

Document downloaded from:

<http://hdl.handle.net/10251/182479>

This paper must be cited as:

García-Fernández, A.; Gracia Calandin, LI.; Solanes Galbis, JE.; Girbés-Juan, V.; Perez-Vidal, C.; Tornero Montserrat, J. (2021). Robotic assistance for industrial sanding with a smooth approach to the surface and boundary constraints. *Computers & Industrial Engineering*. 158:1-16. <https://doi.org/10.1016/j.cie.2021.107366>



The final publication is available at

<https://doi.org/10.1016/j.cie.2021.107366>

Copyright Elsevier

Additional Information

**Title:** Robotic assistance for industrial sanding with a smooth approach to the surface and boundary constraints

**Authors:** Alberto García<sup>a</sup>, Luis Gracia<sup>a</sup>, J. Ernesto Solanes\*<sup>a</sup>, Vicent Girbés-Juan<sup>b</sup>, Carlos Perez-Vidal<sup>c</sup> and Josep Tornero<sup>a</sup>

**\*Corresponding author** (E-mail: [esolanes@idf.upv.es](mailto:esolanes@idf.upv.es))

<sup>a</sup>Instituto de Diseño y Fabricación, Universitat Politècnica de València, Camino de Vera s/n, 46022 Valencia, Spain.

<sup>b</sup>Departament d'Enginyeria Electrònica, Universitat de València, Avda de la Universitat s/n, 46100 Burjassot, Spain.

<sup>c</sup>Departamento de Ingeniería de Sistemas y Automática, Universidad Miguel Hernández, Avda de la Universidad s/n, 03202 Elche, Spain.

**Acknowledgements of the paper:** This research was founded by the Spanish Government and the Generalitat Valenciana under Grants PID2020-117421RB-C21, GV/2021/005 and ACIF/2019/007

**LINKS for the videos in Ref. [40], [41] and [42] (non-double-blind):**

[40] Video of the first experiment,

<https://media.upv.es/player/?id=accfd130-5599-11eb-ba71-19e9cf6f3c04>

[41] Video of the second experiment,

<https://media.upv.es/player/?id=6ef90bd0-5597-11eb-ba71-19e9cf6f3c04>

[42] Video of the third experiment,

<https://media.upv.es/player/?id=d426bab0-5598-11eb-ba71-19e9cf6f3c04>

**Declaration of interests:** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Robotic assistance for industrial sanding with a smooth approach to the surface and boundary constraints

---

### Abstract

Surface treatment operations, such as sanding, deburring, finishing, grinding, polishing, etc. are progressively becoming more automated using robotic systems. However, previous research in this field used a completely automatic operation of the robot system or considered a low degree of human-robot interaction. Therefore, to overcome this issue, this work develops a truly synergistic cooperation between the human operator and the robot system to get the best from both. In particular, in the application developed in this work the human operator provides flexibility, guiding the tool of the robot system to treat arbitrary regions of the workpiece surface; while the robot system provides strength, accuracy and security, not only holding the tool and keeping the right tool orientation, but also guaranteeing a smooth approach to the workpiece and confining the tool within the allowed area close to the workpiece. Moreover, to add more flexibility to the proposed method, when the user is not guiding the robot tool, a robot automatic operation is activated to perform the treatment in prior established regions. Furthermore, a camera network is used to get a global view of the robot workspace in order to obtain the workpiece location accurately and in real-time. The effectiveness of the proposed approach is shown with several experiments using a 6R robotic arm.

*Keywords:* human-robot cooperation, smooth approach, boundary constraints

---

## 1. Introduction

### 1.1. Objective

It is well known that industrial manufacturing is progressively becoming more automated. The goal of this process is to produce goods faster, more precisely and more efficiently. To achieve this objective, human operators tend to be replaced by machines in tough and hazardous tasks. However, the current trend is to allow the operator to share the same workspace with the machines in order to obtain a synergistic cooperation between them, i.e., the operator provides flexibility while the robot system provides strength and accuracy.

This work is focused on surface quality control, which is a very relevant industrial task composed of two subtasks: detecting surface anomalies and repairing them by means of a specific surface treatment: sanding, polishing, grinding, deburring, brushing, etc. Although several solutions have been developed in recent years to solve the detection process by means of vision systems [1], the repairing issue is still mainly performed by human operators [2].

Thus, this work aims to automate surface treatment operations by means of a synergistic cooperation between a robot system and the human operator.

### 1.2. Literature review about robotic surface treatment

Much research has been developed in recent years to completely or partially automate the application of surface treatments, as discussed below.

#### 1.2.1. Robot tool

Some *tools* have been specifically designed to properly accomplish surface treatment tasks using robot systems.

For instance, the manufacturer FerRobotics has a product line of robot tools, named *active contact flange* [3], which basically consist of a pneumatic system that allows the robot to reliably meet the force required to properly accomplish the surface treatment task.

Similarly to the previous approach, an adjustable force regulation mechanism to be placed in the end-effector of the robotic system was designed in [4] to

30 control the contact force for grinding and deburring operations without using a force sensor.

Moreover, the conventional structure of the robot end-effector used for polishing tasks was redesigned in [5] to obtain a low-inertia effect in order to reduce vibrations and to improve the tracking of the required contact force.

### 35 1.2.2. Automatic mode of operation

Most of the previous research that considered a robot system to perform surface treatment tasks used an *automatic mode* of operation, i.e., no kind of cooperation or interaction was considered between the human operator and the robot system to perform the surface treatment task.

40 For instance, a laser sensor mounted on the robot end-effector was used in [6] to previously scan the target workpiece in order to generate the trajectories that were subsequently followed by the robot system to perform the deburring operation. Similarly to the previous approach, a vision system mounted on the robot system was used in [7] to previously position the machining target, which  
45 was subsequently treated by the grinding robot.

In [8, 9], a tool path planning algorithm with controlled force and polishing parameters optimization was developed to perform a polishing operation using a robot arm that held the workpiece that was being polished by an external passive polishing tool.

50 In [10], a serial-parallel robot was considered to polish unknown curved surfaces using data provided by a force sensor to properly adapt the tool posture and the polishing pressure. Similarly, in [11] a polishing pressure model was considered to improve the polishing quality on curved surfaces by ensuring a constant polishing pressure.

55 In [12], a force-sensorless control was developed to grind an object with a robot system using the analytic relation between the contact force and the grinding resistance.

In [13], a machine learning algorithm based on multiple sensor data was developed to determine the state of the robot-assisted polishing process in order

60 to establish the end-point of the polishing process.

In [14], a force planning strategy based on a compliance model was developed to avoid large instantaneous contact forces when the grinding tool of the robot contacted the workpiece. Moreover, an algorithm for automatic contour surface processing and target force tracking was proposed in order to simplify  
65 the programming of the grinding task.

In [15], a path planning application was developed based on the CAD data of the workpiece to be polished. Note that this is a basic approach where the reference points for the robot system were computed offline and no kind of pressure adaptation was performed while the robot was polishing the workpiece.

### 70 1.2.3. Human-robot Cooperation

Some previous research considered some sort of interactions between the human operator and the robot system when performing the surface treatment tasks.

In [16], impedance control was used to program a polishing task using the  
75 teaching by demonstration method. That is, firstly the human operator guided the robot tool by hand in order to “teach” the robot the polishing pattern of positions and forces. Subsequently, the robot automatically performed the polishing task tracking the mentioned pattern of positions and forces. Note that, in this approach, the interactions between the operator and the robot  
80 system were only considered at a previous stage, i.e., to program the polishing task *offline*. Moreover, note that the robot system was not able to adapt to unforeseen changes of the target object, e.g., changes in position, shape, size, consistency, etc.

Similarly to the previous approach, in [17] the teaching by demonstration  
85 method was used to initially register the position and force data measured from a skilled operator. Subsequently, the robot system held the workpiece while it was being polished by an external passive tool. For this purpose, the virtual mechanism approach was used to characterize the closed kinematic chain composed by the union of the robot arm and the external polishing tool.

90 Furthermore, in [18] a human-robot collaboration application for manual  
polishing was developed, where the robot arm held the workpiece and the oper-  
ator used an abrasive tool to perform the polishing task. During the operation  
the robot arm maintained the workpiece in a fixed position and the operator  
was able to modify its orientation by pushing the robot body, which was de-  
95 tected by a force sensor attached to the wrist of the robot. However, it is worth  
noting that this was a “passive” robotic application, since the human user had  
to treat the surface with the tool, keeping both the desired pressure and the  
orthogonality between the tool of the robot system and the workpiece surface.

The application of surface treatments requires not only contact but also or-  
100 thogonality between the workpiece surface and the robot tool [19]. For instance,  
a force/torque sensor mounted on the robot end-effector was utilized in [20] to  
simultaneously regulate the contact pressure and the mentioned perpendicular-  
ity. Furthermore, a second force sensor attached to the robot tool was used  
in [20] to allow the operator guiding the tool to treat an arbitrary region of the  
105 workpiece surface.

In [21], a teleoperation system was developed in order to conduct echogra-  
phies using a medical robot, i.e., a physician teleoperated, using a haptic device,  
a robot equipped with an ultrasound probe in order to obtain ultrasound im-  
ages from a patient. In this application the physician perceived, by means of  
110 the haptic device, the stiffness of the patient, which was detected by means of a  
force sensor mounted on the robot end-effector. Moreover, a virtual progressive  
stiffness before contact was considered to facilitate the physician a smooth tran-  
sition from free space to contact and vice versa. For this purpose, the distance  
from the robot to the patient before contact was obtained using a 3D vision  
115 system mounted on the robot end-effector. However, note that such a smooth  
transition was not guaranteed with the aid of the aforementioned perceived stiff-  
ness, since the application was indeed a teleoperation that ultimately depended  
on the physician’s commands.

### 1.3. Proposal

120 The proposed approach presents several contributions to the previous literature, as discussed below.

#### 1.3.1. Truly cooperative

In contrast to the works [6, 7, 8, 9, 10, 11, 12, 13, 14, 15] mentioned above, where an automatic mode of operation was considered for the robot system to perform the surface treatment, the proposed approach develops, from a qualitative perspective, a synergistic cooperation between the human operator and the robot system to get the best from both. That is, the ability of the human operator to establish the regions of the workpiece surface that require the treatment, and the strength and precision of the robot system. In particular, in the application developed in this work the human operator guides the robot tool to the workpiece regions to be treated; while the robot system not only holds the tool and keeps the right tool orientation, but also guarantees a smooth approach to the workpiece and confines the tool within the allowed area close to the workpiece.

135 Furthermore, the degree of human-robot cooperation achieved by the proposed application is higher than that achieved by the works mentioned above in Section 1.2.3, as discussed below.

Both [16] and [17] considered the interactions between a skilled operator and the robot only to program the task and, subsequently, the surface treatment was performed by the robot alone using an automatic mode of operation. Hence, in contrast to the proposed approach, the robot was not able to adapt to unforeseen changes of the target object, e.g., changes in position, shape, etc.

Both [17] and [18] used the robot system to hold the workpiece that was being polished by an external passive tool [17] or by the human operator equipped with the tool [18]. Thus, the cooperation degree provided by the robot in these works is rather limited. In contrast, the proposed approach takes advantage of the robot system not only to hold the tool but also to keep, in real-time, the right tool orientation and to ensure a smooth approach to the workpiece surface.



Moreover, the proposed approach is also suitable to treat large workpieces, e.g.,  
150 the body of a car, which obviously would not be possible with the methods  
in [17] and [18].

In [20] the perpendicularity between the robot tool and the workpiece surface  
was ensured by means of a force sensor mounted on the robot end-effector.  
However, this approach only applied to the contact phase. In contrast, the  
155 proposed approach ensures the aforementioned orthogonality for both contact  
and non-contact phases, i.e., the robot cooperation applies in both phases.

The medical application developed in [21] was essentially a teleoperation task  
to perform an echography. Although the physician perceived the stiffness of the  
patient and surrounding areas through a haptic device, which was an aid to  
160 smooth the approach of the robot tool to the patient, such approach ultimately  
depended on the physician's commands. Thus, the cooperation degree provided  
by the robot in this work is rather limited. In contrast, the proposed approach  
uses the robot system in a more active and automatic manner, i.e., a smooth  
approach to the target object is guaranteed by means of a robot control system.

### 165 1.3.2. Camera network

A machine vision system is used in this work to get the location of the  
target workpiece in order to guarantee, whilst the user is guiding the tool to  
treat arbitrary regions of the workpiece surface, the perpendicularity between  
the workpiece surface and the tool and to ensure that the robot tool approaches  
170 the workpiece surface smoothly. As mentioned above, in contrast to other type  
of sensors such as force sensors, the vision system allows the robot to control not  
only the contact phase between the tool and the workpiece but also the non-  
contact phase. However, note that using computer vision implicitly assumes  
that the target workpiece is rigid and known a priori.

