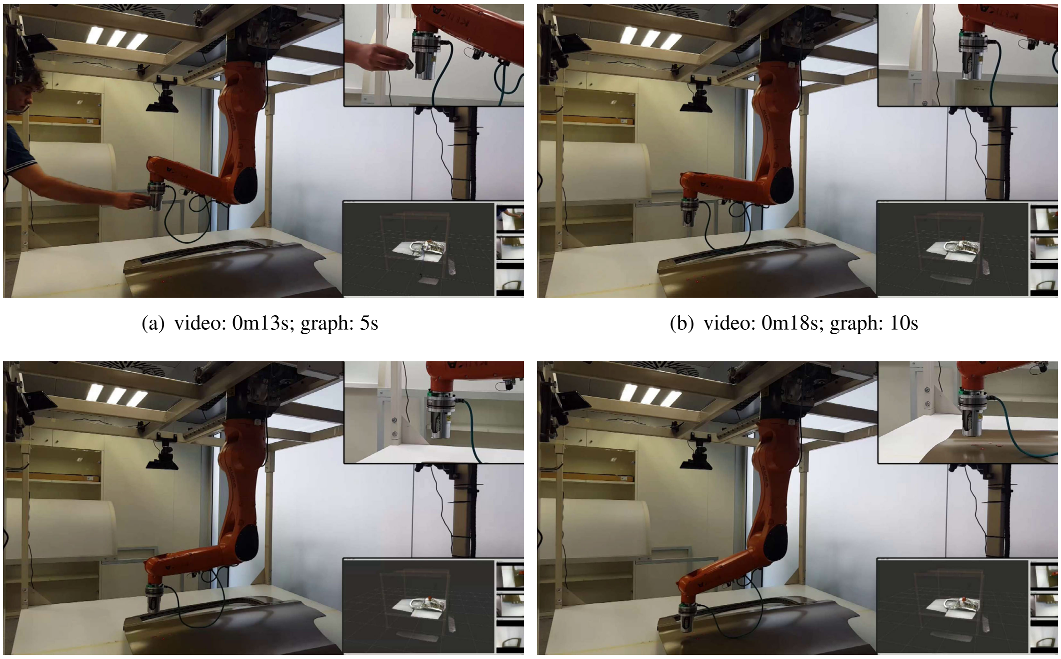
$$\dot{\mathbf{x}} = \begin{bmatrix}\mathbf{O} & \mathbf{I}\\ \mathbf{O} & \mathbf{O}\end{bmatrix}\mathbf{x} + \mathbf{d}_c + \begin{bmatrix}\mathbf{O}\\ \mathbf{I}\end{bmatrix}\mathbf{u}. \tag{14}$$

Taking into account the control law (9) and the constraint (12), the acceleration equality for this level results in:

$$\begin{aligned}\mathrm{pos}\,(\phi_d)\,\mathbf{L_g}\phi_d\,\ddot{\mathbf{q}}_c &= -\mathrm{pos}\,(\phi_d)\;u_2^+,\\ &\rightarrow \mathbf{A}_2\ddot{\mathbf{q}}_c = \mathbf{b}_2,\end{aligned} \tag{15}$$

where u_2^+ corresponds to the SMC switching gain, $\mathbf{b}_2$ and $\mathbf{A}_2$ are the vector and matrix for the second task in (4) and the Lie derivative $\mathbf{L_g}\phi_d$ is given by:

$$\begin{aligned}\mathbf{L_g}\phi_d &= (\partial\phi_d/\partial\mathbf{x})^{\mathrm{T}}\,\mathbf{g} = (\partial\phi_d/\partial\dot{\mathbf{q}})^{\mathrm{T}}\\ &= -K_2\left(\partial\dot{d}/\partial\dot{\mathbf{q}}\right)^{\mathrm{T}} = -K_2\,(\partial d/\partial\mathbf{q})^{\mathrm{T}} = K_2\,\mathbf{n}^{\mathrm{T}}\,\mathbf{J}_v.\end{aligned} \tag{16}$$



(a) video: 0m13s; graph: 5s

(b) video: 0m18s; graph: 10s

(c) video: 0m26s; graph: 18s

(d) video: 0m40s; graph: 32s

FIGURE 4. Frames of the first experiment recording. The time instant is indicated for each frame.

D. LEVEL 3: MANUAL AND AUTOMATIC MODES

This level is used to treat a specific point of the object surface with the robot tool considering two modes of operation. In particular, if the module of the forces detected by the guidance sensor is above a certain threshold F_{th}, i.e., $|\mathbf{F}| \geq F_{th}$, the manual mode is active and an admittance controller is considered to track the user's forces. Otherwise, $|\mathbf{F}| < F_{th}$, the automatic mode is active and a hybrid controller is used to track a reference Cartesian position.

Next, the control law for each mode of operation is detailed.

