

“© 2018 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.”

A Sliding Mode Control Architecture for Human-Manipulator Cooperative Surface Treatment Tasks

Luis Gracia¹, J. Ernesto Solanes^{1*}, Pau Muñoz-Benavent¹,
Jaime Valls Miro², Carlos Perez-Vidal³ and Josep Tornero¹

Abstract—This paper presents a control architecture readily suitable for surface treatment tasks such as polishing, grinding, finishing or deburring as carried out by a human operator, with the added benefit of accuracy, recurrence and physical strength as administered by a robotic manipulator partner. The shared strategy effectively couples the human operator proprioceptive abilities and fine skills through his interactions with the autonomous physical agent. The novel proposed control scheme is based on task prioritization and a non-conventional sliding mode control, which is considered to benefit from its inherent robustness and low computational cost. The system relies on two force sensors, one located between the last link of the robot and the surface treatment tool, and the other located in some place of the robot end-effector: the former is used to suitably accomplish the conditioning task, while the latter is used by the operator to manually guide the robotic tool. When the operator chooses to cease guiding the tool, the robot motion safely switches back to an automatic reference tracking. The paper presents the theories for the novel collaborative controller, whilst its effectiveness for robotic surface treatment is substantiated by experimental results using a redundant 7R manipulator and a mock-up conditioning tool.

I. INTRODUCTION

The automation of industrial processes has generated great improvements in terms of product quality, cost reduction and operator safety and comfort. However the complexity of many industrial processes still dictate the need to continue with manual handling in many cases. Despite the advanced dexterity that robotic manipulators can contribute in some of these operations, it is widely acknowledged they still cannot compete with the adaptability of humans and as such there is currently a strong tendency to combine robots and humans to collaboratively accomplish elaborate tasks.

One of the least automated processes is surface treatment operations [1], which continues to be mainly a manual process being carried out by skilled workers, giving rise to adverse issues such as subjectivity in the evaluation criteria, human errors and the like [2]. In surface conditioning operations the tool has to remain in contact with the product surface whilst applying the specific treatment (polishing, grinding, etc.). To that end, the forces exerted by the tool

This work was supported in part by the Spanish Government under the project DPI2017-87656-C2-1-R and the Generalitat Valenciana under Grant VALi+d APOSTD/2016/044.

¹Instituto de Diseño y Fabricación, Universitat Politècnica de València, Camino de vera s/n, 46022, Valencia, Spain

²Centre for Autonomous Systems, University of Technology Sydney, NSW 2007 Sydney, Australia

³Departamento de Ingeniería de Sistemas y Automática, Universidad Miguel Hernández, Avda de la Universidad s/n, 03202 Elche, Spain

*Corresponding author: esolanes@idf.upv.es

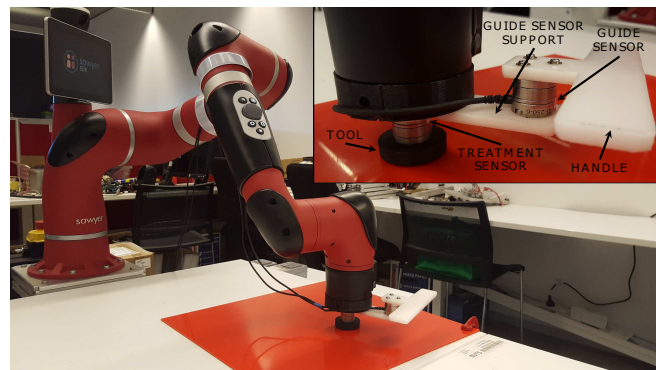


Fig. 1. Experimental setup: 7R “Sawyer” serial cobot manipulator with *treatment* and *guide* force sensors (silver), a rectangular plastic rigid object (white) supporting the *guide* sensor, a guiding element consisting of a T-shaped plastic rigid object (white) to be handled by the human operator (see examples in Figs 2 and 6), a tool consisting of a cylinder (black) and a flat rectangular surface target (red).

have to be adequately controlled at all times, maintaining constant pressure whilst ensuring perpendicularity to the surface to homogenize the pressure on all contact points [3]. This is not a trivial attainment when human operators have to deal with complex shapes. This difficulty is compounded when the product to be treated is part of a production line where it might be in motion, or its morphology constantly changing, making it even more challenging for both, humans and robots alike, to carry out treatment operations efficiently.