175 The machine vision system developed in this research is based on a camera  
network, which has several advantages compared to the previous literature, as  
discussed below.

In [6] and [7] a laser sensor and a vision system, respectively, mounted on the

robot end-effector were used to previously scan and locate, also respectively, the  
180 target workpiece. However, these approaches had several drawbacks: a *previous*  
*phase* was required to scan or locate the workpiece; if the workpiece were sub-  
stantially modified (e.g., if a larger workpiece were considered), the mentioned  
previous phase would have to be *reprogrammed*; the workpiece location was not  
updated in real-time and, hence, the robot system was not able to adapt to  
185 changes in the workpiece location, i.e., the workpiece had to be *static*; and the  
data obtained by the laser or camera was *local* to the robot end-effector and,  
hence, it was difficult to cover large areas.

In contrast, the proposed approach has the following advantages: no previ-  
ous phase is required; the method does not depend on the workpiece considered;  
190 the workpiece location is updated in real-time and, hence, the workpiece can be  
moved; and the workpiece location is obtained globally, i.e., it is not constrained  
by the robot kinematics, and, hence, larger areas may be covered. Moreover,  
note that the camera network represents a *redundant* system that provides ac-  
curacy and robustness, e.g., when the robot or human operator occlude the  
195 field of view of a camera, the remaining cameras are able to properly locate the  
workpiece.

### 1.3.3. Smooth approach

In this work, a robust control system is presented to regulate the maxi-  
mum velocity at which the tool of the robot system is allowed to approach the  
200 workpiece surface, being zero when the tool contacts the workpiece. Note that  
this control is not active either when the user moves the tool away from the  
workpiece or when the user moves it slowly toward the workpiece.

To the best of the authors' knowledge, [21] is the only work dealing with  
human-robot cooperation that developed a method to facilitate a smooth ap-  
205 proach to the workpiece. In this application, the user teleoperated the robot  
by means of a haptic device while perceiving the *stiffness* of the target object  
and its surrounding areas, which were estimated using a force sensor and a 3D  
camera, respectively. However, this approach had several drawbacks compared

to the proposed method: the smooth approach to the target object was *not*  
210 *ensured*, since perceiving the aforementioned stiffness did not guarantee that  
the user performed proper teleoperation commands; the perceived stiffness de-  
pended only on the *tool position*, i.e., the user did not get feedback about the  
tool speed, which is also very relevant to perform a smooth approach to the tar-  
get object; and the perceived stiffness around the target object was estimated  
215 using a *local* sensor (i.e., a 3D camera mounted on the robot end-effector), which  
might suffer from occlusions and incomplete data.

In contrast, the proposed approach has the following advantages: the smooth  
approach to the target workpiece is guaranteed by an *approach control system*,  
which only uses degrees of freedom of the robot when it becomes active; not only  
220 the tool position but also the *tool speed* is considered to evaluate whether the  
robot approach control becomes active, i.e., the approach control can become  
active at any point of the workspace depending on the tool speed; and the  
position and speed of the tool relative to the target workpiece are obtained from  
a redundant *global* sensor system (i.e., the camera network mentioned above)  
225 and, hence, the problems related to incomplete information are avoided.

#### 1.3.4. *Boundary constraints*

Another distinctive feature of this work compared to the previous literature  
is that the proposed method confines the robot tool to an area close to the  
workpiece. Thus, when the operator guides the robot tool far away from the  
230 workpiece, a boundary constraint becomes active to prevent the tool from leav-  
ing the allowed area close to the workpiece. This approach, which represents a  
significant aid for the for the human operator when performing the surface treat-  
ment on the workpiece using the robot system, has two main advantages. On  
the one hand, unnecessary movements are avoided since the robot tool should  
235 not go far away from the workpiece while performing the surface treatment  
task. On the other hand, potential collisions between the robot tool and other  
objects within the robot workspace are prevented, i.e., the tool does not access  
forbidden areas of the robot environment where other objects are located.

### 1.3.5. Combination with automatic operation

240 Finally, another distinctive feature of this work compared to the previous literature is the combination of automatic operation together with manual operation, which adds more flexibility to the proposed method. Thus, the user is able to guide the tool of the robotic system to treat arbitrary regions of the workpiece (manual operation) and, when the user is not guiding the robot  
245 tool, a robot automatic operation is activated to perform the treatment in prior established regions.

### 1.4. Content of the article

Section 2 gives some theoretical basis used in this research, whereas Section 3 presents the proposed method. The implementation details of the algorithm  
250 are given in Section 4. Moreover, the feasibility of the proposal is proved in Section 5 with several experiments. Finally, Section 6 presents the conclusions of this work.

## 2. Theoretical basis

This section presents the theoretical basis used to develop the proposed  
255 approach in Section 3. 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  
260 flexibility and precision. The camera can be placed on the end-effector of the robot system (eye-in-hand configuration), e.g., see [22], or, alternatively, it can be placed on a structure to “observe” the whole workspace of the robot system (eye-to-hand configuration), e.g., see [23].

If a 2D camera is used, the depth can be estimated by processing the acquired  
265 data [24]. Alternatively, this estimation is not needed if a 3D camera is used [25], e.g., the Microsoft Kinect.

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 [26]. 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 Iterative Closest Point (ICP) method [27] to obtain the pose of the workpiece since it is widely used due to its effectiveness and simplicity.

## 2.2. Kinematics

The kinematics of the robot system can be expressed as:

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

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

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

being  $\mathbf{p} = [x \ y \ z \ \alpha \ \beta \ \gamma]^T$  the pose of the robotic system, where  $\alpha$ ,  $\beta$  and  $\gamma$  represent the orientation angles (roll, pitch and yaw, respectively),  $\mathbf{q} = [q_1 \ \dots \ q_n]^T$  the configuration of the robotic system,  $\mathbf{J}$  the Jacobian matrix and  $\mathbf{l}$  the so-called kinematic function [28].

## 2.3. Task prioritization method

This method is useful to address a set of tasks with different priorities [29], where the error of the task equations has to be minimized. The recursive equations of this strategy are given below [30]:

$$\mathbf{A}_i \bar{\mathbf{x}} = \mathbf{b}_i, \quad i = 1, \dots, M, \quad (4)$$

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

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

being  $M$  the considered number of equalites or tasks,  $\bar{\mathbf{x}}$  is the unknown vector to be computed,  $\mathbf{A}_i$  and  $\mathbf{b}_i$  the matrix and vector, respectively, for the  $i$ -th task ( $i = 1$  represents the highest priority),  $\bar{\mathbf{x}}_i$  the solution that hierarchically  
 285 minimizes the error of the first  $i$  task equations,  $\mathbf{N}_0 = \mathbf{I}$  the identity matrix,  $\bar{\mathbf{x}}_0 = \mathbf{0}$  the null vector and superscript  $\dagger$  the pseudoinverse of a matrix [31] (a threshold can be used to neglect the small singular values).

Note that, in this work, the unknown vector  $\bar{\mathbf{x}}$  to be computed corresponds to the commanded joint accelerations, i.e.,  $\ddot{\mathbf{q}}_c$ . Moreover,  $\bar{\mathbf{x}}_M$  denotes the so-  
 290 lution to this unknown vector for the  $M$  task equations, which is computed by the above task prioritization algorithm. Basically, this algorithm hierarchically minimizes the Least Square Error (LSE) of the task equations, i.e., a lower-priority task is satisfied only by using the degrees of freedom in the null space of the higher-priority ones. See [32] for further details.

#### 2.4. Non-conventional sliding mode control

Sliding Mode Control (SMC) [33] is widely utilized in robot tasks since it has the inherent advantages of robustness and low computation cost, e.g., see [34, 35], among others.

The modified SMC presented in [20] is used in this research both to limit the  
 300 approach speed of the robot tool and to confine the tool to an area close to the workpiece. In particular, this approach is useful to satisfy *inequality constraints* as discussed next.

Consider the inequality constraints and state equation below:

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

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

being vector  $\mathbf{x}$  the state<sup>1</sup>,  $N_{in}$  the number of constraints,  $\phi_{in,i}$  the  $i$ -th constraint

---

<sup>1</sup>Vector  $\mathbf{x}$ , which denotes as usual the state vector of a generic dynamical system, is defined in Section [3] in terms of the robot configuration  $\mathbf{q}$  and its derivatives. Note that variable  $\mathbf{x}$  is not directly related to variable  $\bar{\mathbf{x}}$ , which was used above to represent the unknown vector of the task equations.

function, which is assumed to be differentiable, vector  $\mathbf{u}$  the control action,  
 305 vector  $\mathbf{d}$  the disturbance,  $\mathbf{f}$  the drift vector field and  $\mathbf{g}$  a set of control vector  
 fields.

Then, the fulfillment of the constraints (7) is guaranteed if the value of the  
 control action  $\mathbf{u}$  fulfills the following expression [20]:

$$\text{v2dm}(\text{pos}(\boldsymbol{\phi}_{in})) \mathbf{L}_{\mathbf{g}} \boldsymbol{\phi}_{in} \mathbf{u} = -\text{pos}(\boldsymbol{\phi}_{in}) u_{in}^+, \quad (9)$$

where:  $\text{v2dm}(\cdot)$  is a function that returns a diagonal matrix from a vector;  
 $\text{pos}(\cdot)$  denotes the *positive function*<sup>2</sup>, that is, if  $x \leq 0$  then  $\text{pos}(x) = 0$ , otherwise  
 $\text{pos}(x) = 1$ ; column vector  $\boldsymbol{\phi}_{in}$  is composed of the constraint functions  $\phi_{in,i}$  of all  
 310 the inequalities;  $\mathbf{L}_{\mathbf{g}} \boldsymbol{\phi}_{in}$  is a matrix containing the row vectors  $\mathbf{L}_{\mathbf{g}} \phi_{in,i} = \frac{\partial \phi_{in,i}}{\partial \mathbf{x}} \mathbf{g}$   
 of all inequality constraints; and  $u_{in}^+$  represents the switching gain, which is a  
 high enough positive scalar.

See [20] for further details about the above SMC.

### 3. Proposed method

#### 3.1. General overview

Fig. 1 shows the general overview of the proposed method. In particu-  
 lar, three prioritized levels are considered to simultaneously accomplish several  
 tasks. The first level, i.e., the highest priority level, is utilized both to limit the  
 approach speed of the robot tool and to confine the tool to an area close to the  
 320 workpiece. The second level is included to ensure that the tool is perpendicular  
 to the workpiece surface. Lastly, the third level, i.e., the lowest priority level, is  
 utilized to conduct the surface treatment on a particular part of the workpiece  
 by means of a manual or an automatic mode of operation.

---

<sup>2</sup>Note that, when applying the positive function  $\text{pos}(\cdot)$  to a generic vector  $\mathbf{a}$  of  
 dimension  $r$ , the following result is obtained:  $\text{pos}(\mathbf{a}) = \text{pos}\left(\begin{bmatrix} a_1 & \dots & a_r \end{bmatrix}^T\right) =$   
 $\begin{bmatrix} \text{pos}(a_1) & \dots & \text{pos}(a_r) \end{bmatrix}^T$