1) MANUAL MODE

In this mode of operation the user guides the robot tool to apply the surface treatment on a specific point of the object surface. For this purpose, and similarly to other works considering human-robot interaction [16], an admittance controller is employed to allow the operator to guide the robot tool. Thus, the forces performed by the user correspond to the desired tool velocities. Typically, an admittance controller is given by:

$$\mathbf{M}_3\,\dot{\mathbf{v}}_n + \mathbf{C}_3\,\mathbf{v}_n = \mathbf{F}, \tag{17}$$

where $\mathbf{v}_n = \mathbf{J}_n\dot{\mathbf{q}}$ is the tool velocity vector relative to the tool frame, $\mathbf{J}_n$ is the so-called geometric Jacobian [17] relative to the tool frame, $\mathbf{F}$ is vector of forces performed by the user relative to the tool frame and the parameters $\mathbf{M}_3$ and $\mathbf{C}_3$ are diagonal matrices of dimension 6×6 denoting the virtual inertia and damping, respectively. It is interesting to remark that this controller does not include the virtual stiffness due to the fact that ''restoring'' forces is not suitable for the proposed application of human-robot cooperation [16].

It is also worth noting that, since the roll α and pitch β angles of the tool are controlled in Level 1, only the Cartesian position $\overline{\mathbf{p}}$ and yaw angle γ of the robot tool are available at this level to be guided by the user.

Replacing the relation $\mathbf{v}_n = \mathbf{J}_n\dot{\mathbf{q}}$ in (17) and solving for $\ddot{\mathbf{q}}$ yields:

$$\mathbf{M}_3\,\mathbf{J}_n\,\ddot{\mathbf{q}} = \mathbf{F} - \mathbf{C}_3\,\mathbf{J}_n\,\dot{\mathbf{q}} - \mathbf{M}_3\,\dot{\mathbf{J}}_n\,\dot{\mathbf{q}}. \tag{18}$$

In order to avoid taking the time derivative of $\mathbf{J}_n$ for the last term in (18), the following hybrid control equation is considered for Level 2:

$$\begin{aligned}\mathbf{M}_3\,\mathbf{J}_n\,\ddot{\mathbf{q}}_c \;&=\; \mathbf{F} - \mathbf{C}_3\,\mathbf{J}_n\,\dot{\mathbf{q}} + \operatorname{sign}(\mathbf{F} - \mathbf{C}_3\,\mathbf{J}_n\,\dot{\mathbf{q}})\,u_{3m}^{+}\\ &\rightarrow \mathbf{A}_{3m}\ddot{\mathbf{q}}_c = \mathbf{b}_{3m},\end{aligned} \tag{19}$$

where u_{3m}^{+} denotes the gain for the last switching term and $\mathbf{A}_{3m}$ and $\mathbf{b}_{3m}$ are the matrix and vector for the third task in (4) when the manual mode is active. Note that the last term is introduced to compensate for the term $\mathbf{M}_3\,\dot{\mathbf{J}}\,\dot{\mathbf{q}}$ when the steady state given by $\mathbf{C}_3\,\mathbf{v}_n = \mathbf{F}$ has been reached, see (17).

2) AUTOMATIC MODE

Similarly to (11), the following hybrid control equation is considered to track the desired reference pose $\mathbf{p}_{ref}$ in the automatic mode:

$$\begin{aligned}\mathbf{J}\ddot{\mathbf{q}}_c \;&=\; \ddot{\mathbf{p}}_{ref} + K_{d3}\dot{\mathbf{e}} + K_{p3}\mathbf{e} + \operatorname{sign}\left(\dot{\mathbf{e}} + (K_{p3}/K_{d3})\mathbf{e}\right)u_{3a}^{+}\\ &\rightarrow \mathbf{A}_{3a}\ddot{\mathbf{q}}_c = \mathbf{b}_{3a}\end{aligned} \tag{20}$$

where vector $\mathbf{e} = \mathbf{p}_{ref} - \mathbf{p}$ represents the error of the tool pose; K_{p3} and K_{d3} are the gains to correct the tool pose error and its derivative, respectively; $\dot{\mathbf{p}}$ is computed from the first-order kinematics in (2); u_{3a}^{+} is the gain for the last switching term; and $\mathbf{A}_{3a}$ and $\mathbf{b}_{3a}$ are the matrix and vector for the third task in (4) when the automatic mode is active.

The reference pose $\mathbf{p}_{ref}$ for the robot tool is composed of the reference Cartesian position $\overline{\mathbf{p}}_{ref}$ and the reference orientation angles roll α_{ref}, pitch β_{ref} and yaw γ_{ref}. However, as before, the values used for roll and pitch angles have no effect since these angles are already constrained in Level 1. Whereas, the reference value used for the yaw angle remains fixed to the value that this angle had when the automatic mode became active, i.e., the objective is to keep the yaw angle still during the automatic mode.

Regarding the reference values used for the Cartesian position of the robot tool, four phases are considered to treat a specific point of the object surface. Firstly, an approach movement is generated to place the robot tool at a certain distance from the surface point to be treated. Secondly, a slow linear movement is used to place the robot tool on the surface point. Thirdly, the tool remains still for a certain time lapse to guarantee that the treatment has been completed. Fourthly, a slow linear movement is generated to departure from the object surface. Thus, these four phases are cyclically repeated for each surface point to be treated. Since the manual mode can interrupt this automatic process, when the automatic mode becomes active again, it resumes the mentioned phases for the surface point being considered before the interruption. Additionally, when the automatic mode becomes active, the reference for the tool Cartesian position stays still at the current value for a brief time lapse in order to deal with sudden switching between manual and automatic modes.