To mitigate the drawbacks of both, manual and robotic automatic surface treatment, this work proposes an original control framework for human-robot collaboration. The proposal adopts the form of a human operator performing the task of finely “guiding” the tool along the surface of an object, whilst operating in synergy with a robotic manipulator in charge of automatically maintaining both the tool’s pressure on the surface and the tool’s perpendicularity to the surface, hence ensuring a flexible surface treatment. For this purpose, this work proposes a multi-task, sliding mode control (SMC) architecture, and resorts to an arrangement with two force sensors to accomplish the collaborative operation, namely a *treatment* and a *guide* sensor, depicted in Fig. 1. Note that the force sensor used to accomplish the surface treatment cannot be used simultaneously to guide the robot since the forces exerted by the human operator and those for the surface conditioning would not be distinguished, which would prevent the satisfactory accomplishment of the surface treatment task. This would be the case for instance

should the operator exert a force in the direction away from the surface to be treated larger than the desired pressure for the treatment task. In this troubled scenario the robot would end up moving away from the object to be treated and contact would be lost.

While solutions of robots working collaboratively ‘‘alongside’’ humans in polishing tasks exist (e.g., see <https://www.youtube.com/watch?v=kOPVvYapE1Q>, where a robot is used for loudspeakers manufacturing), this work proposes a truly collaborative human-robot solution to cooperatively perform surface treatment operations. Next, a literature review is presented about the main aspects concerning this work: automatic surface treatment, robot technique and robot force control using sliding mode (SM) techniques.

Many approaches can be found in the literature tackling the problem of *automatic surface treatment* using robot manipulators with force feedback, e.g., see [3] among others. The *motion guidance* for robot manipulators is typically obtained via a wrist-mounted force sensor which evaluates the forces exerted by the human operator. The sensor measurements are typically converted into changes of the robot position through compliance control [4], yet other variants can be found in the literature [5]. Other approaches tackling the problem of *robot force control* and human-robot collaborative applications are based on SMC theory, see [6], [7] among others. In this sense, SMC is used in this work to benefit from its inherent robustness and low computational cost [8], e.g., only the first-order kinematics of the robot is required for the proposed approach, as detailed below.

The paper is organized as follows: next section introduces the required background theory and mathematical proofs of the proposed SMC collaborative controller. Subsequently, Section III presents the proposed method for robotic surface treatment, while its actual implementation is detailed in Section IV for other researchers to be able to replicate the proposed scheme. The feasibility of the method is substantiated by experimental results in Section V using a redundant 7R manipulator. Finally, conclusions are drawn in Section VI.

II. BACKGROUND THEORY

A. Kinematics

The robot *pose* \mathbf{p} and *configuration* \mathbf{q} , and their derivatives, are related by the following equations:

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

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

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

where \mathbf{l} and \mathbf{J} are the kinematic function and Jacobian matrix of the robot, respectively.

B. Robot control

This work assumes the existence of a robot controller in charge of achieving a particular joint acceleration from the

commanded or reference vector $\ddot{\mathbf{q}}_c$, and that its dynamics is fast enough compared to that of $\ddot{\mathbf{q}}_c$. Hence, the relationship:

$$\ddot{\mathbf{q}} = \ddot{\mathbf{q}}_c + \mathbf{d}_c \quad (4)$$

holds approximately true, where \mathbf{d}_c represents inaccuracies due to disturbances, which is assumed to be bounded and smooth.

C. Task-priority scheme

The task-priority strategy [9] allows to tackle several objectives simultaneously assigning an order of priority to each one. Let us consider M tasks which consist in calculating the reference joint acceleration vector $\ddot{\mathbf{q}}_c$ to fulfill the following equality constraints:

$$\mathbf{A}_i \ddot{\mathbf{q}}_c = \mathbf{b}_i, \quad i = 1, \dots, M, \quad (5)$$

where matrix \mathbf{A}_i and vector \mathbf{b}_i of the i th task are assumed known and index i represents the priority order ($i = 1$ for highest priority). The solution $\ddot{\mathbf{q}}_{c,M}$ that hierarchically minimizes the error of equations in (5) is given by [10]:

$$\ddot{\mathbf{q}}_{c,i} = \ddot{\mathbf{q}}_{c,i-1} + (\mathbf{A}_i \mathbf{N}_{i-1})^\dagger (\mathbf{b}_i - \mathbf{A}_i \ddot{\mathbf{q}}_{c,i-1}) \quad (6)$$

$$\mathbf{N}_i = \mathbf{N}_{i-1} (\mathbf{I} - (\mathbf{A}_i \mathbf{N}_{i-1})^\dagger (\mathbf{A}_i \mathbf{N}_{i-1})), \quad (7)$$

$$\text{with } i = 1, \dots, M, \quad \ddot{\mathbf{q}}_{c,0} = \mathbf{0} \text{ and } \mathbf{N}_0 = \mathbf{I},$$

where \mathbf{I} and $\mathbf{0}$ denote the identity matrix and zero column vector, respectively, superscript \dagger denotes the Moore-Penrose pseudoinverse and $\ddot{\mathbf{q}}_{c,i}$ and \mathbf{N}_i are the solution vector and null-space projection matrix for the set of first i tasks.