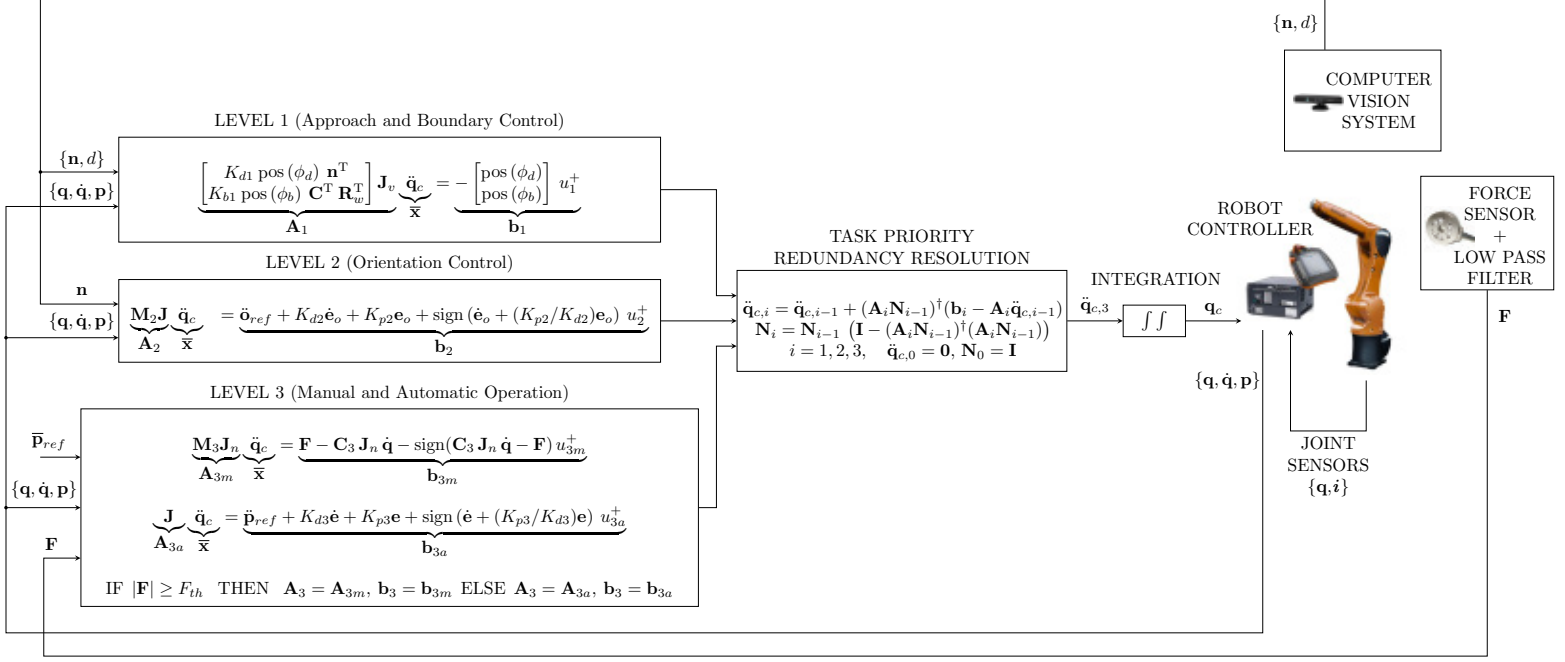
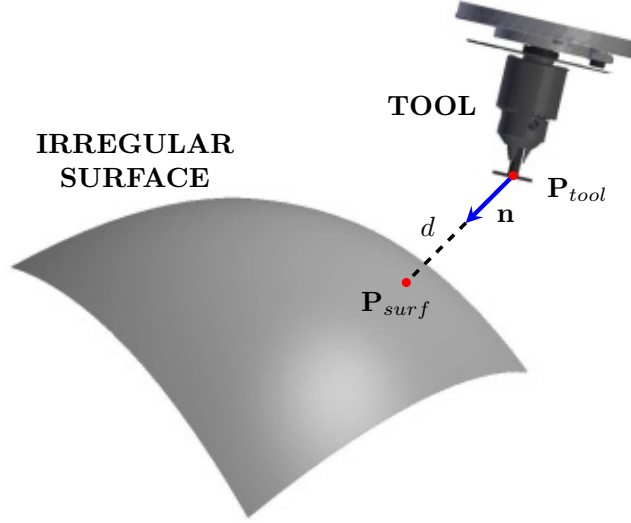


Fig. 1. Block diagram of the proposed method.

The following input information is considered for these levels: the robot  
 325 pose  $\mathbf{p}$  and the robot configuration and its derivative  $\{\mathbf{q}, \dot{\mathbf{q}}\}$ , which are obtained  
 from the robot controller; the force vector  $\mathbf{F}$  obtained from the *guidance sensor*,  
 which is located at the robot tool; the data  $\{d, \mathbf{n}\}$  obtained using machine vision,  
 where  $d$  represents the length of the vector from the robot tool, see Fig. 2 to  
 the nearest point of the workpiece, whereas  $\mathbf{n}$  denotes the unit vector of the  
 330 mentioned vector (note that  $\mathbf{n}$  is normal to the workpiece surface as long as it  
 is smooth at the nearest point to the tool); and the reference  $\bar{\mathbf{p}}_{ref}$  for the tool  
 position  $\bar{\mathbf{p}} = \begin{bmatrix} x & y & z \end{bmatrix}^T$ .

The equation  $\mathbf{A}_i \bar{\mathbf{x}} = \mathbf{b}_i$  (4) for each priority level is obtained below, where  $\bar{\mathbf{x}}$   
 corresponds to the commanded acceleration  $\ddot{\mathbf{q}}_c$  for the robot system. The errors  
 335 of these equations are minimized using (5) and (6), as shown in Fig. 1. Thus, the  
 acceleration command  $\ddot{\mathbf{q}}_{c,3}$  is double integrated to get the robot configuration  
 command  $\mathbf{q}_c$ . Finally, the robot controller defines an inner control loop to track





**Fig. 2.** Graphical illustration of the information obtained from the machine vision system.

the commanded values taking into account the measurements of the joint angles  $\mathbf{q}$  and joint currents  $\mathbf{i}$ . Hence,  $\mathbf{d}_c$  stands for the inaccuracy of this inner loop, which is assumed to be bounded. However, note that the value of  $\mathbf{d}_c$  does not need to be computed nor estimated since the proposed SMC-based controller, which is detailed below, is inherently robust against this error.

### 3.2. Level 1: Approach and boundary control

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

$$\phi_d = \epsilon_d - d - K_{d1}\dot{d} \leq 0, \quad (10)$$

where  $\epsilon_d$  represents a security margin between the tip of the tool and the workpiece surface and  $K_{d1}$  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 (10) is readily obtained from the robot velocity as detailed below:

$$\begin{aligned} \dot{d} &= (\partial d / \partial \mathbf{q})^T \dot{\mathbf{q}} = \left( (\partial \bar{\mathbf{p}} / \partial \mathbf{q})^T (\partial d / \partial \bar{\mathbf{p}}) \right)^T \dot{\mathbf{q}} \\ &= (\mathbf{J}_v^T (-\mathbf{n}))^T \dot{\mathbf{q}} = -\mathbf{n}^T \mathbf{J}_v \dot{\mathbf{q}}, \end{aligned} \quad (11)$$

where matrix  $\mathbf{J}_v$  represents the top  $3 \times 3$  submatrix of the Jacobian  $\mathbf{J}$ .

In order to confine the robot tool to an area close to the workpiece, the following constraint is considered:

$$\begin{aligned} \phi_b &= \sigma_b + K_{b1} \dot{\sigma}_b \leq 0 \\ \sigma_b &= -1 + \left( \left| \frac{x_w - x_c}{W} \right|^m + \left| \frac{y_w - y_c}{H} \right|^m + \left| \frac{z_w - z_c}{M} \right|^m \right), \end{aligned} \quad (12)$$

350 where  $\sigma_b$  defines the boundary of the allowed area for the tool position as a superellipse, which looks like a rectangular prism with rounded corners,  $m$  is a design parameter that establishes the rounding of the prism corners, parameters  $W$ ,  $H$  and  $M$  define the length of each side of the prism,  $\bar{\mathbf{p}}_w = [x_w \ y_w \ z_w]^T$  is the tool position relative to the workpiece coordinate system,  $\bar{\mathbf{p}}_c = [x_c \ y_c \ z_c]^T$  is the center of the superellipse<sup>3</sup> relative to the  
355 workpiece coordinate system and  $K_{b1}$  is a free design parameter in order to limit (similarly to the approach constraint) the approach speed of the tool to the boundary of the allowed area, i.e., the superellipse.

As before, considering that the motion of the workpiece is significantly slower than the motion of the robot system, the derivative of  $\sigma_b$  in (12) is obtained

---

<sup>3</sup>It has been assumed that the orientation of the superellipse matches the orientation of the workpiece coordinate system. However, if that would not be the case, the formula of the boundary constraint could be easily modified. Details omitted for brevity.

from the robot velocity as detailed below:

$$\begin{aligned}\dot{\sigma}_b &= (\partial\sigma_b/\partial\mathbf{q})^T \dot{\mathbf{q}} = \left( (\partial\bar{\mathbf{p}}_w/\partial\mathbf{q})^T (\partial\sigma_b/\partial\bar{\mathbf{p}}_w) \right)^T \dot{\mathbf{q}} \\ &= \left( (\mathbf{R}_w^{-1}\mathbf{J}_v)^T \mathbf{C} \right)^T \dot{\mathbf{q}} = \mathbf{C}^T \mathbf{R}_w^T \mathbf{J}_v \dot{\mathbf{q}},\end{aligned}\quad (14)$$

where  $\mathbf{R}_w$  is the rotation matrix of the workpiece coordinate system with respect to the robot base coordinate system (note that the inverse of a rotation matrix is its transpose) and matrix  $\mathbf{C}$  is given by:

$$\mathbf{C} = \begin{bmatrix} \frac{m \operatorname{sign}(x_w - x_c) |x_w - x_c|^{m-1}}{W^m} \\ \frac{m \operatorname{sign}(y_w - y_c) |y_w - y_c|^{m-1}}{H^m} \\ \frac{m \operatorname{sign}(z_w - z_c) |z_w - z_c|^{m-1}}{M^m} \end{bmatrix}. \quad (15)$$

In order to use the SMC detailed in Section 2.4 to satisfy the approach constraint in (10) and the boundary constraint in (12), the following second-order dynamical system (8) is considered:

$$\dot{\mathbf{x}} = \begin{bmatrix} \mathbf{O} & \mathbf{I} \\ \mathbf{O} & \mathbf{O} \end{bmatrix} \mathbf{x} + \mathbf{d} + \begin{bmatrix} \mathbf{O} \\ \mathbf{I} \end{bmatrix} \mathbf{u}, \quad (16)$$

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

From Eqs. (9), (10) and (12), the control equation for Level 1 results in:

$$\begin{aligned} \begin{bmatrix} \operatorname{pos}(\phi_d) & 0 \\ 0 & \operatorname{pos}(\phi_b) \end{bmatrix} \mathbf{L}_g \phi_1 \ddot{\mathbf{q}}_c &= - \begin{bmatrix} \operatorname{pos}(\phi_d) \\ \operatorname{pos}(\phi_b) \end{bmatrix} u_1^+, \\ &\rightarrow \mathbf{A}_1 \ddot{\mathbf{q}}_c = \mathbf{b}_1, \end{aligned}\quad (17)$$

where  $u_1^+$  represents switching gain of the SMC,  $\mathbf{b}_1$  and  $\mathbf{A}_1$  denote the vector

and matrix for the control equation of Level 1 and, according to (10)–(16), matrix  $\mathbf{L}_g\phi_1$  is given by:

$$\begin{aligned}\mathbf{L}_g\phi_1 &= \begin{bmatrix} (\partial\phi_d/\partial\mathbf{x})^\top \\ (\partial\phi_b/\partial\mathbf{x})^\top \end{bmatrix} \mathbf{g} = \begin{bmatrix} (\partial\phi_d/\partial\dot{\mathbf{q}})^\top \\ (\partial\phi_b/\partial\dot{\mathbf{q}})^\top \end{bmatrix} \\ &= \begin{bmatrix} -K_{d1} (\partial\dot{d}/\partial\dot{\mathbf{q}})^\top \\ K_{b1} (\partial\dot{\sigma}_b/\partial\dot{\mathbf{q}})^\top \end{bmatrix} = \begin{bmatrix} K_{d1} \mathbf{n}^\top \\ K_{b1} \mathbf{C}^\top \mathbf{R}_w^\top \end{bmatrix} \mathbf{J}_v.\end{aligned}\quad (18)$$

### 3.3. Level 2: 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. 2) must point in the direction of  $\mathbf{n}$ . Thus, the reference for the tool orientation is vector  $\mathbf{n}$ , which can be easily transformed [36] to roll and pitch reference values, i.e.,  $\alpha_{ref}$  and  $\beta_{ref}$ . It is worth noting that there is no requirement for the yaw angle and, hence, it can be used, for instance, for tool guidance, see Section 3.4.1.

Thus, the control equation for Level 2 results in:

$$\begin{aligned}\mathbf{M}_2\mathbf{J}\ddot{\mathbf{q}}_c &= \ddot{\mathbf{o}}_{ref} + K_{d2}\dot{\mathbf{e}}_o + K_{p2}\mathbf{e}_o + \text{sign}(\dot{\mathbf{e}}_o + (K_{p2}/K_{d2})\mathbf{e}_o) u_2^+ \\ &\rightarrow \mathbf{A}_2\ddot{\mathbf{q}}_c = \mathbf{b}_2,\end{aligned}\quad (19)$$

where matrix  $\mathbf{M}_2 = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$  is used to affect only  $\alpha$  and  $\beta$  angles

(i.e., roll and pitch) of the pose vector  $\mathbf{p}$ ; vector  $\mathbf{o}_{ref} = [\alpha_{ref} \ \beta_{ref}]^\top$  represents the reference orientation; vector  $\mathbf{e}_o = \mathbf{o}_{ref} - [\alpha \ \beta]^\top$  denotes the error of roll and pitch angles;  $K_{p2}$  and  $K_{d2}$  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 equations in (2));  $u_2^+$  represents a switching gain; and  $\mathbf{b}_2$  and  $\mathbf{A}_2$  denote the vector and matrix for the control equation of Level 2.

It is worth noting that (19) represents a hybrid controller, where the last switching term is used to cancel out the last term in (3) and, hence, the compu-

tation of the Jacobian derivative is avoided. Note that this hybrid controller in sort uses conventional SMC, whose proof of convergence can be found in [33].