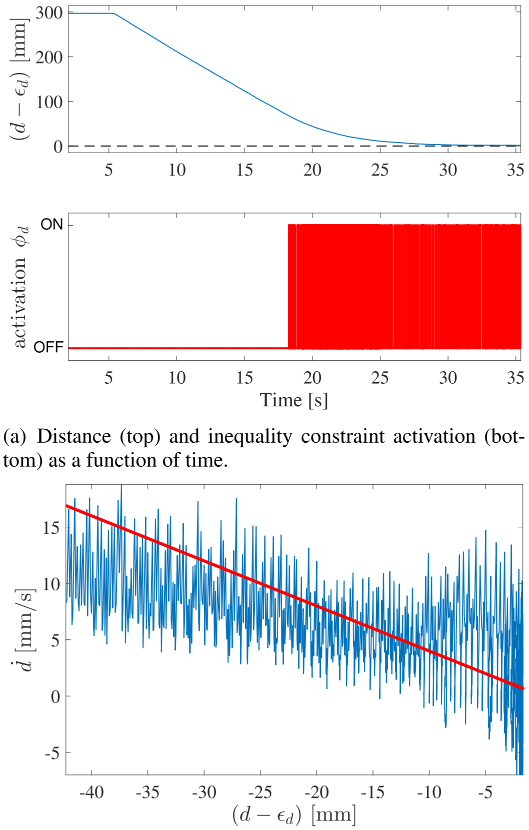
E. LIMITATIONS OF THE METHOD

The main limitation of the proposed approach is the so-called *chattering* effect [13], which is present in all three levels above due to the switching terms in the control equations. In particular, discrete-time implementations of the above discontinuous control laws make the system leave the ideal sliding mode behavior and oscillate with finite frequency and amplitude inside a band around the set-point.

Several approaches can be found in the literature to theoretically avoid this drawback [3], either by softening the discontinuous control or by using a dynamical or high-order discontinuous control. However, this issue is out of the scope of this research and remains as further work. See [3] for further details.

IV. CONTROLLER IMPLEMENTATION

The pseudo-code of the proposed approach is given in Table 1, which considers several functions: robot function $\mathbf{l}(\mathbf{q})$; robot Jacobians $\mathbf{J}$, $\mathbf{J}_v$ and $\mathbf{J}_n$; Moore-Penrose pseudoinverse $(\cdot)^{\dagger}$ (a threshold is considered to discard the small singular values [20]); *GetRobot&Forces&VisionData*, which provides the current robot angles and velocities $\{\mathbf{q}, \dot{\mathbf{q}}\}$, the force vector $\mathbf{F}$ (it is assumed that the sensor electronics has filtered the force measurements) and the vision data $\{\mathbf{n}, d\}$; *OrientationOfVector*($\mathbf{n}$), which provides [17] the orientation of a vector (that is, the roll α and pitch β angles); and *SendToJointControllers*($\ddot{\mathbf{q}}_c$), which commands to the low-level controller the desired accelerations. Furthermore, the reference pose $\mathbf{p}_{ref}$ for the robot tool is supplied by the automatic mode with the procedure detailed in Section III-D2. The computational load of the proposed method (compiled C code in a modern computer) is about 15 microseconds for the case of the real experimentation in Section V.



(a) Distance (top) and inequality constraint activation (bottom) as a function of time.