D. SMC to satisfy equality constraints

Theorem II.1. Consider the following dynamical system:

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

where $\mathbf{x}(t)$ is the state vector, $\mathbf{d}(t)$ an unmeasured disturbance, $\mathbf{u}(t)$ the control input vector, \mathbf{f} a vector field and \mathbf{g} a set of vector fields.

Consider also that the state vector \mathbf{x} is subject to equality constraints $\phi_{eq,i}(\mathbf{x}) = 0$, $i = 1, \dots, N_{eq}$, where $\phi_{eq,i}(\mathbf{x})$ is the i th equality constraint function. Thus, the region Φ_{eq} compatible with the constraints is given by:

$$\Phi_{eq} = \{ \mathbf{x} \mid \phi_{eq,i}(\mathbf{x}) = 0 \}, \quad i = 1, \dots, N_{eq}. \quad (9)$$

Then, assuming that the constraint functions $\phi_{eq,i}$ are differentiable, the control action \mathbf{u} that fulfills the variable structure control below guarantees that the system converges to Φ_{eq} in finite time and remains there henceforth:

$$\mathbf{L}_g \phi_{eq} \mathbf{u} = -\mathbf{W}_{eq} \text{sign}(\phi_{eq}) u_{eq}^+ \quad (10)$$

$$u_{eq}^+ > \|L_f \phi_{eq}\|_1 / \text{diag}_{\min}(\mathbf{W}_{eq}), \quad (11)$$

where ϕ_{eq} is a column vector with all the constraint functions $\phi_{eq,i}$, the scalar $L_f \phi_{eq,i} = \frac{\partial \phi_{eq,i}}{\partial \mathbf{x}} \mathbf{f}$ and the row vector $\mathbf{L}_g \phi_{eq,i} = \frac{\partial \phi_{eq,i}}{\partial \mathbf{x}} \mathbf{g}$ denote the Lie derivatives of $\phi_{eq,i}(\mathbf{x})$ in the direction of vector field \mathbf{f} and in the direction of the set of vector fields \mathbf{g} , respectively, column vector $L_f \phi_{eq}$ contains

the elements $L_f \phi_{eq,i}$ of all equality constraints, matrix $\mathbf{L}_g \phi_{eq}$ contains the row vectors $\mathbf{L}_g \phi_{eq,i}$ of all equality constraints, $\text{sign}(\cdot)$ represents the sign function, positive scalar u_{eq}^+ is the switching gain, \mathbf{W}_{eq} is a diagonal matrix of switching gain weights for the constraints, $\|\cdot\|_1$ represents the 1-norm norm and function $\text{diag}_{\min}(\cdot)$ computes the minimum value of the diagonal elements of a matrix.

Proof. The proof is straightforward from the Proof 2.1 in [11] and its generalization. Details omitted for brevity. \square

E. SMC to satisfy inequality constraints

Theorem II.2. *Considering the system (8) and that the state vector \mathbf{x} is subject to inequality constraints $\phi_{in,i}(\mathbf{x}) \leq 0$, $i = 1, \dots, N_{in}$, where $\phi_{in,i}(\mathbf{x})$ is the i th constraint function, the region Φ_{in} compatible with the constraints is given by:*

$$\Phi_{in} = \{\mathbf{x} \mid \phi_{in,i}(\mathbf{x}) \leq 0\}, \quad i = 1, \dots, N_{in}. \quad (12)$$

Then, assuming that the constraint functions $\phi_{in,i}$ are differentiable, the control action \mathbf{u} that fulfills the variable structure control below guarantees that the system converges to Φ_{in} in finite time and remains there henceforth:

$$v2dm(\text{pos}(\phi_{in})) \mathbf{L}_g \phi_{in} \mathbf{u} = -\mathbf{W}_{in} \text{pos}(\phi_{in}) u_{in}^+ \quad (13)$$

$$u_{in}^+ > \sum_{i=1}^{n_a} (\max(L_f \phi_{in,i}, 0)) / \text{diag}_{\min}(\mathbf{W}_{in}), \quad (14)$$