### 3.4. Level 3: Modes of operation

380 This level is included to conduct the surface treatment on a particular part of the workpiece by means of a manual or an automatic mode of operation. Specifically, if the guidance sensor detects a significant force value ( $|\mathbf{F}| \geq F_{th}$ , where  $F_{th}$  represents a threshold) the manual operation becomes active, i.e., the user guides the robot tool using the guidance sensor. Otherwise, the automatic  
385 operation is activated, and the robot tool follows a reference value. The control equations for both cases are detailed below.

Note that this level controls the tool yaw angle and the tool position, either by the manual or the automatic mode of operation, whereas the tool pitch and roll angles are established in Level 2.

#### 390 3.4.1. Manual operation

In a similar way to other works that use human-robot interaction [37, 38], the human operator guides the tool exerting forces that are transformed by the following admittance controller into the desired values of the tool speed:

$$\mathbf{M}_3 \dot{\mathbf{v}}_n + \mathbf{C}_3 \mathbf{v}_n = \mathbf{F}, \quad (20)$$

where vector  $\mathbf{v}_n = \mathbf{J}_n \dot{\mathbf{q}}$  denotes the tool speed with respect to the tool coordinate system, matrix  $\mathbf{J}_n$  represents the *geometric* Jacobian [36] with respect to the tool coordinate system, vector  $\mathbf{F}$  contains the force measurements of the guidance sensor with respect to the tool coordinate system and the controller  
395 gains  $\mathbf{C}_3$  and  $\mathbf{M}_3$  are  $6 \times 6$  diagonal matrices representing the virtual damping and inertia, respectively.

Taking into account that  $\mathbf{v}_n = \mathbf{J}_n \dot{\mathbf{q}}$ , Eq. (20) results in:

$$\mathbf{M}_3 \mathbf{J}_n \ddot{\mathbf{q}} = \mathbf{F} - \mathbf{C}_3 \mathbf{J}_n \dot{\mathbf{q}} - \mathbf{M}_3 \dot{\mathbf{J}}_n \dot{\mathbf{q}}. \quad (21)$$

Similarly to Level 2, the computation of the Jacobian derivative in (21) can be avoided using the following hybrid controller:

$$\begin{aligned} \mathbf{M}_3 \mathbf{J}_n \ddot{\mathbf{q}}_c &= \mathbf{F} - \mathbf{C}_3 \mathbf{J}_n \dot{\mathbf{q}} - \text{sign}(\mathbf{C}_3 \mathbf{J}_n \dot{\mathbf{q}} - \mathbf{F}) u_{3m}^+ \\ &\rightarrow \mathbf{A}_{3m} \ddot{\mathbf{q}}_c = \mathbf{b}_{3m}, \end{aligned} \quad (22)$$

where  $u_{3m}^+$  represents a switching gain and  $\mathbf{b}_{3m}$  and  $\mathbf{A}_{3m}$  denote the vector and matrix for the control equation of Level 3 when the manual operation is active.

### 3.4.2. Automatic operation

In a similar way to (19), the following controller is used to follow the reference pose  $\mathbf{p}_{ref}$  when the automatic operation is active:

$$\begin{aligned} \mathbf{J} \ddot{\mathbf{q}}_c &= \ddot{\mathbf{p}}_{ref} + K_{d3} \dot{\mathbf{e}} + K_{p3} \mathbf{e} + \text{sign}(\dot{\mathbf{e}} + (K_{p3}/K_{d3})\mathbf{e}) u_{3a}^+ \\ &\rightarrow \mathbf{A}_{3a} \ddot{\mathbf{q}}_c = \mathbf{b}_{3a} \end{aligned} \quad (23)$$

400 where  $\mathbf{e} = \mathbf{p}_{ref} - \mathbf{p}$  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;  $u_{3a}^+$  represents a switching gain; and  $\mathbf{b}_{3a}$  and  $\mathbf{A}_{3a}$  denote the vector and matrix for the control equation of Level 3 when the automatic operation is active.

During the automatic operation the tool yaw angle is kept still, i.e., its  
405 reference value  $\gamma_{ref}$  corresponds to the angle value at the moment the automatic operation was activated. Moreover, the reference value  $\bar{\mathbf{p}}_{ref}$  for the tool position is established according to the following four stages that are cyclically repeated to treat each point on the workpiece surface:

1. The robot tool is placed at a certain distance from the point on the work-  
410 piece surface using a fast approach trajectory.
2. The tool is taken to the point on the workpiece surface using a slow linear trajectory.
3. The tool is kept still during a time lapse to ensure that the surface treatment is properly completed.

415 4. The robot tool is moved away from the workpiece surface using a slow linear trajectory.

Since the above cycle can be interrupted by the manual operation, when the robot system goes back to automatic operation, it resumes the stage where it was before the interruption.

## 420 4. Control algorithm

### 4.1. Code of the control algorithm

Table I shows the pseudo-code of the algorithm developed in this work.

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<sup>4</sup>. LC2 is used to compute the orientation angles of  $\mathbf{n}$ , which is provided by the computer vision. LC3 and LC4  
425 are used to compute the robot pose and its derivative from the robot kinematics. LC5 to LC7 are used to compute the constraint functions of the inequalities in Level 1. LC8, LC9, LC10 and LC11 are used to compute the time derivative of several signals. LC12 to LC15 are used to compute the orientation and pose  
430 errors and their derivatives (the automatic operation provides the reference pose  $\mathbf{p}_{ref}$  using the procedure outlined in Section 3.4.2). LC16 and LC17 are used to compute the matrix and vector, respectively, for the control equation of Level 1. LC18 and LC19 are used to compute the matrix and vector, respectively, for the control equation of Level 2. LC20 to LC26 are used to compute the matrix and  
435 vector for the control equation of Level 3. LC27 to LC31 are used to compute the solution of the commanded accelerations that minimizes the errors of the control equations of the three levels. LC32 and LC33 are used to compute the commanded positions by integrating twice the commanded accelerations. LC34 is used to send the commanded values to the robot controller. LC35 to LC38

---

<sup>4</sup>It is assumed that the electronics of the F/T sensor (guidance sensor) has filtered the force measurements and that all disturbance forces have already been compensated, e.g., the weight and inertia of the robot tool.

Table 1. Code of the algorithm

---

Executed every sampling time  $T_s$

---

```

1  $[\mathbf{q}, \dot{\mathbf{q}}, \mathbf{F}, \mathbf{n}, d] = \text{GetRobotStateAndForcesAndVisionData};$ 
2  $\mathbf{o}_{ref} = \text{OrientationOfVector}(\mathbf{n});$ 
3  $\mathbf{p} = \mathbf{l}(\mathbf{q});$  // Eq. (1)
4  $\dot{\mathbf{p}} = \mathbf{J}\dot{\mathbf{q}};$  // Eq. (2)
5  $\phi_d = \epsilon_d - d + K_{d1} \mathbf{n}^T \mathbf{J}_v \dot{\mathbf{q}};$  // Eqs. (10),(11)
6  $\sigma_b = -1 + \left( \left| \frac{x_w - x_c}{W} \right|^m + \left| \frac{y_w - y_c}{H} \right|^m + \left| \frac{z_w - z_c}{M} \right|^m \right);$  // Eq. (13)
7  $\phi_b = \sigma_b + K_{b1} \mathbf{C}^T \mathbf{R}_w^T \mathbf{J}_v \dot{\mathbf{q}};$  // Eqs. (12),(14)
8  $\dot{\mathbf{o}}_{ref} = (\mathbf{o}_{ref} - \mathbf{o}_{ref,prev})/T_s;$  // Derivative
9  $\ddot{\mathbf{o}}_{ref} = (\dot{\mathbf{o}}_{ref} - \dot{\mathbf{o}}_{ref,prev})/T_s;$  // Derivative
10  $\dot{\mathbf{p}}_{ref} = (\mathbf{p}_{ref} - \mathbf{p}_{ref,prev})/T_s;$  // Derivative
11  $\ddot{\mathbf{p}}_{ref} = (\dot{\mathbf{p}}_{ref} - \dot{\mathbf{p}}_{ref,prev})/T_s;$  // Derivative
12  $\mathbf{e}_o = (\mathbf{o}_{ref} - \mathbf{M}_2 \mathbf{p});$  // Orientation error
13  $\dot{\mathbf{e}}_o = (\dot{\mathbf{o}}_{ref} - \mathbf{M}_2 \dot{\mathbf{p}});$  // Derivative of orientation error
14  $\mathbf{e} = (\mathbf{p}_{ref} - \mathbf{p});$  // Pose error
15  $\dot{\mathbf{e}} = (\dot{\mathbf{p}}_{ref} - \dot{\mathbf{p}});$  // Derivative of pose error
16  $\mathbf{A}_1 = \begin{bmatrix} K_{d1} \text{pos}(\phi_d) \mathbf{n}^T \\ K_{b1} \text{pos}(\phi_b) \mathbf{C}^T \mathbf{R}_w^T \end{bmatrix} \mathbf{J}_v;$  // Eqs. (17),(18)
17  $\mathbf{b}_1 = - \begin{bmatrix} \text{pos}(\phi_d) \\ \text{pos}(\phi_b) \end{bmatrix} u_1^+;$  // Eq. (17)
18  $\mathbf{A}_2 = \mathbf{M}_2 \mathbf{J};$  // Eq. (19)
19  $\mathbf{b}_2 = \ddot{\mathbf{o}}_{ref} + K_{d2} \dot{\mathbf{e}}_o + K_{p2} \mathbf{e}_o + \text{sign}(\dot{\mathbf{e}}_o + (K_{p2}/K_{d2}) \mathbf{e}_o) u_2^+;$  // Eq. (19)
20 if  $\mathbf{F} \geq F_{th}$  then
21  $\mathbf{A}_3 = \mathbf{M}_3 \mathbf{J}_n;$  // Eq. (22)
22  $\mathbf{b}_3 = \mathbf{F} - \mathbf{C}_3 \mathbf{J}_n \dot{\mathbf{q}} - \text{sign}(\mathbf{C}_3 \mathbf{J}_n \dot{\mathbf{q}} - \mathbf{F}) u_{3m}^+;$  // Eq. (22)
23 else
24  $\mathbf{A}_3 = \mathbf{J};$  // Eq. (23)
25  $\mathbf{b}_3 = \ddot{\mathbf{p}}_{ref} + K_{d3} \dot{\mathbf{e}} + K_{p3} \mathbf{e} + \text{sign}(\dot{\mathbf{e}} + (K_{p3}/K_{d3}) \mathbf{e}) u_{3a}^+;$  // Eq. (23)
26 end
27  $\ddot{\mathbf{q}}_{c,1} = \mathbf{A}_1^\dagger \mathbf{b}_1;$  // Eq. (5),  $i = 1$ 
28  $\mathbf{N}_1 = \mathbf{I} - \mathbf{A}_1^\dagger \mathbf{A}_1;$  // Eq. (6),  $i = 1$ 
29  $\ddot{\mathbf{q}}_{c,2} = \ddot{\mathbf{q}}_{c,1} + (\mathbf{A}_2 \mathbf{N}_1)^\dagger (\mathbf{b}_2 - \mathbf{A}_2 \ddot{\mathbf{q}}_{c,1});$  // Eq. (5),  $i = 2$ 
30  $\mathbf{N}_2 = \mathbf{N}_1 (\mathbf{I} - (\mathbf{A}_2 \mathbf{N}_1)^\dagger (\mathbf{A}_2 \mathbf{N}_1));$  // Eq. (6),  $i = 2$ 
31  $\ddot{\mathbf{q}}_{c,3} = \ddot{\mathbf{q}}_{c,2} + (\mathbf{A}_3 \mathbf{N}_2)^\dagger (\mathbf{b}_3 - \mathbf{A}_3 \ddot{\mathbf{q}}_{c,2});$  // Eq. (5),  $i = 3$ 
32  $\dot{\mathbf{q}}_c = \dot{\mathbf{q}}_{c,prev} + \ddot{\mathbf{q}}_{c,3} T_s;$  // Integration
33  $\mathbf{q}_c = \mathbf{q}_{c,prev} + \dot{\mathbf{q}}_c T_s;$  // Integration
34 SendToJointControllers( $\mathbf{q}_c$ );
35  $\mathbf{o}_{ref,prev} = \mathbf{o}_{ref};$  // For next iteration
36  $\dot{\mathbf{o}}_{ref,prev} = \dot{\mathbf{o}}_{ref};$  // For next iteration
37  $\mathbf{p}_{ref,prev} = \mathbf{p}_{ref};$  // For next iteration
38  $\dot{\mathbf{p}}_{ref,prev} = \dot{\mathbf{p}}_{ref};$  // For next iteration

```

---



440 are used to update the previous value of several signals in order to use them in  
the next iteration.

Note that the kinematic function and jacobian matrices of the robot arm can  
be readily obtained from its Denavit-Hartenberg parameters as detailed in [39].