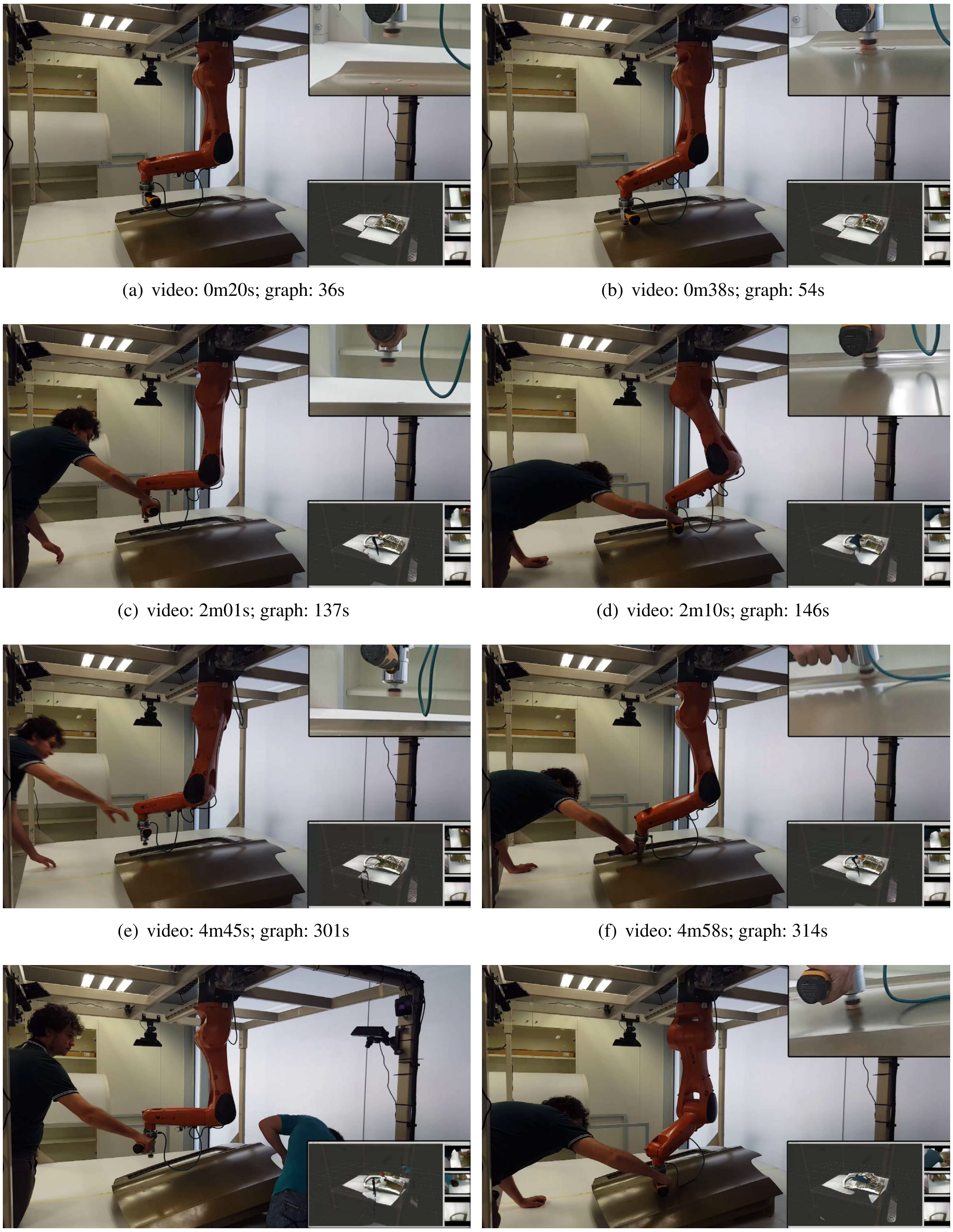
(b) Phase plane relating the rate of distance reduction and distance: thin line, actual values; thick line, boundary of distance reduction given by the inequality constraint.

FIGURE 5. **Time response and phase plane for the first experiment.**

V. REAL EXPERIMENTATION

A. SETUP

Fig. 3 shows the setup considered for the real experimentation, which consists of: a Kuka KR6 r900 sixx robot; an industrial sanding tool (Mirka AROS-B 150NV 32mm 10.8V 2.0 Ah Orbit 5.0) attached to the end-effector of the robot by means of a self-developed adapter; a force/torque sensor Axia80 located between the last link of the robot and the sanding tool, which is employed by the user to guide the tool; a sanding disc consisting of a cylinder of 28 × 28 × 22 mm; a network of three RGB-D Microsoft Kinect sensors; and a car door as object to apply the surface treatment.



(a) video: 0m20s; graph: 36s

(b) video: 0m38s; graph: 54s

(c) video: 2m01s; graph: 137s

(d) video: 2m10s; graph: 146s

(e) video: 4m45s; graph: 301s

(f) video: 4m58s; graph: 314s

(g) video: 6m07s; graph: 383s

(h) video: 6m29s; graph: 405s

FIGURE 6. Frames of the second experiment recording. The time instant is indicated for each frame.

The proposed algorithm was implemented in an external PC workstation (Intel-R Core-TM i7-5820K CPU 3.30GHz) with: *Ubuntu 16.04* as Operative System; *ROS* (Robot Operating System) *Kinetic* distribution as middle-ware to interface with the Kuka robot; and the *netft_rdt_driver* ROS package provided by the sensor manufacturer (ATI Industrial Automation). An Ethernet switch device was used to connect and communicate (UDP protocol) the robot, the force-torque sensor and the workstation. The Microsoft Kinect network was connected to serial ports of the PC workstation.

B. CONDITIONS FOR THE EXPERIMENTS

i) A sampling frequency T_s^{-1} of 100Hz was used and the accelerations $\ddot{\mathbf{q}}_{c,3}$ were integrated twice and commanded to the low-level controller.

ii) The measurements of the force-torque sensor were filtered with a low-pass filter (implemented in the sensor electronics) with a cut-off frequency of 115 Hz.

iii) Level 1 parameters (Section III-B): $K_{p1} = 1.5$, $K_{d1} = 1.8$ and $u_1^+ = 0.01$.

iv) Level 2 parameters (Section III-C): $\epsilon_d = 2$ mm, $K_2 = 2.5$ and $u_2^+ = 0.65$.

v) Level 3 parameters (Section III-D): $F_{th} = 1$, $\mathbf{M}_3 = 10\,\mathbf{I}$, $\mathbf{C}_3 = 70\,\mathbf{I}$, $u_{3m}^+ = 0.01$, $K_{p3} = 2$, $K_{d3} = 4.2$ and $u_{3a}^+ = 0.01$.

C. RESULTS

A first experiment was carried out to study the performance of the approach of the robot end-effector to the car door surface whose video can be played at https://media.upv.es/player/?id=b4e1be40-6c53-11ea-8c87-2ba15c40d192 (note that the sanding tool was removed from the robot end-effector to have a clearer view). Fig. 4 presents a few frames of the recording: Fig. 4(a) (time instant 13s in the recording) shows how the user attaches a small object with a mass of 0.15 Kg to the end-effector of the robot, giving rise to the activation of the manual mode; Fig. 4(b) and Fig. 4(c) (interval 18s–26s in the video) show how the robot end-effector "falls down", approaching the car door while progressively reducing the vertical speed; and Fig. 4(d) (time instant 40s in the recording) shows that the robot completely stops, keeping the safety distance to the car door.