where function $v2dm(\cdot)$ converts a vector into a diagonal matrix, function $\text{pos}(\cdot)$ represents the positive function (i.e., $\text{pos}(x) = 0$ if $x < 0$ and $\text{pos}(x) = 1$ otherwise), ϕ_{in} is a column vector with all the inequality constraint functions $\phi_{in,i}$, matrix $\mathbf{L}_g \phi_{in}$ contains the row vectors $\mathbf{L}_g \phi_{in,i}$ of all inequality constraints, the scalar $L_f \phi_{in,i} = \frac{\partial \phi_{in,i}}{\partial \mathbf{x}} \mathbf{f}$ and the row vector $\mathbf{L}_g \phi_{in,i} = \frac{\partial \phi_{in,i}}{\partial \mathbf{x}} \mathbf{g}$ denote the Lie derivatives of the constraints in the direction of vector field \mathbf{f} and in the direction of the set of vector fields \mathbf{g} , respectively, positive scalar u_{in}^+ is the switching gain, \mathbf{W}_{in} is a diagonal matrix representing the switching gain weights and n_a is the number of active inequality constraints, i.e., those with $\phi_{in,i} \geq 0$.

Proof. The proof is similar to that of Theorem II.1 but replacing the commutation function $\text{sign}(\cdot)$ with $\text{pos}(\cdot)$. Details omitted for brevity. \square

F. Modified constraints

The actual constraints $\sigma_{eq,i}$ and $\sigma_{in,i}$ will be modified to include the speed of movement as follows:

$$\phi_{eq,i} = \sigma_{eq,i} + K_{eq,i} \dot{\sigma}_{eq,i} = 0 \quad (15)$$

$$\phi_{in,i} = \sigma_{in,i} + K_{in,i} \dot{\sigma}_{in,i} \leq 0, \quad (16)$$

where $K_{eq,i}$ and $K_{in,i}$ are design parameters that determine the rate of approach to the original constraint.

G. Chattering

The upper bound for the amplitude $\Delta\phi$ of the chattering [8] can be obtained from (10) as:

$$\Delta\phi = T_s |\mathbf{L}_g \phi \mathbf{u}| = T_s u^+ \text{diag}(\mathbf{W}), \quad (17)$$

where T_s is the sampling time of the robot system and function $\text{diag}(\cdot)$ gives a column vector with the diagonal elements of a square matrix.

III. METHOD

A. Prioritized tasks

- The first level (highest-priority task) includes the *equality constraints* required to properly accomplish the surface treatment with the robot tool.
- The second level includes the *inequality constraints* required for the robot to track the operator's movements.
- The third level is used for *reference tracking* in order to apply the treatment with the tool on a specific area of the surface: deviations from the reference trajectory are allowed if they are required to follow the operator's movements or to satisfy the above equality constraints.
- The fourth level (lowest-priority task) is used to keep the robot close to the *home* configuration.

B. Lie derivatives

A dynamical system in the form of Eq. (8) is considered with the state $\mathbf{x} = [\mathbf{q}^T \quad \dot{\mathbf{q}}^T]^T$, the disturbance $\mathbf{d} = \mathbf{d}_c$ and the input $\mathbf{u} = \ddot{\mathbf{q}}_c$. Hence, the model is a double integrator, and from (4) the state equation results in:

$$\dot{\mathbf{x}} = \begin{bmatrix} \mathbf{O} & \mathbf{I} \\ \mathbf{O} & \mathbf{O} \end{bmatrix} \mathbf{x} + \begin{bmatrix} \mathbf{0} \\ \mathbf{d}_c \end{bmatrix} + \begin{bmatrix} \mathbf{O} \\ \mathbf{I} \end{bmatrix} \mathbf{u}, \quad (18)$$

and, therefore, the Lie derivatives of ϕ_i are given by:

$$\mathbf{L}_g \phi_i = \nabla \phi_i^T \mathbf{g} = (\partial \phi_i / \partial \dot{\mathbf{q}})^T \quad (19)$$

$$L_f \phi_i = \nabla \phi_i^T \mathbf{f} = (\partial \phi_i / \partial \mathbf{q})^T \dot{\mathbf{q}} + (\partial \phi_i / \partial \mathbf{q})^T \mathbf{d}_c. \quad (20)$$

C. Force model

The constraints for the first two levels are defined below depending on the vectors \mathbf{F}_t and \mathbf{F}_g of forces and torques measured by the treatment and guide sensors, respectively. In many applications, the interaction forces \mathbf{F} between the robot/tool and the environment can be approximated by the ideal elastic model below [12]:

$$\mathbf{F}(\mathbf{q}, t) = \mathbf{K}_s \Delta \mathbf{s}(\mathbf{q}, \mathbf{p}_s) = [F_x \quad F_y \quad F_z \quad F_\alpha \quad F_\beta \quad F_\gamma]^T, \quad (21)$$

where \mathbf{F} is the force vector relative to the tool coordinate system, \mathbf{K}_s is a diagonal matrix with the stiffness coefficients for each tool axis and vector $\Delta \mathbf{s}$ is the mechanical deformation of the sensor relative to the tool coordinate system, which depends on the robot configuration \mathbf{q} and the position and orientation \mathbf{p}_s of the object in contact with the robot, i.e., the object being treated or the human operator guiding the robot tool.