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

#### 4.2. Design of the control algorithm parameters

The practical guidelines to choose the parameters of the control algorithm  
are summarized in the following steps:

1<sup>st</sup>) The sampling period  $T_s$  is chosen as small as possible but ensuring: that  
450 the noise introduced by the numerical derivatives in Table 1 is negligible;  
and that the SMC frequency  $f_{SMC} = (2T_s)^{-1}$  is lower than the bandwidth  
of the low-level joint controllers (otherwise, the SMC actions would not  
be properly “followed” by the robot system).

2<sup>nd</sup>a) The bandwidth of the kinematic control performed in Level 1 (given by  
455  $K_{d1}$  and  $K_{b1}$ ), Level 2 (given by  $K_{p2}$  and  $K_{d2}$ ) and Level 3 (given by  
 $\mathbf{M}_3$  and  $\mathbf{C}_3$  or  $K_{p3}$  and  $K_{d3}$ ) should be significantly lower than the SMC  
frequency  $f_{SMC}$  for stability reasons.

2<sup>nd</sup>b) For Level 1, the parameters  $K_{d1}$  and  $K_{b1}$  are chosen small enough to reduce  
the chattering effect of the SMC, but satisfying the condition indicated in  
460 step 2<sup>nd</sup>a).

2<sup>nd</sup>c) For Level 2, the parameters  $K_{p2}$  and  $K_{d2}$  are chosen large enough to  
obtain a fast and damped enough response, respectively, but satisfying  
the condition indicated in step 2<sup>nd</sup>a).

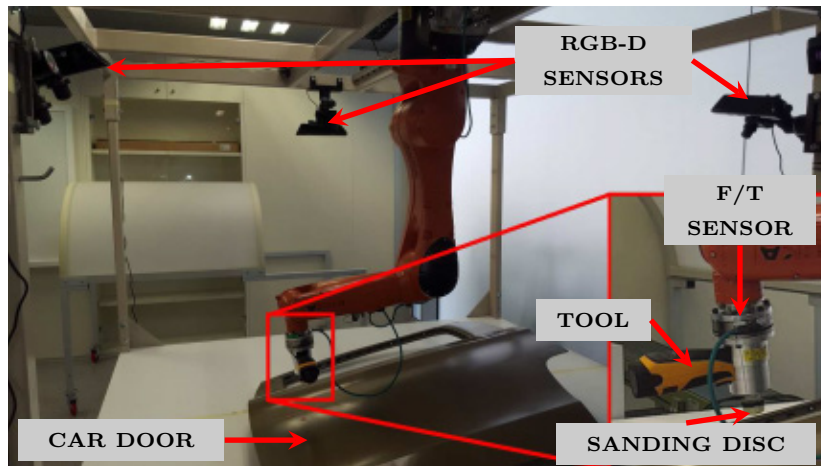
2<sup>nd</sup>d) For the automatic operation in Level 3, the parameters  $K_{p3}$  and  $K_{d3}$   
465 are chosen large enough to obtain a fast and damped enough response,  
respectively, but satisfying the condition indicated in step 2<sup>nd</sup>a).

- 2<sup>nd</sup>e) For the manual operation in Level 3, firstly the diagonal elements of matrix  $\mathbf{C}_3$  are chosen small enough to obtain the desired sensitivity of the tool guidance and, subsequently, the diagonal elements of matrix  $\mathbf{M}_3$  are chosen small enough to obtain a fast enough response. Moreover,  $\mathbf{M}_3$  and  $\mathbf{C}_3$  must satisfy the condition indicated in step 2<sup>nd</sup>a).
- 3<sup>rd</sup>) For all three levels, the switching gains  $\{u_1^+, u_2^+, u_{3m}^+, u_{3a}^+\}$  are empirically tuned to be as small as possible to alleviate the chattering effect of the SMC, but ensuring that the sliding mode behavior remains effective.
- 4<sup>th</sup>a) For Level 1, the parameters  $\{m, W, H, M\}$  of the superellipse are chosen to fit the allowed area for the specific robot application at hand, whereas the parameter  $\epsilon_d$  of the approach constrain is chosen large enough to cater for possible inaccuracies in the robot control.
- 4<sup>th</sup>b) For the manual operation in Level 3, the parameter  $F_{th}$  should be small enough to properly activate the tool guidance when the user is guiding the robot tool, but not too small, since that could cause undesired activations due to force sensor noise.
- 4<sup>th</sup>c) For the automatic operation in Level 3, the time lapse used to ensure that the surface treatment is properly completed has to be established depending on the requirements of the actual surface treatment task, where the knowledge of an experienced operator could be very useful.

## 5. Experiments

### 5.1. Experimental platform

The experimental platform used in this work is shown in Fig. 3, which was composed of: a 6R robot arm (Kuka KR6 Agilus), whose Denavit-Hartenberg parameters are shown in Table 2; a tool consisting of a spot repair sander (Mirka AROS-B 150NV) placed at the robot end-effector using a self-developed adapter; an Axia80 F/T (Force/Torque) sensor used as guidance sensor, which



**Fig. 3.** Experimental platform used for the real experimentation: a 6R robot arm, a F/T sensor, 3 RGB-D cameras, an industrial sander and a car door.

**Table 2.** Denavit-Hartenberg parameters for the robot used in the experiments ( $d_{tool} = 0.08$ )

Link $i$	$\theta_i$ (rad)	$d_i$ (m)	$a_i$ (m)	$\alpha_i$ (rad)
1	$q_1$	-0.4	0.025	$\pi/2$
2	$q_2$	0	-0.455	0
3	$q_3$	0	-0.035	$-\pi/2$
4	$q_4$	-0.42	0	$\pi/2$
5	$q_5$	0	0	$-\pi/2$
6	$q_6$	$-0.08 - d_{tool}$	0	$\pi$

is attached between the end-effector of the robot arm and the sander; a cylinder  
 495 of 29x29x23 mm used as sanding disc; 3 RGB-D cameras (Microsoft Kinect);  
 and a workpiece consisting of a car door.

An external computer was used to implement the algorithm detailed in Sec-  
 tion 4.1. Moreover, the robot arm, F/T sensor and external computer commu-  
 nicated by means of an Ethernet switch. In addition, the RGB-D cameras were  
 500 connected to serial ports of the external computer.

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

505 *5.2. Values of the parameters*

The values used for the control algorithm parameters are given below. They were established according to the practical guidelines given in Section 4.2

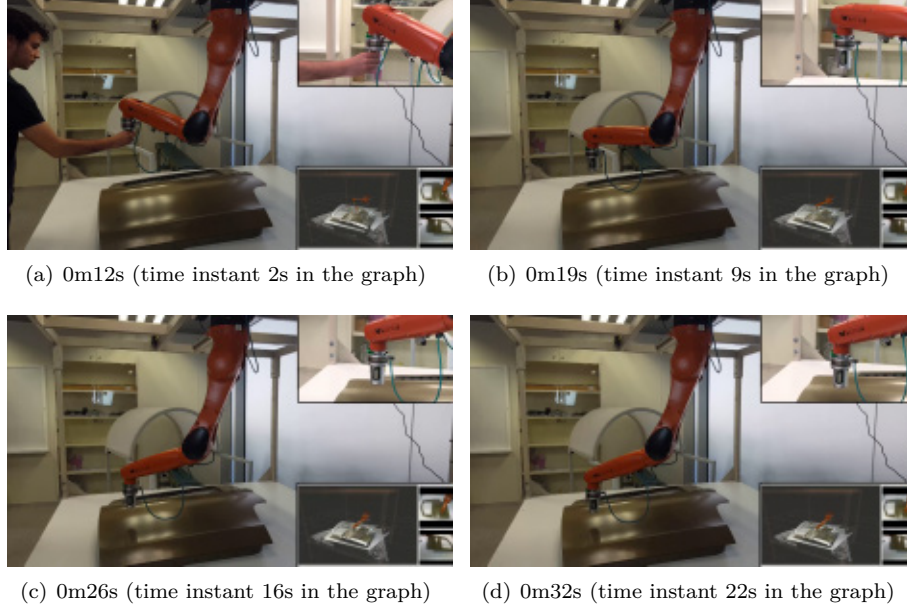
- i) Sampling period:  $T_s = 0.01$  s.
- ii) Parameters of Level 1 (Section 3.2):  $\epsilon_d = 2$  mm,  $K_{d1} = 2.5$ ,  $m = 4$ ,  
510  $W = 0.2$ ,  $H = 0.25$ ,  $M = 0.35$ ,  $K_{b1} = 1.6$ , and  $u_1^+ = 0.65$ .
- iii) Parameters of Level 2 (Section 3.3):  $K_{p2} = 1.5$ ,  $K_{d2} = 1.8$  and  $u_2^+ = 0.01$ .
- iv) Parameters of Level 3 (Section 3.4):  $\mathbf{M}_3 = 10\mathbf{I}$ ,  $\mathbf{C}_3 = 70\mathbf{I}$ ,  $F_{th} = 1$ ,  
 $u_{3m}^+ = 0.01$ ,  $u_{3a}^+ = 0.01$ ,  $K_{p3} = 2$  and  $K_{d3} = 4.2$ .

*5.3. Results*

515 Three experiments were conducted to study the behavior of: the approach constraint, the boundary constraint and the combination of manual and automatic modes of operation, respectively. Hence, for the first two experiments only the manual operation was considered.

Therefore, in order to analyze the behavior of the robot approaching the  
520 surface of the workpiece (i.e., the car door) a first experiment was conducted, see the video 40 (in order to obtain a clearer view, the sander was detached from the robotic arm end-effector). Several frames of this video recording are shown in Fig. 4: at around 12s, see Fig. 4(a), the human operator places a weight of about 0.175 Kg in the robotic arm end-effector and, thus, the tool  
525 guidance of the manual operation becomes active; in the interval 19s–26s, see Fig. 4(b) and Fig. 4(c), the end-effector of the robotic arm goes down towards the workpiece while its vertical speed is progressively reduced; and at around 32s, see Fig. 4(d), the robot arm stops its movement, keeping the security distance with the workpiece surface.

530 Next, several graphs are presented to show the quantitative performance of the first experiment. In particular, the distance of separation between the end-effector of the robotic arm and the workpiece is initially around 250mm,



**Fig. 4.** Frames of the video of the first experiment.

see the middle graph in Fig. 5(a). Then, at around 2.3s the operator places  
the small weight in the end-effector of the robotic arm, see the negative vertical  
535 force  $F_z$  detected by the guidance sensor in the top graph of Fig. 5(a), and the  
aforementioned separation is progressively reduced. Note that the reduction  
rate is roughly constant during the interval 2.3s–6.3s, see the middle graph of  
Fig. 5(a). This is because the tool guidance of the manual operation in Level 3  
converts the small weight, which is constant, to a downward speed  $v_{n,z}$  for  
540 the end-effector of the robotic arm, as shown in the top graph of Fig. 5(a).  
Subsequently, as shown in the bottom graph of Fig. 5(a) at about 6.3s the  
approach constraint becomes active. From then, the distance reduction rate  
and the negative vertical velocity  $v_{n,z}$  of the end-effector of the robotic arm  
are progressively reduced due to the approach constraint in Level 1. Finally,  
545 at about 22s the robot arm approximately reaches the security distance, i.e.,  
 $d = \epsilon_d$ , and stops its movement. A graph relating the distance and the distance  
reduction rate is shown in Fig. 5(b), where it can be appreciated that the system

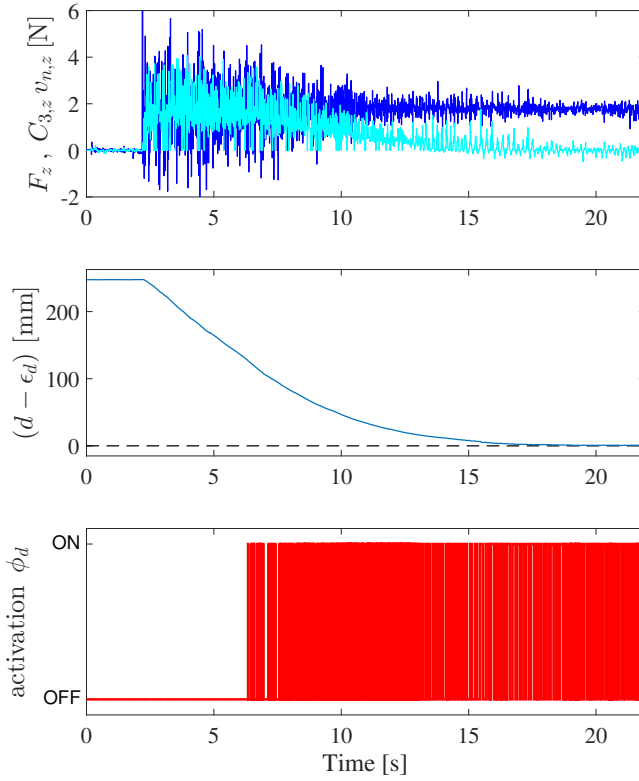
has a switching behavior (as usual in SMC) around the boundary of the approach constraint, which is obtained from (10) as  $\dot{d} = -(1/K_2)(d - \epsilon_d)$ .