Fig. 5 shows the quantitative behavior of the robot approach to the car door surface. In particular, it can be seen in the top plot of Fig. 5(a) that the distance from the robot end-effector to the car door surface is initially about 300mm and starts decreasing around time instant 5s, which corresponds to the moment when the user attaches the small object to the robot end-effector. Then, the rate of distance reduction is approximately constant in the interval 5s–18s, which is due to the fact that the constant weight of the small object is converted by the admittance controller of Level 3 to a negative vertical velocity for the robot end-effector, see the manual mode in Section III-D1. Around time instant 18s the inequality constraint becomes active, see the bottom plot in Fig. 5(a), and the rate of distance reduction is progressively decreased by the non-conventional SMC of Level 2 until the robot completely stops when the distance d equals the safety margin ϵ_d, which occurs around time instant 32s. Fig. 5(b) shows a detailed view of the rate of distance reduction as a function of the distance, where it can be seen how the system is switching around the boundary of the inequality constraint, which is given by a straight line with a slope of $-1/K_2$, see (12).

Another experiment was performed to show the generality of the proposed approach whose video can be played at https://media.upv.es/player/?id=ed3e6250-6c55-11ea-8c87-2ba15c40d192. In this experiment, two points of the car door surface have been considered for the automatic mode, whereas the user activates the manual mode several times during the experiment in order to treat other points of the car door surface. Moreover, the position and orientation of the car door is modified around time instant 6m10s of the video in order to show how the vision sensors update in real-time the information required for the surface treatment task, i.e., the positions of the points to be treated and the distance and normal vector to the car door surface. Fig. 6 presents a few frames of the recording: Fig. 6(a) and 6(b) (interval 20s–38s in the recording) show how the tool applies the surface treatment for the first point of the automatic mode; Fig. 6(c) and 6(d) (interval 2m01s–2m10s) show how the user activates the manual mode in order to treat a specific point of the car door surface; Fig. 6(e) and 6(f) (interval 4m45s–4m58s) show how the user activates again the manual mode in order to treat a point close to a style-line of the car door surface, which is done properly by the robot despite that this point is more challenging due to the sharpness of this area of the car door; and Fig. 6(g), and 6(h) (interval 6m07s–6m29s) show how the robot keeps working properly after the relocation of the car door thanks to the real-time data provided by the vision system.

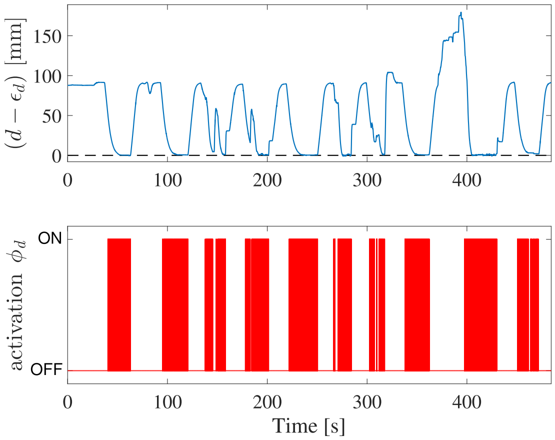


FIGURE 7. Distance (top) and inequality constraint activation (bottom) as a function of time for the second experiment.

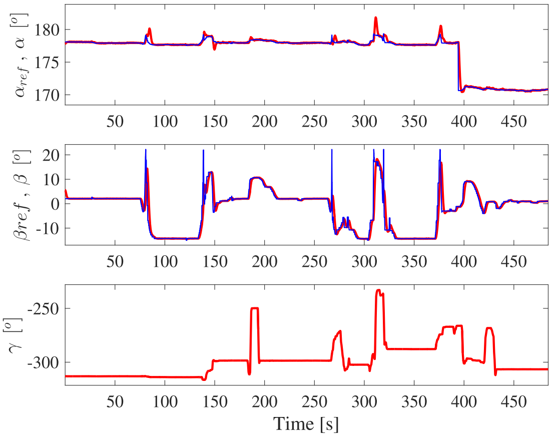


FIGURE 8. Angles of the robot tool in the second experiment: top, roll angle; middle, pitch angle; and bottom, yaw angle. In the first and second plots the thick line represents the actual values for the angle, whereas the thin line represents the reference values provided by the computer vision data.

Fig. 7 shows the behavior of the approach of the robot tool to the car door surface, where it can be appreciated that the activation of the inequality constraint in Level 2 (see the bottom plot) ensures that the distance d from the robot tool to the car door surface never exceeds the safety margin ϵ_d (see the top plot).