D. Level 1: Constraints for the surface treatment task

Three equality constraints are defined as follows:

$$\sigma_{1,z}(\mathbf{F}_t) = \sigma_{1,z}(\mathbf{q}, t) = F_{t,z} - F_{z,ref} = 0 \quad (22)$$

$$\sigma_{1,\alpha}(\mathbf{F}_t) = \sigma_{1,\alpha}(\mathbf{q}, t) = F_{t,\alpha} = 0 \quad (23)$$

$$\sigma_{1,\beta}(\mathbf{F}_t) = \sigma_{1,\beta}(\mathbf{q}, t) = F_{t,\beta} = 0, \quad (24)$$

where $F_{t,z}$ is the linear force measured by the treatment sensor in the tool Z-axis, $F_{t,\alpha}$ and $F_{t,\beta}$ are the torques measured by the treatment sensor in the tool X- and Y-axes, and $F_{z,ref}$ is the desired force between the tool and the surface being treated in the tool Z-axis. Hence, the first constraint is used to attain the desired force $F_{z,ref}$ between the tool and the surface, whereas the last two constraints are used to keep the tool orientation perpendicular to the surface.

Taking into account (15), (19) and (21)–(24), the Lie derivative $\mathbf{L}_g\phi_1$ required for the SMC in (10) is given by:

$$\begin{aligned} \mathbf{L}_g\phi_1 &= \left(\frac{\partial\phi_1}{\partial\dot{\mathbf{q}}}\right)^T = \mathbf{K}_1 \left(\frac{\partial\sigma_1}{\partial\mathbf{q}}\right)^T \\ &= \mathbf{K}_1 \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \mathbf{K}_{t,s}\mathbf{J} = \mathbf{K}_1\mathbf{H}_1\mathbf{K}_{t,s}\mathbf{J}_n, \end{aligned} \quad (25)$$

where σ_1 is a column vector composed of all equality constraints $\sigma_{1,i}$, \mathbf{K}_1 is a diagonal matrix composed of all the approaching parameters $K_{1,i}$, see (15), $\mathbf{K}_{t,s}$ is the stiffness diagonal matrix for the treatment sensor and \mathbf{J}_n is the geometric Jacobian relative to the tool coordinate system [12], i.e., the Jacobian matrix relating the joint velocities $\dot{\mathbf{q}}$ and the linear and angular velocities of the end-effector relative to the tool coordinate system.

Since the stiffness coefficients $\mathbf{K}_{t,s}$ in $\mathbf{L}_g\phi_1$ (25) may not be known, they can be included without loss of generality in the switching gain weight matrix \mathbf{W}_1 , so that the conventional SMC given by (10) is modified as follows:

$$\mathbf{K}_1\mathbf{H}_1\mathbf{J}_n\dot{\mathbf{q}}_c = -\bar{\mathbf{W}}_1\text{sign}(\phi_1)u_1^+ \rightarrow \mathbf{A}_1\dot{\mathbf{q}}_c = \mathbf{b}_1, \quad (26)$$

where \mathbf{A}_1 and \mathbf{b}_1 are the values for the first task in (5) and:

$$\begin{aligned} \bar{\mathbf{W}}_1 &= \begin{bmatrix} W_{1,z}/K_{t,s,z} & 0 & 0 \\ 0 & W_{1,\alpha}/K_{t,s,\alpha} & 0 \\ 0 & 0 & W_{1,\beta}/K_{t,s,\beta} \end{bmatrix} \\ &= \begin{bmatrix} \bar{W}_{1,z} & 0 & 0 \\ 0 & \bar{W}_{1,\alpha} & 0 \\ 0 & 0 & \bar{W}_{1,\beta} \end{bmatrix}. \end{aligned} \quad (27)$$

E. Level 2: Constraints to track human operator's movement

The following inequality constraint is considered:

$$\sigma_2(\mathbf{F}_g) = \sigma_2(\mathbf{q}, t) = \sqrt{F_{g,x}^2 + F_{g,y}^2} - F_{l,0} = F_l - F_{l,0} \leq 0, \quad (28)$$

where $F_{g,x}$ and $F_{g,y}$ are the linear forces detected by the guide sensor in its X- and Y-axes, which are perpendicular to the robot end-effector, F_l is the magnitude of these linear forces and $F_{l,0}$ is a threshold so that the constraint becomes active when the magnitude F_l is larger than this threshold, in which case the robot tool is moved by the proposed SMC in the direction of the detected forces to fulfill the constraint.