550 The tool orientation angles for the first experiment are shown in Fig. 6. In particular, the first and second graphs show that the values of roll and yaw angles are very similar to the reference values supplied by the machine vision system, which is due to the orientation control implemented in Level 2. Moreover, the yaw angle is approximately constant, see the bottom graph. This is due to the  
555 fact that the small weight placed in the end-effector of the robotic arm produces no torque in the  $Z$ -axis of the end-effector of the robotic arm and, hence, no guidance is performed for the yaw angle in Level 3.

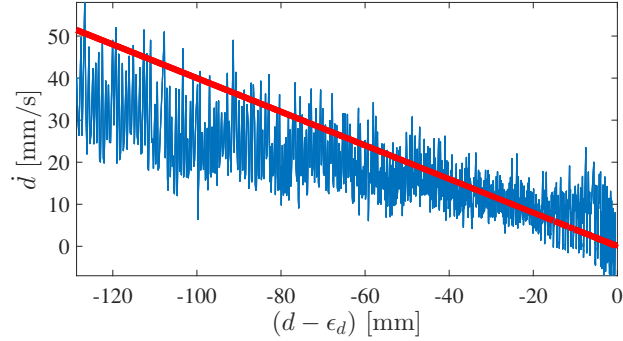
The Cartesian position of the end-effector of the robotic arm in the first experiment is represented in Fig. 7, where it can be noted that the trajectory  
560 followed by the robot end-effector is approximately a vertical straight line of about 0.25m.

The control signals in the first experiment are presented in Fig. 8, where the commanded accelerations computed by each control level are shown. In particular, the contribution of Level 2 is very little since only small corrections  
565 are required to keep the end-effector of the robotic arm perpendicular to the workpiece surface as it moves downwards. Moreover, the contribution of Level 1 starts when the approach constraint becomes active, which occurs at around 6.3s. Furthermore, the contribution of Level 3 starts at about 2.3s, i.e., when the human operator places the small weight in the end-effector of the robotic  
570 arm.

A second experiment was performed to analyze the behavior of the boundary constraint, see the video 41. Several frames of this video recording are shown in Fig. 9: at around 16s, see Fig. 9(a), the human operator is guiding the robotic arm tool close to the workpiece; at around 26s, see Fig. 9(b), the human operator  
575 moves up the robot tool and the boundary constraint becomes active, preventing the tool from leaving the allowed area, i.e., the rectangular prism with rounded corners relative to the workpiece; at around 1m09s, see Fig. 9(c), the workpiece (i.e., the car door) is relocated; and at around 1m33s, see Fig. 9(d), the human

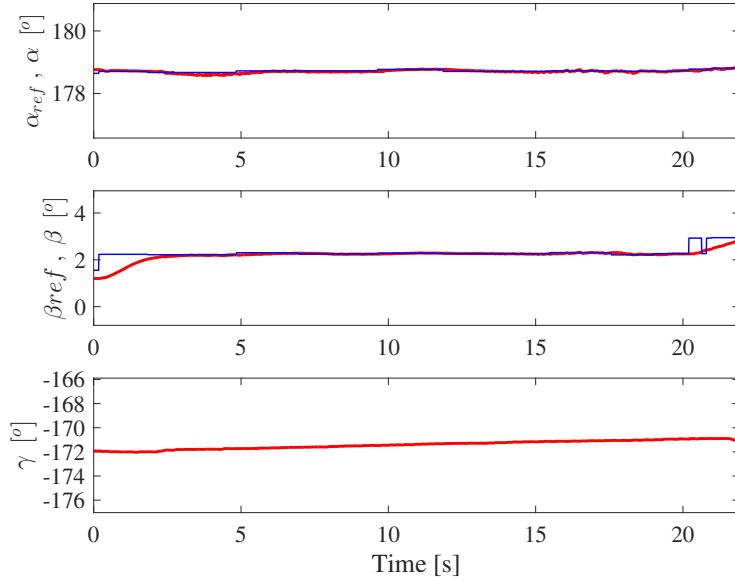


(a) Top graph: dark-blue line, negative vertical force  $F_z$  detected by the guidance sensor; and light-cyan line, negative vertical velocity  $v_{n,z}$  of the robot end-effector multiplied by the virtual damping coefficient  $C_{3,z}$  (note that the unit of the velocity is m/s, whereas the unit of the velocity multiplied by the damping coefficient is Newton, see Section 3.4.1 and Eq. (20)). Middle graph: distance of separation between the end-effector of the robotic arm and the workpiece surface. Bottom graph: activation of the approach constraint.



(b) Distance reduction rate of the robot end-effector towards the workpiece as a function of distance (thin-blue line) and boundary given by the approach constraint (thick-red line).

**Fig. 5.** Graphs for the approach constraint in the first experiment.



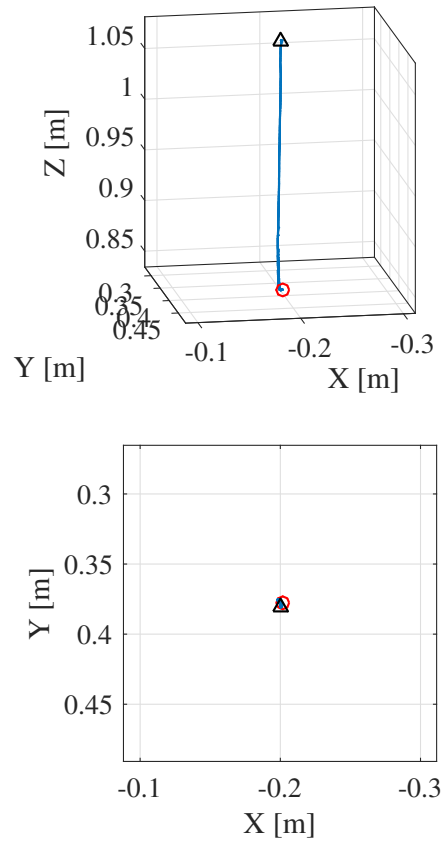
**Fig. 6.** Tool orientation angles in the first experiment. From top to bottom: roll, pitch and yaw angles. In the first two graphs: thin line, reference values supplied by the machine vision system; thick line, actual angle values.

operator is again guiding the tool away from the workpiece and the boundary  
 580 constraint becomes active, preventing the tool from leaving the allowed area,  
 which has been properly updated after the workpiece relocation due to the real-  
 time information supplied by the 3D camera network.

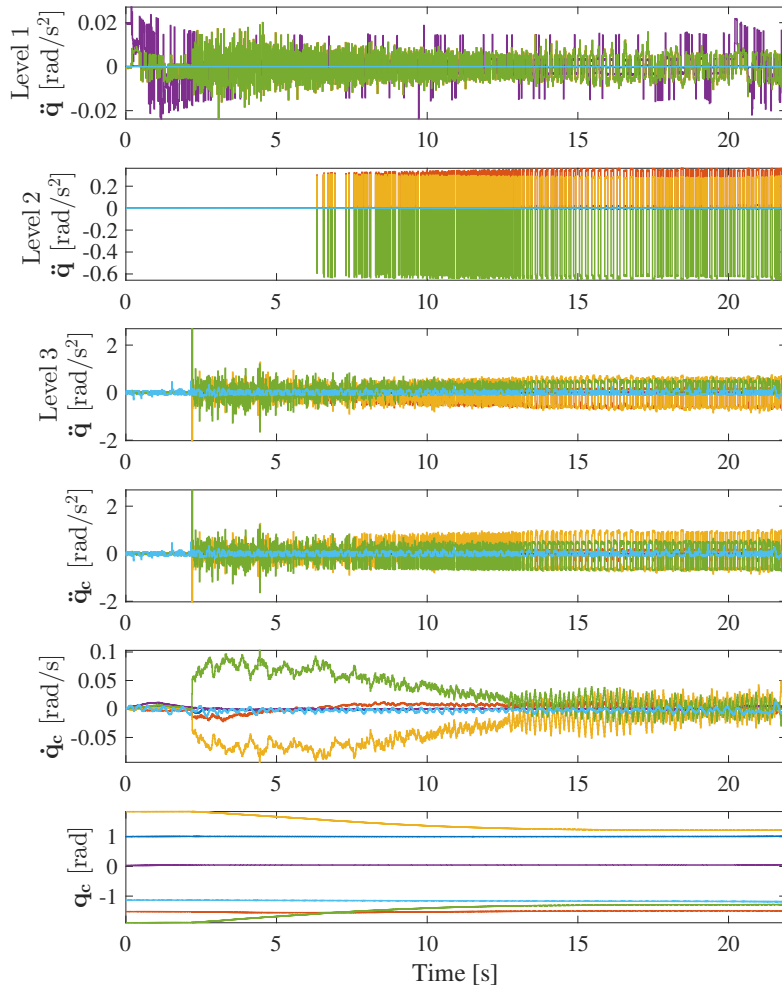
Next, several graphs are presented to show the quantitative performance  
 of the second experiment. Fig. 10 shows the functions and activation of the  
 585 boundary constraint. Note that the constraint is activated during six intervals  
 (see the bottom graph) but the tool position  $\bar{\mathbf{p}}_w$  (relative to the workpiece) is  
 properly confined within the allowed area since the value of  $\sigma_b$  is always less  
 than zero, see the top graph and (13).

The tool orientation angles for the second experiment are shown in Fig. 11.  
 590 As before, the first and second graphs show that the values of roll and yaw angles  
 are very similar to the reference values supplied by the machine vision system,  
 which is due to the orientation control implemented in Level 2. Moreover, the  
 yaw angle is mainly modified at the beginning of the experiment, which is due

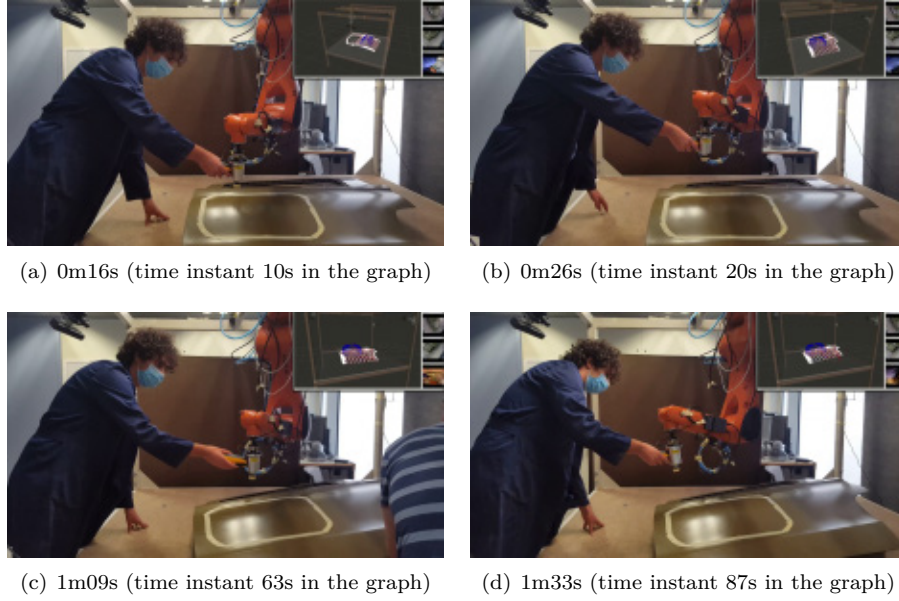




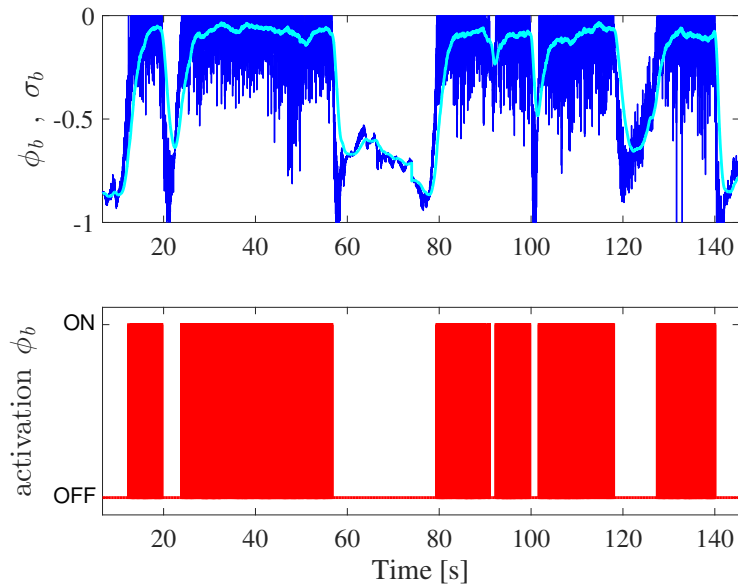
**Fig. 7.** Trajectory followed by the robot end-effector in the first experiment (triangle and circle symbols denote the initial and final positions, respectively).