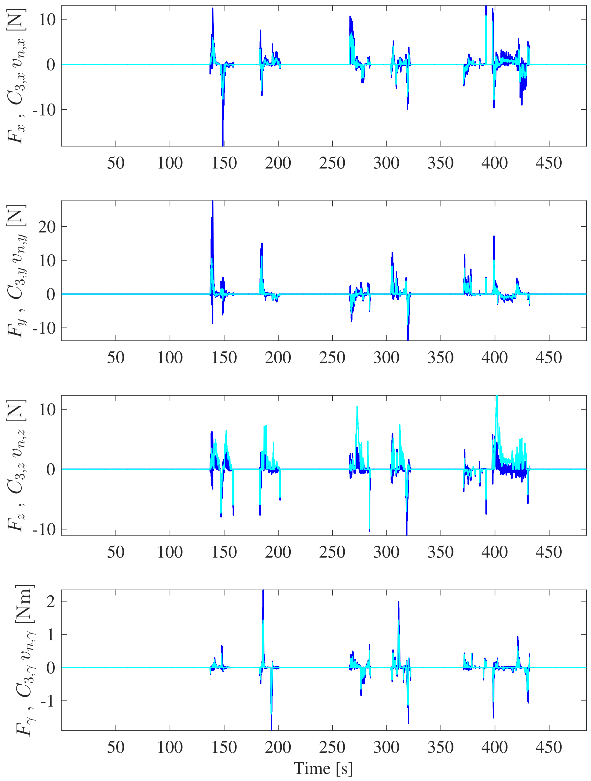


FIGURE 9. Forces of the user (dark-blue) and tool velocities (light-cyan) multiplied by the virtual damping C_3: first plot, linear X-axis; second plot, linear Y-axis; third plot, linear Z-axis; and fourth plot, angular Z-axis (coordinates relative to the tool frame).

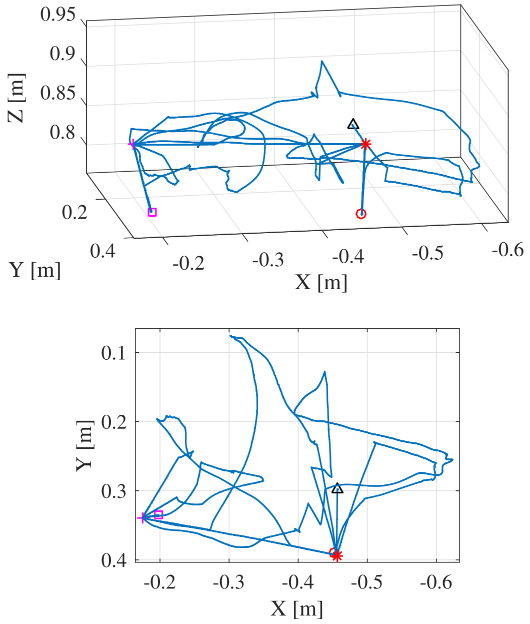


FIGURE 10. 3D representation of the trajectory described by the tool in the second experiment. The following symbols are used to represent several positions: triangle, initial position of the robot tool; circle, position of the first point to be treated; star, approach position for the first point to be treated; square, position of the second point to be treated; and cross, approach position for the second point to be treated.

Fig. 8 shows the tool angles (roll, pitch and yaw) during the second experiment. In particular, the orientation control in Level 1 ensures that the roll and yaw angles follow closely the reference signals provided by the vision system, see the first and second plots. Moreover, the manual mode in Level 3 allows the user to modify the yaw angle as shown in the bottom plot.

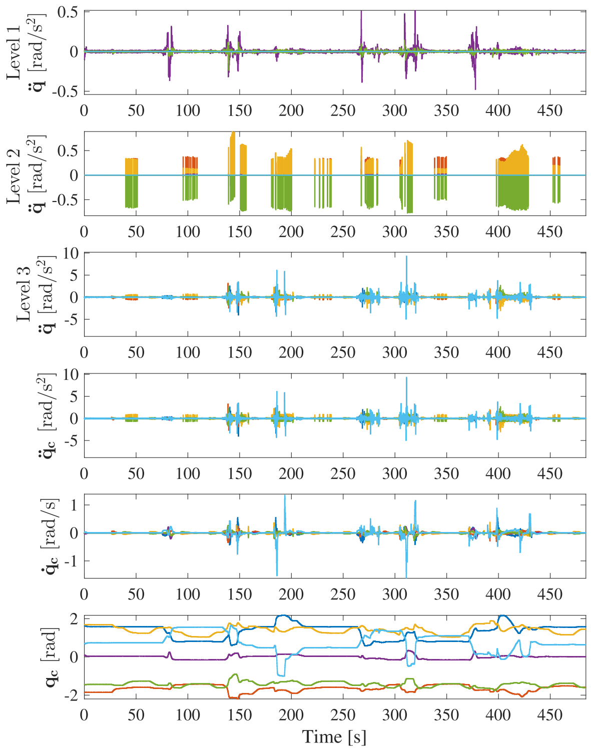


FIGURE 11. Control actions for the second experiment: contribution of each priority level to the commanded joint accelerations in the first three plots, the commanded joint accelerations and velocities in the fourth and fifth plot, respectively, and joint positions commanded to the robot controller in the bottom plot.

Fig. 9 shows the forces of the user and the tool velocities during the second experiment, which are very similar thanks to the admittance controller used in Level 3 for the manual mode.

The Cartesian position of the robot tool during the second experiment is depicted in Fig. 10, where it can be appreciated that the tool has covered a large area to treat not only the points of the automatic mode but also the points established by the user in the manual mode.