Taking into account (16), (19) and (28), the Lie derivative $\mathbf{L}_g\phi_2$ for the above constraint is given by:

$$\begin{aligned} \mathbf{L}_g\phi_2 &= K_2 \begin{bmatrix} \bar{F}_{g,x} & \bar{F}_{g,y} & 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{K}_{g,s}\mathbf{J}_n \\ &= K_2\mathbf{H}_2\mathbf{K}_{g,s}\mathbf{J}_n, \end{aligned} \quad (29)$$

where K_2 is the approaching parameter to the original constraint (28), see (16), $\bar{F}_{g,i} = F_{g,i}/F_l$ represents the normalized linear force in the i axis and $\mathbf{K}_{g,s}$ is the stiffness diagonal matrix for the guide sensor.

The acceleration equality for the second level results in:

$$v2dm(\text{pos}(\phi_2))\mathbf{L}_g\phi_2\ddot{\mathbf{q}}_c = -W_2\text{pos}(\phi_2)u_2^+, \quad (30)$$

where W_2 and u_2^+ are the switching gain weight and switching gain, respectively, for the second level.

As before, the stiffness coefficients can be included in the switching gain weights so that the SMC given by (30) is modified as follows:

$$\begin{aligned} v2dm(\text{pos}(\phi_2))K_2\mathbf{H}_2\mathbf{J}_n\ddot{\mathbf{q}}_c &= -\bar{W}_2\text{pos}(\phi_2)u_2^+ \\ &\rightarrow \mathbf{A}_2\ddot{\mathbf{q}}_c = \mathbf{b}_2, \end{aligned} \quad (31)$$

where \mathbf{A}_2 and \mathbf{b}_2 are the values for the 2nd task in (5) and:

$$\bar{W}_2 = W_2/K_{s,l}, \quad (32)$$

where it has been assumed the same stiffness coefficient $K_{s,l}$ for both linear coordinates, i.e., $K_{s,l} = K_{g,s,x} = K_{g,s,y}$ and, hence, the linear motion of the robot tool given by (31) is in the same direction as the human operator's forces, which are detected by the guide sensor.

F. Level 3: Reference tracking

The following equality is considered for this level:

$$\begin{aligned} \mathbf{J}\ddot{\mathbf{q}}_c &= \ddot{\mathbf{p}}_{ref} + K_{T,v}\dot{\mathbf{e}} + K_{T,p}\mathbf{e} + \text{sign}(\dot{\mathbf{e}} + K_{T,p}K_{T,v}^{-1}\mathbf{e})u_3^+ \\ &\rightarrow \mathbf{A}_3\ddot{\mathbf{q}}_c = \mathbf{b}_3, \end{aligned} \quad (33)$$

where \mathbf{p}_{ref} is the reference for the tool pose, $\mathbf{e} = \mathbf{p}_{ref} - \mathbf{p}$ is the tool pose error, $K_{T,p}$ and $K_{T,v}$ are the correction gains for the position and velocity errors, respectively, the tool speed $\dot{\mathbf{p}}$ is obtained from the first-order kinematics (2), u_3^+ is the switching gain for the last term, and \mathbf{A}_3 and \mathbf{b}_3 are the values for the third task in (5). In this hybrid control law the switching term represents a conventional SMC used to compensate the term $\mathbf{J}\dot{\mathbf{q}}$ of the robot second-order kinematics (3), which yields two advantages: the Jacobian derivative is not required; and, due to the other continuous terms in the control action, the switching gain u_3^+ can be relatively small, reducing the chattering effects. Moreover, the magnitude of the error $\mathbf{e} = [x_{ref} \ y_{ref} \ z_{ref}]^T - [x \ y \ z]^T$ for the tool Cartesian position could be saturated to a value $e_{c,max}$ in order to avoid extremely large values in the control action of this level when the operator has moved the tool far away from the reference trajectory.

G. Level 4: Home configuration

This level applies only for *redundant* robots. The following equality is considered for “pushing” the robot to a home configuration \mathbf{q}_0 to avoid an uncontrolled self-motion of the redundant robot (i.e., the joint positions change without modifying the pose of the robot tool), which may lead achieving critical areas (joint limits, etc.):

$$\ddot{\mathbf{q}}_c = -K_{4,v}\dot{\mathbf{q}} + K_{4,p}(\mathbf{q}_0 - \mathbf{q}) \rightarrow \mathbf{A}_4\ddot{\mathbf{q}}_c = \mathbf{b}_4, \quad (34)$$

where $K_{4,v}$ and $K_{4,p}$ are the velocity and position correction gains and \mathbf{A}_4 and \mathbf{b}_4 are the values for the 4th task in (5).