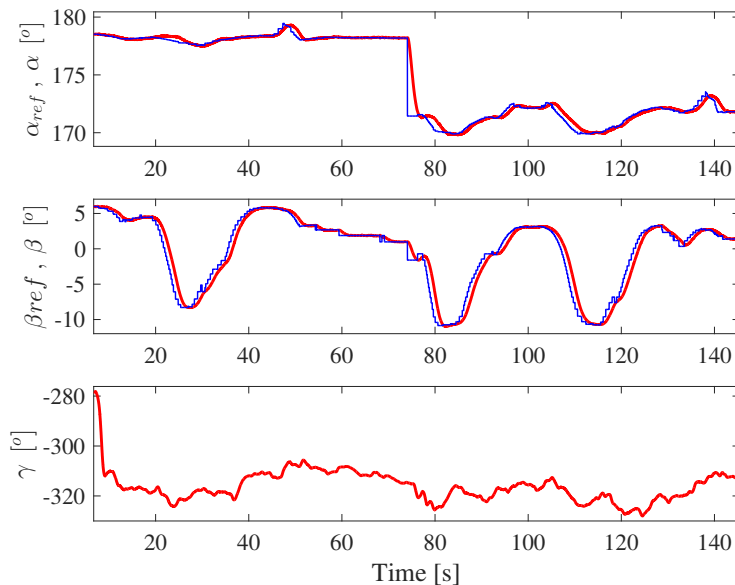
**Fig. 8.** Control signals in the first experiment. From top to bottom: commanded accelerations computed by each control level; joint accelerations, velocities and positions to be sent to the robot controller. In the graphs, a different color is used for each robot joint, i.e., from the first to the sixth joint: blue, brown, yellow, magenta, green and cyan.



**Fig. 9.** Frames of the video of the second experiment.



**Fig. 10.** Boundary constraint in the second experiment: top graph, constraint functions  $\phi_b$  (dark-blue) and  $\sigma_b$  (light-cyan); bottom graph, activation of the boundary constraint.

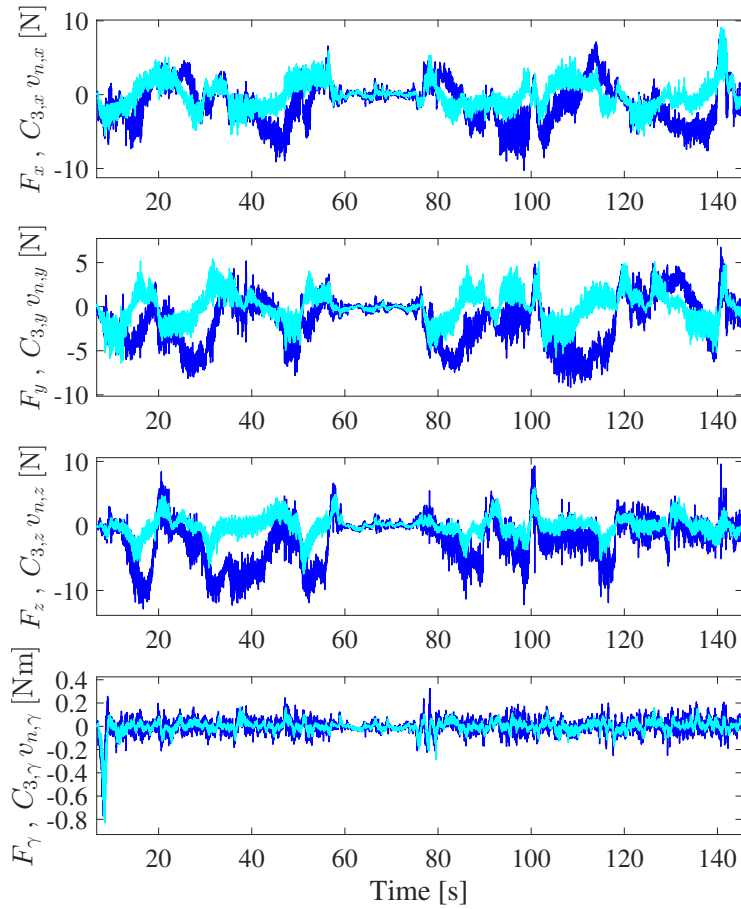


**Fig. 11.** Tool orientation angles in the second experiment:  $\alpha$ ,  $\beta$  and  $\gamma$ . In the first two graphs: thin line, reference values supplied by the machine vision system; thick line, actual angle values.

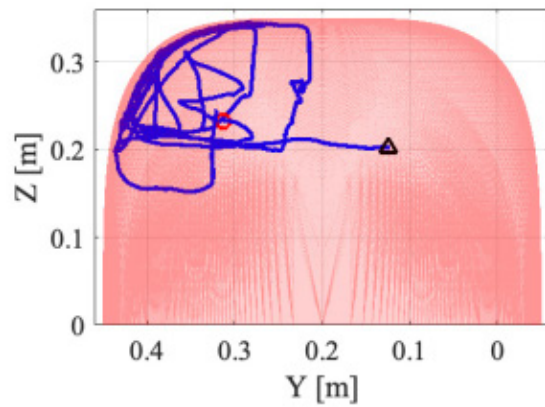
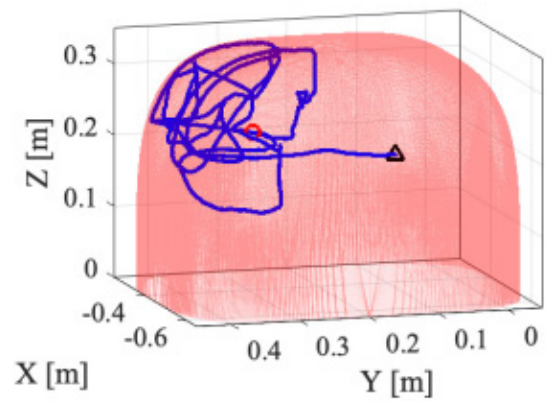
to the tool guidance of the manual operation. In this sense, Fig. 12 shows the  
 595 behavior of the tool guidance in Level 3, where it can be seen that the forces  
 exerted by the human operator are properly followed by the tool velocities except  
 at some intervals (see the second and third graphs) due to the activation of the  
 boundary constraint, as shown in Fig. 10. That is, the tool does not follow the  
 operator forces if that means leaving the allowed area.

600 The tool position in the second experiment is represented in Fig. 13. Note  
 that, despite the tool guidance performed by the user, the trajectory followed  
 by the robot tool is properly confined within the allowed area, whose boundary  
 is given by a superellipse, which looks like a rectangular prism with rounded  
 corners as shown in the figure.

605 The control signals in the second experiment are presented in Fig. 14, where  
 the commanded accelerations computed by each control level are shown. Note  
 that the commanded acceleration computed by Level 1 (see the top graph)  
 is non-zero when the boundary constraint is active (see the bottom graph in



**Fig. 12.** Tool guidance in the second experiment: tool velocities (multiplied by  $C_d$ ) in light-cyan and forces of the human operator in dark-blue. From top to bottom: linear  $X$ , linear  $Y$ , linear  $Z$  and angular  $Z$  components of the vectors (all four components are relative to the tool coordinate system).

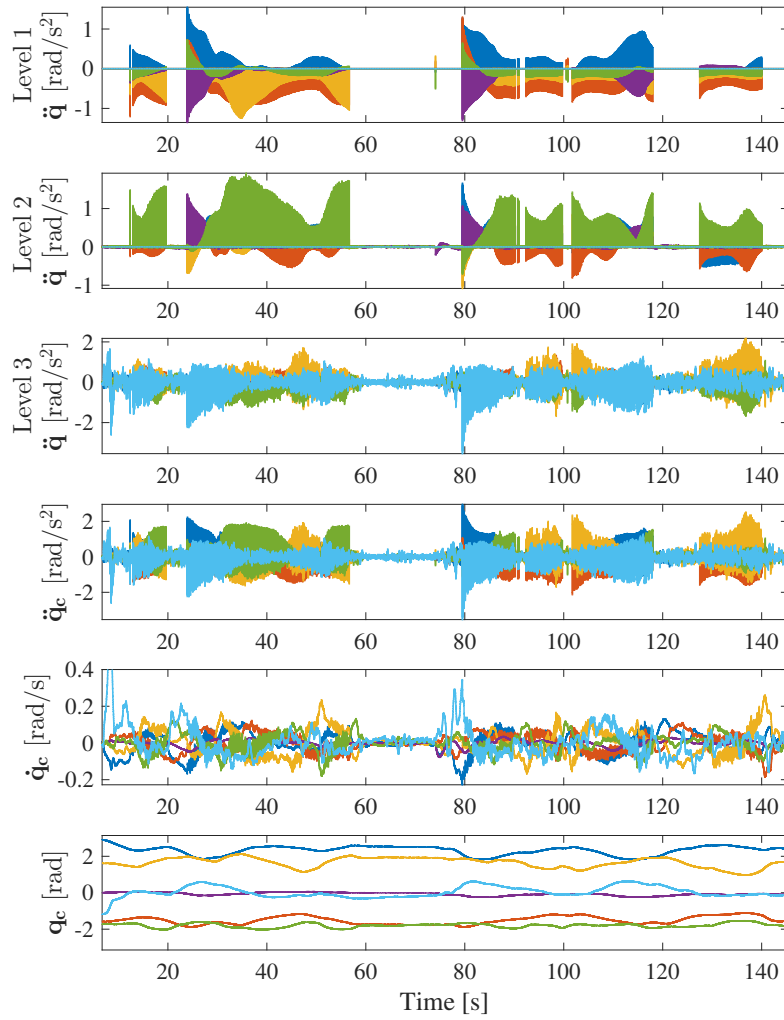


**Fig. 13.** Trajectory of the tool position in the second experiment (triangle and circle symbols denote the initial and final positions, respectively) and mesh representing the boundary of the allowed area.

Fig. 10). Moreover, the commanded acceleration computed by Level 2 (see the  
610 second graph) is also non-zero when the boundary constraint is active, which is  
due to the fact that the orientation control has to compensate the corrections  
introduced by Level 1 in order to keep the right tool orientation. Finally, note  
that the commanded acceleration computed by Level 3 (see the third graph) is  
approximately zero in the interval 60s-75s, which corresponds to the part of the  
615 experiment where the user is not guiding the tool, see Fig. 12

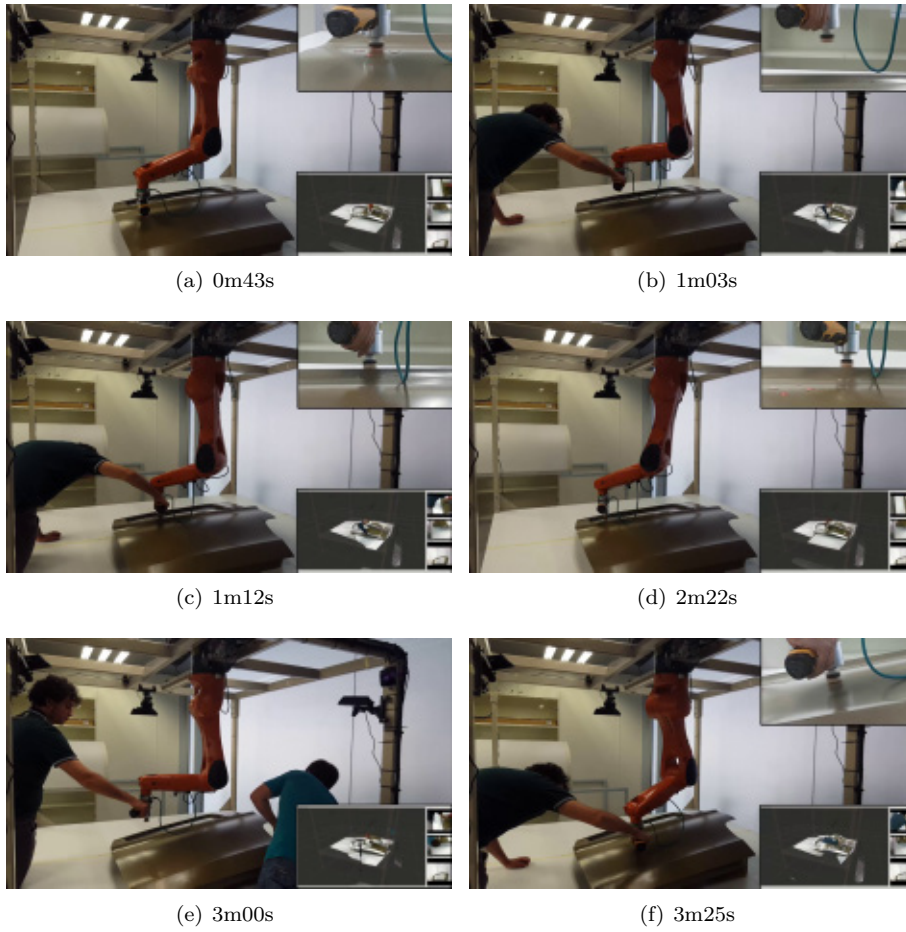
A third experiment was performed to show the utility of combining the two  
modes of operation, i.e., manual and automatic operation, see the video 42.  
Several frames of this video recording are shown in Fig. 15: at around 43s, see  
Fig. 15(a) the robotic arm is using the automatic operation to perform the  
620 sanding on a pre-established point of the workpiece surface (two pre-established  
points of the workpiece surface were cyclically used by the automatic operation);  
in the interval 1m03s-1m12s, see Fig. 15(b) and Fig. 15(c), the human operator  
uses the tool guidance to treat other regions of the workpiece surface; at around  
2m22s, see Fig. 15(d), the robotic arm is again using the automatic operation  
625 to perform the sanding on a pre-established point of the workpiece surface; at  
around 3m00s, see Fig. 15(e), the workpiece (i.e., the car door) is relocated; and  
at around 3m25s, see Fig. 15(f), the human operator is again guiding the tool  
to perform the sanding on an arbitrary region of the workpiece surface, which  
is done properly despite the previous workpiece relocation due to the real-time  
630 information supplied by the 3D camera network.