Finally, the control signals are shown in Fig. 11, where it can be appreciated how each level contributes to the commanded accelerations.

VI. CONCLUSION

This work developed a human-robot cooperation to carry out treatments such as sanding, polishing, etc. on the surface of a known rigid object. To achieve this goal, a vision system was used to obtain the location of the object in order to ensure not only the orthogonality of the robot tool to the object surface but also a smooth approach of the tool to the surface.

Furthermore, in order to add flexibility to the application, the proposal included the simultaneous combination of automatic and manual modes of operation. Thus, the human operator can guide the robot tool to treat arbitrary areas (manual mode) and, when the operator releases the tool, the robot goes into the automatic mode to treat prior established areas.

The applicability and efficacy of the proposed method was shown experimentally using a conventional 6R arm manipulator.

REFERENCES

[1] S. S. Martínez, J. G. Ortega, J. G. García, A. S. García, and E. E. Estévez, ''An industrial vision system for surface quality inspection of transparent parts,'' *Int. J. Adv. Manuf. Technol.*, vol. 68, nos. 5–8, pp. 1123–1136, Sep. 2013, doi: 10.1007/s00170-013-4904-2.

[2] J. Molina, J. E. Solanes, L. Arnal, and J. Tornero, ''On the detection of defects on specular car body surfaces,'' *Robot. Comput.-Integr. Manuf.*, vol. 48, pp. 263–278, Dec. 2017, doi: 10.1016/j.rcim.2017.04.009.

[3] J. E. Solanes, L. Gracia, P. Muñoz-Benavent, J. Valls Miro, C. Perez-Vidal, and J. Tornero, ''Robust hybrid position-force control for robotic surface polishing,'' *J. Manuf. Sci. Eng.*, vol. 141, no. 1, pp. 011013-1–011013-14, Jan. 2019, doi: 10.1115/1.4041836.

[4] J. Yuan, Y. Qian, L. Gao, Z. Yuan, and W. Wan, ''Position-based impedance force controller with sensorless force estimation,'' *Assem. Autom.*, vol. 39, no. 3, pp. 489–496, Aug. 2019, doi: 10.1108/AA-09-2018-0124.

[5] T. Segreto and R. Teti, ''Machine learning for in-process end-point detection in robot-assisted polishing using multiple sensor monitoring,'' *Int. J. Adv. Manuf. Technol.*, vol. 103, nos. 9–12, pp. 4173–4187, Aug. 2019, doi: 10.1007/s00170-019-03851-7.

[6] L. Gracia, J. E. Solanes, P. Muñoz-Benavent, A. Esparza, J. V. Miro, and J. Tornero, ''Cooperative transport tasks with robots using adaptive non-conventional sliding mode control,'' *Control Eng. Pract.*, vol. 78, pp. 35–55, Sep. 2018, doi: 10.1016/j.conengprac.2018.06.005.

[7] A. Roswell, F. Xi, and G. Liu, ''Modelling and analysis of contact stress for automated polishing,'' *Int. J. Mach. Tools Manuf.*, vol. 46, nos. 3–4, pp. 424–435, Mar. 2006, doi: 10.1016/j.ijmachtools.2005.05.006.

[8] M. Nilsen, F. Sikström, A.-K. Christiansson, and A. Ancona, ''Robust vision-based joint tracking for laser welding of curved closed-square-butt joints,'' *Int. J. Adv. Manuf. Technol.*, vol. 101, nos. 5–8, pp. 1967–1978, Apr. 2019, doi: 10.1007/s00170-018-3044-0.

[9] P. Muñoz-Benavent, J. E. Solanes, L. Gracia, and J. Tornero, ''Robust auto tool change for industrial robots using visual servoing,'' *Int. J. Syst. Sci.*, vol. 50, no. 2, pp. 432–449, Jan. 2019, doi: 10.1080/00207721.2018.1562129.

[10] S. Astanin, D. Antonelli, P. Chiabert, and C. Alletto, ''Reflective workpiece detection and localization for flexible robotic cells,'' *Robot. Comput.-Integr. Manuf.*, vol. 44, pp. 190–198, Apr. 2017, doi: 10.1016/j.rcim.2016.09.001.

[11] A. Ferraro, M. Indri, and I. Lazzero, ''Dynamic update of a virtual cell for programming and safe monitoring of an industrial robot,'' in *Proc. 10th IFAC Symp. Robot Control*, vol. 45, no. 22, 2012, pp. 822–827, doi: 10.3182/20120905-3-HR-2030.00112.

[12] Z. M. Bi and L. Wang, ''Advances in 3D data acquisition and processing for industrial applications,'' *Robot. Comput.-Integr. Manuf.*, vol. 26, no. 5, pp. 403–413, Oct. 2010, doi: 10.1016/j.rcim.2010.03.003.

[13] C. Edwards and S. Spurgeon, *Sliding Mode Control: Theory And Applications*, 1st ed. Abingdon, U.K.: Taylor & Francis, 1998, doi: 10.1201/9781498701822.

[14] K. Liu, Y. Wu, J. Xu, Y. Wang, Z. Ge, and Y. Lu, ''Fuzzy sliding mode control of 3-DOF shoulder joint driven by pneumatic muscle actuators,'' *Int. J. Robot. Autom.*, vol. 34, no. 1, pp. 38–45, 2019, doi: 10.2316/J.2019.206-5063.