H. Additional remarks

Control action. The joint accelerations have been considered as the SM discontinuous control action, which yields two advantages: the joint velocities are continuous (smoother control) and it allows to reach smoothly the equality constraints manifold and the boundary of the inequality constraints, see Section II-F. If the actual control action are the joint velocities (or positions), a pure single (or double) integrator can be applied to the discontinuous control signal to compute the actual continuous control action.

Time derivatives. The method requires the time derivatives $\dot{\mathbf{F}}_t$ and $\dot{\mathbf{F}}_g$ for the SMC in the first and second levels and $\dot{\mathbf{q}}$ for the tasks in the remaining levels. The simplest way is to use numerical differentiation, although filtering must be first applied when non-negligible noise is present, applied with care not to limit the bandwidth of the control law. A low-pass filter will be used in the experimental work described in Section V.

IV. CONTROLLER IMPLEMENTATION

The pseudo-code of the method is shown in Algorithm 1, which makes use of the following functions: kinematic function $\mathbf{I}(\mathbf{q})$; Jacobian matrices \mathbf{J} and \mathbf{J}_n ; Moore-Penrose pseudoinverse $(\cdot)^\dagger$, using a tolerance to set to zero the very small singular values [13]; *GetRobotStateAndForces()*, which returns the current robot state $\{\mathbf{q}, \dot{\mathbf{q}}\}$ and the current force vectors \mathbf{F}_t and \mathbf{F}_g , which have been already filtered by the sensor electronics; and *SendToJointControllers*($\ddot{\mathbf{q}}_c$), which sends the current reference acceleration vector to the joint controllers. The computation time per iteration using compiled C code in a modern computer was around 15 microseconds for the experiments shown in the following Section.

V. REAL EXPERIMENTS

The setup used for the experiment consists of (see Fig. 1): a Sawyer cobot; a force sensor Nano25 (*treatment* sensor) located at the end of the last link of the robot; another force sensor Nano25 (*guide* sensor) located with an offset of 80mm along the end-effector Z-axis; a rectangular rigid unit, 100mm in length as support for the guide sensor; a guiding element consisting of a T-shaped rigid handle of 150x70x5mm; a mock-up surface treatment tool consisting of a 43x43x10mm cylinder; and a flat rectangular plastic object of 190x95x3mm as target.