It is worth remarking that the proposed approach is able to dynamically  
adapt to arbitrary changes of the workpiece location due to the real-time in-  
formation supplied by the 3D camera network. This feature has been shown  
both in the second and third experiments. In particular, as explained above,  
635 in the second experiment the workpiece (i.e., the car door) is arbitrarily re-  
located at around 1m09s and, subsequently, the boundary constraint is properly  
activated to prevent the tool from leaving the allowed area, which is properly  
updated after the workpiece relocation. Furthermore, as mentioned above, in  
the third experiment the workpiece is arbitrarily relocated at around 3m00s



**Fig. 14.** Control signals in the second 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. 15.** Frames of the third experiment recording.

640 and, subsequently, the human operator guides the tool to perform the sanding on a specific region of the workpiece surface, which is done properly. That is, the tool approaching and perpendicularity are properly controlled by the robot system despite the previous workpiece relocation.

## 6. Conclusions

645 A human-robot cooperation method was developed in this work to conduct industrial operations such as finishing, sanding, deburring, grinding, etc. on the surface of a workpiece. Previous works dealing with the automation of these industrial operations use a completely automatic operation of the robot system or consider a low degree of human-robot interaction. In contrast, this work proposed a synergistic cooperation between the human operator and the robot system to get the best from both: the human operator provides *flexibility*, 650 guiding the tool of the robot system to treat arbitrary regions of the workpiece surface; while the robot system provides *strength, accuracy and security*, not only holding the tool and keeping the right tool orientation, but also guaranteeing a smooth approach to the workpiece and confining the tool within the 655 allowed area close to the workpiece.

In this way, the robot system provides a crucial aid to the human operator to perform the surface treatment, instead of replacing them, which allows to benefit from the virtues of both, as aforementioned.

660 Moreover, to add more flexibility to the proposed method, when the user is not guiding the robot tool, a robot automatic operation is activated to perform the treatment in prior established regions. Furthermore, a camera network was used to get a global view of the robot workspace in order to obtain the workpiece location accurately and in real-time, avoiding occlusions.

665 The effectiveness and feasibility of the proposed approach was shown with several experiments using a 6R robotic arm.

As further work, it is proposed to include the possibility for the user to perform the guidance of the robot tool remotely using an advanced teleopera-

tion system. For this purpose, several technologically advanced devices could  
670 be used, such as haptic systems, virtual reality headsets, interactive screens,  
etc. This feature would improve the proposed application in terms of usability,  
comfort and security.

## References

- [1] P. Stavropoulos, A. Papacharalampopoulos, D. Petridis, A vision-based  
675 system for real-time defect detection: a rubber compound part case study,  
Procedia CIRP 93 (2020) 1230–1235.
- [2] K. K. Kieselbach, M. Nöthen, H. Heuer, 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,  
680 Journal of Coatings Technology and Research 16 (2019) 1033–1042.
- [3] Active contact flange (ACF) tool from FerRobotics, <https://www.ferrobotics.com/en/technologie-produkte/produkte/acf/> (Accessed  
2020/06/10).
- [4] Y. Kuo, S. Huang, C. Lan, Sensorless force control of automated grind-  
685 ing/deburring using an adjustable force regulation mechanism, in: 2019 Int.  
Conference on Robotics and Automation (ICRA), 2019, pp. 9489–9495.
- [5] A. E. K. Mohammad, J. Hong, D. Wang, Design of a force-controlled end-  
effector with low-inertia effect for robotic polishing using macro-mini robot  
approach, Robotics and Computer-Integrated Manuf. 49 (2018) 54 – 65.
- 690 [6] E. Villagrossi, C. Cenati, N. Pedrocchi, M. Beschi, L. Molinari Tosatti,  
Flexible robot-based cast iron deburring cell for small batch production  
using single-point laser sensor, The International Journal of Advanced  
Manufacturing Technology 92 (2017) 1425–1438.
- [7] S. Diao, . Chen, X, J. Luo, Development and experimental evaluation of a  
695 3D vision system for grinding robot, Sensors 18 (2018) 1–20.

- [8] I. Mohsin, K. He, J. Cai, H. Chen, R. Du, Robotic polishing with force controlled end effector and multi-step path planning, in: 2017 IEEE Int. Conference on Information and Automation (ICIA), 2017, pp. 344–348.
- [9] I. Mohsin, K. He, Z. Li, R. Du, Path planning under force control in robotic polishing of the complex curved surfaces, Applied Sciences 9 (2019) 5489.
- [10] Y. Oba, Y. Kakinuma, Simultaneous tool posture and polishing force control of unknown curved surface using serial-parallel mechanism polishing machine, Precision Engineering 49 (2017) 24–32.
- [11] F. Tian, Z. Li, C. Lv, G. Liu, Polishing pressure investigations of robot automatic polishing on curved surfaces, The International Journal of Advanced Manufacturing Technology 87 (2016) 639–646.
- [12] A. Sato, K. Shen, M. Minami, T. Matsuno, Improvement of force-sensorless grinding accuracy with resistance compensation, Artificial Life and Robotics 22 (2017) 509–514.
- [13] T. Segreto, R. Teti, Machine learning for in-process end-point detection in robot-assisted polishing using multiple sensor monitoring, Int. J. Adv. Manuf. Technol. 103 (2019) 4173–4187.
- [14] S. Huang, Z. Zhu, J. Chen, X. Zhou, J. Yu, P. Gao, H. Wang, Target force tracking and automatic contour surface processing in grinding of industrial robots, in: 2020 6th International Conference on Control, Automation and Robotics (ICCAR), 2020, pp. 188–195.
- [15] A. Kharidege, D. Ting, Z. Yajun, A practical approach for automated polishing system of free-form surface path generation based on industrial arm robot, International Journal of Advanced Manufacturing Technology 93 (2017) 3921 – 3934.
- [16] H. Ochoa, R. Cortesão, Control architecture for robotic-assisted polishing tasks based on human skills, in: IECON 2019 - 45th Annual Conference of the IEEE Industrial Electronics Society, volume 1, 2019, pp. 630–637.

- [17] B. Nemeč, K. Yasuda, N. Mullennix, N. Likar, A. Ude, Learning by demonstration and adaptation of finishing operations using virtual mechanism approach, in: 2018 IEEE International Conference on Robotics and Automation (ICRA), 2018, pp. 7219–7225.
- [18] C. Gaz, E. Magrini, A. De Luca, A model-based residual approach for human-robot collaboration during manual polishing operations, *Mechatronics* 55 (2018) 234 – 247.
- [19] C. Perez-Vidal, L. Gracia, S. Sanchez-Caballero, J. E. Solanes, A. Saccon, J. Tornero, Design of a polishing tool for collaborative robotics using minimum viable product approach, *International Journal of Computer Integrated Manufacturing* 32 (2019) 848–857.
- [20] L. Gracia, J. Solanes, P. Muñoz-Benavent, J. Valls Miro, C. Perez-Vidal, J. Tornero, Human-robot collaboration for surface treatment tasks, *Interaction Studies* 20 (2019) 148–184.
- [21] L. Santos, R. Cortesão, Computed-torque control for robotic-assisted tele-echography based on perceived stiffness estimation, *IEEE Transactions on Automation Science and Engineering* 15 (2018) 1337–1354.
- [22] L. Cui, H. Wang, X. Liang, J. Wang, W. Chen, Visual servoing of a flexible aerial refueling boom with an eye-in-hand camera, *IEEE Transactions on Systems, Man, and Cybernetics: Systems* (2020) In press (doi: 10.1109/TSMC.2019.2957992).
- [23] Taryudi, M.-S. Wang, Eye to hand calibration using anfis for stereo vision-based object manipulation system, *Microsystem Technologies* 24 (2018) 305–317.
- [24] S.-H. Lai, C.-W. Fu, S. Chang, A generalized depth estimation algorithm with a single image, *IEEE transactions on pattern analysis and machine intelligence* 14 (1992) 405–411.

- [25] D. Kent, C. Saldanha, S. Chernova, Leveraging depth data in remote robot teleoperation interfaces for general object manipulation, *The International Journal of Robotics Research* 39 (2020) 39–53.
- [26] Y. Gong, E. J. Seibel, Feature-based three-dimensional registration for repetitive geometry in machine vision, *Journal of information technology & software engineering* 6 (2016) 184–1–184–5.
- [27] Y. He, B. Liang, J. Yang, S. Li, J. He, An iterative closest points algorithm for registration of 3D laser scanner point clouds with geometric features, *Sensors* 17 (2017) 1862–1–1862–16.
- [28] J. Angeles, *Fundamentals of Robotic Mechanical Systems: Theory, Methods, and Algorithms*, 3rd ed., Springer-Verlag, New York, NJ, 2007.
- [29] W. Lee, Y. Lee, G. Park, S. Hong, Y. Kang, A whole-body rescue motion control with task-priority strategy for a rescue robot, *Autonomous Robots* 41 (2017) 243–258.
- [30] F. Flacco, A. De Luca, O. Khatib, Prioritized multi-task motion control of redundant robots under hard joint constraints, in: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 3970–3977.
- [31] G. Golub, W. Kahan, Calculating the singular values and pseudo-inverse of a matrix, *Journal of the Society for Industrial and Applied Mathematics, Series B: Numerical Analysis* 2 (1965) 205–224.
- [32] Y. Nakamura, H. Hanafusa, T. Yoshikawa, Task-priority based redundancy control of robot manipulators, *Int. J. of Robotics Research* 6 (1987) 3–15.
- [33] V. Utkin, J. Guldner, J. Shi, *Sliding Mode Control in Electro-Mechanical Systems*, 2nd ed., Taylor & Francis, London, 2009.
- [34] J. Baek, M. Jin, S. Han, A new adaptive sliding-mode control scheme for application to robot manipulators, *IEEE Transactions on Industrial Electronics* 63 (2016) 3628–3637.

- [35] J. Lee, P. H. Chang, M. Jin, Adaptive integral sliding mode control with time-delay estimation for robot manipulators, *IEEE Transactions on Industrial Electronics* 64 (2017) 6796–6804.
- [36] B. Siciliano, O. E. Khatib, *Handbook of robotics*, Springer-Verlag, London, UK, 2008.
- [37] B. Yao, Z. Zhou, L. Wang, W. Xu, Q. Liu, A. Liu, Sensorless and adaptive admittance control of industrial robot in physical human-robot interaction, *Robotics and Computer-Integrated Manufacturing* 51 (2018) 158 – 168.
- [38] Z. Li, B. Huang, Z. Ye, M. Deng, C. Yang, Physical human-robot interaction of a robotic exoskeleton by admittance control, *IEEE Transactions on Industrial Electronics* 65 (2018) 9614–9624.
- [39] P. Corke, *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*, 2nd ed., Springer-Verlag, Berlin, Germany, 2017.
- [40] Video of the first experiment, <https://media.upv.es/player/?id=7449ae40-fc15-11ea-9ede-d1ad8f82e7cd>, Accessed 10/30/2020.
- [41] Video of the second experiment, <https://media.upv.es/player/?id=ae62de20-fc11-11ea-9ede-d1ad8f82e7cd>, Accessed 10/30/2020.
- [42] Video of the third experiment, <https://media.upv.es/player/?id=75ac5d70-fc13-11ea-9ede-d1ad8f82e7cd>, Accessed 10/30/2020.