[15] N. Adhikary and C. Mahanta, ''Sliding mode control of position commanded robot manipulators,'' *Control Eng. Pract.*, vol. 81, pp. 183–198, Dec. 2018, doi: 10.1016/j.conengprac.2018.09.011.

[16] F. Dimeas and N. Aspragathos, ''Online stability in human-robot cooperation with admittance control,'' *IEEE Trans. Haptics*, vol. 9, no. 2, pp. 267–278, Apr. 2016, doi: 10.1109/TOH.2016.2518670.

[17] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*. London, U.K.: Springer-Verlag, 2009, doi: 10.1007/978-1-84628-642-1.

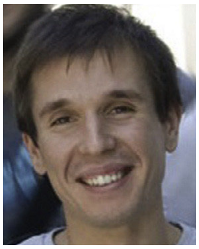
[18] Y. Nakamura, H. Hanafusa, and T. Yoshikawa, ''Task-priority based redundancy control of robot manipulators,'' *Int. J. Robot. Res.*, vol. 6, no. 2, pp. 3–15, Jun. 1987, doi: 10.1177/027836498700600201.

[19] B. Siciliano and J.-J.-E. Slotine, ''A general framework for managing multiple tasks in highly redundant robotic systems,'' in *Proc. 5th Int. Conf. Adv. Robot. 'Robots Unstructured Environ.*, 1991, pp. 1211–1216, doi: 10.1109/ICAR.1991.240390.

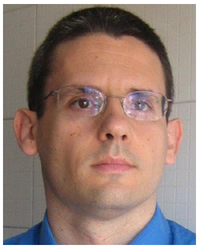
[20] G. Golub and C. Van Loan, *Matrix Computations*, 3rd ed. Baltimore, MD, USA: The Johns Hopkins University Press, 1996, doi: 10.2307/3616959.

ALBERTO GARCÍA received the B.Eng. degree in industrial technologies and the M.Sc. degree in industrial engineering from the Universitat Politècnica de València (UPV), in 2016 and 2018, respectively. He is currently pursuing the Ph.D. degree in automation, robotics and computer science with the Instituto de Diseño y Fabricación. He is also a Researcher with the Instituto de Diseño y Fabricación. His research interests include human–robot interaction, industrial robotics, and robust and human–robot collaboration.

VICENT GIRBÉS-JUAN received the B.Eng. and M.Sc. degrees in industrial electronics and control, in 2009 and 2011, respectively, and the Ph.D. degree in automation, robotics and computer science from the Universitat Politècnica de València (UPV), in 2016. From 2009 to 2019, he was with the Robotics and Automation Research Group, Design and Manufacturing Institute, UPV (IDF-UPV). He was a Visiting Researcher with The University of Manchester, U.K., for a period of six months, and Imperial College London, U.K., for a period of one year. Since October 2019, he has been an Assistant Professor with the Department of Electronic Engineering, Universitat de València. His research interests include smooth path planning and obstacle avoidance, collaborative robotics, human-robot interaction, ADAS, and machine learning.



J. ERNESTO SOLANES received the B.S. degree in industrial electronics engineering, the B.S. degree in industrial automatics, the M.S. degree in automatics and industrial informatics, and the Ph.D. degree in robotics, automatics and industrial informatics from the Universitat Politècnica de València (UPV), Spain, in 2007, 2009, 2011, and 2015, respectively. He is currently an Assistant Professor with the Department of Systems Engineering and Control (DISA), UPV. His research interests include nonlinear and robust control, computer vision, and robotics.



LUIS GRACIA received the B.Sc. degree in electronic engineering, the M.Sc. degree in control systems engineering, and the Ph.D. degree in automation and industrial computer science from the Universitat Politècnica de València (UPV), Spain, in 1998, 2000, and 2006, respectively. He is currently a Professor with the Department of Systems Engineering and Control (DISA), UPV. His research interests include mobile robots, robotic manipulators, and system modeling and control.

CARLOS PEREZ-VIDAL received the B.Sc. degree in industrial engineering, in 1998, the M.Sc. degree in control engineering, in 2000, and the Ph.D. degree in industrial technologies from the Technical University of Valencia and UMH, Spain, in 2008. He has been active in research and development with several projects on advanced robotics and vision since 2001. He is currently an Associate Professor of control and systems engineering with Universidad Miguel Hernández, Spain. He is also a Researcher with the Systems Engineering and Automatic Control Division. He is the author or coauthor of multiple research publications. His current research interests include robotics, direct visual servoing, and automation/control and medical applications.

JOSEP TORNERO received the M.S. degree in systems and control from The University of Manchester, in 1982, and the Ph.D. degree in electrical engineering from the Universitat Politècnica de València (UPV), in 1985. He has been a Visiting Professor with CIRSSE (NASA Center for Intelligent Robotics Systems for Space Exploration), the Rensselaer Polytechnic Institute, Troy, NY, USA, and the Department of Mechanical Engineering, University of California, Berkeley. He is currently a Professor with the Department of Systems Engineering and Control (DISA), UPV, where he is also responsible with the Design and Manufacturing Institute (IDF). His research interests include control of robot systems and multirate sampled data systems.