Algorithm 1: SMC collaborative controller (executed at sampling time of T_s seconds)

```

1 [ $\mathbf{q}, \dot{\mathbf{q}}, \mathbf{F}_t, \mathbf{F}_g$ ] = GetRobotStateAndForces ();
2  $\mathbf{p} = \mathbf{l}(\mathbf{q})$  ; // Eq. (1)
3  $\dot{\mathbf{p}} = \mathbf{J}\dot{\mathbf{q}}$  ; // Eq. (2)
4  $\dot{\mathbf{p}}_{ref} = (\mathbf{p}_{ref} - \mathbf{p}_{ref,prev})/T_s$  ; // Derivative
5  $\ddot{\mathbf{p}}_{ref} = (\dot{\mathbf{p}}_{ref} - \dot{\mathbf{p}}_{ref,prev})/T_s$  ; // Derivative
6  $\dot{\mathbf{F}}_t = (\mathbf{F}_t - \mathbf{F}_{t,prev})/T_s$  ; // Derivative
7  $\dot{\mathbf{F}}_g = (\mathbf{F}_g - \mathbf{F}_{g,prev})/T_s$  ; // Derivative
8 If  $|\mathbf{e}_c| > e_{c,max}$  then  $\mathbf{e}_{c,s} = \mathbf{e}_c(e_{c,max}/|\mathbf{e}_c|)$ ;
9 else  $\mathbf{e}_{c,s} = \mathbf{e}_c$  ; // Saturation
10  $\phi_1 = \begin{bmatrix} F_{t,z} - F_{z,ref} + K_{1,z}\dot{F}_{t,z} \\ F_{t,\alpha} + K_{1,\alpha}\dot{F}_{t,\alpha} \\ F_{t,\beta} + K_{1,\beta}\dot{F}_{t,\beta} \end{bmatrix}$  ; // Eqs. (15), (22)–(24)
11  $\phi_2 = \sqrt{F_{g,x}^2 + F_{g,y}^2} - F_{l,0} + K_2 \frac{\dot{F}_{g,x}F_{g,x} + \dot{F}_{g,y}F_{g,y}}{\sqrt{F_{g,x}^2 + F_{g,y}^2}}$  ; // Eqs. (16), (28)
12  $\mathbf{A}_1 = \mathbf{K}_1\mathbf{H}_1\mathbf{J}_n$  ; // Eq. (26)
13  $\mathbf{b}_1 = -\bar{\mathbf{W}}_1 \text{sign}(\phi_1) u_1^+$  ; // Eq. (26)
14  $\mathbf{A}_2 = v2dm(\text{pos}(\phi_2))K_2\mathbf{H}_2\mathbf{J}_n$  ; // Eq. (31)
15  $\mathbf{b}_2 = -\bar{\mathbf{W}}_2 \text{pos}(\phi_2) u_2^+$  ; // Eq. (31)
16  $\mathbf{A}_3 = \mathbf{J}$  ; // Eq. (33)
17  $\mathbf{b}_3 = \dot{\mathbf{p}}_{ref} + K_{T,v}\dot{\mathbf{e}} + K_{T,p}\mathbf{e} + \text{sign}(\dot{\mathbf{e}} + K_{T,p}K_{T,v}^{-1}\mathbf{e})u_3^+$  ; // Eq. (33)
18  $\mathbf{A}_4 = \mathbf{I}$  ; // Eq. (34)
19  $\mathbf{b}_4 = -K_{4,v}\dot{\mathbf{q}} + K_{4,p}(\mathbf{q}_0 - \mathbf{q})$  ; // Eq. (34)
20  $\ddot{\mathbf{q}}_{c,1} = \mathbf{A}_1^\dagger\mathbf{b}_1$  ; // Eq. (6)
21  $\mathbf{N}_1 = \mathbf{I} - \mathbf{A}_1^\dagger\mathbf{A}_1$  ; // Eq. (7)
22  $\ddot{\mathbf{q}}_{c,2} = \ddot{\mathbf{q}}_{c,1} + (\mathbf{A}_2\mathbf{N}_1)^\dagger(\mathbf{b}_2 - \mathbf{A}_2\ddot{\mathbf{q}}_{c,1})$  ; // Eq. (6)
23  $\mathbf{N}_2 = \mathbf{N}_1(\mathbf{I} - (\mathbf{A}_2\mathbf{N}_1)^\dagger(\mathbf{A}_2\mathbf{N}_1))$  ; // Eq. (7)
24  $\ddot{\mathbf{q}}_{c,3} = \ddot{\mathbf{q}}_{c,2} + (\mathbf{A}_3\mathbf{N}_2)^\dagger(\mathbf{b}_3 - \mathbf{A}_3\ddot{\mathbf{q}}_{c,2})$  ; // Eq. (6)
25  $\mathbf{N}_3 = \mathbf{N}_2(\mathbf{I} - (\mathbf{A}_3\mathbf{N}_2)^\dagger(\mathbf{A}_3\mathbf{N}_2))$  ; // Eq. (7)
26  $\ddot{\mathbf{q}}_{c,4} = \ddot{\mathbf{q}}_{c,3} + (\mathbf{A}_4\mathbf{N}_3)^\dagger(\mathbf{b}_4 - \mathbf{A}_4\ddot{\mathbf{q}}_{c,3})$  ; // Eq. (6)
27 SendToJointControllers( $\ddot{\mathbf{q}}_{c,4}$ );
28  $\mathbf{p}_{ref,prev} = \mathbf{p}_{ref}$  ; // For next iteration
29  $\dot{\mathbf{p}}_{ref,prev} = \dot{\mathbf{p}}_{ref}$  ; // For next iteration
30  $\mathbf{F}_{t,prev} = \mathbf{F}_t$  ; // For next iteration
31  $\mathbf{F}_{g,prev} = \mathbf{F}_g$  ; // For next iteration

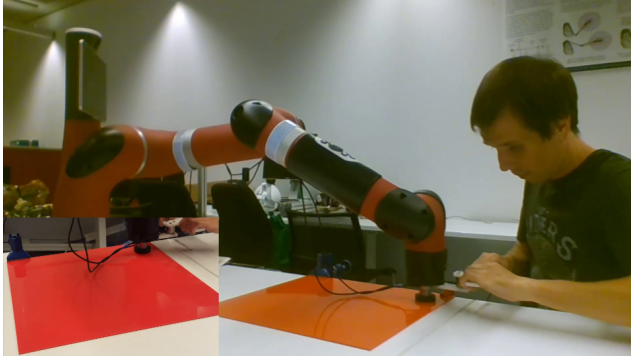
```

A. Experiment conditions and parameter values

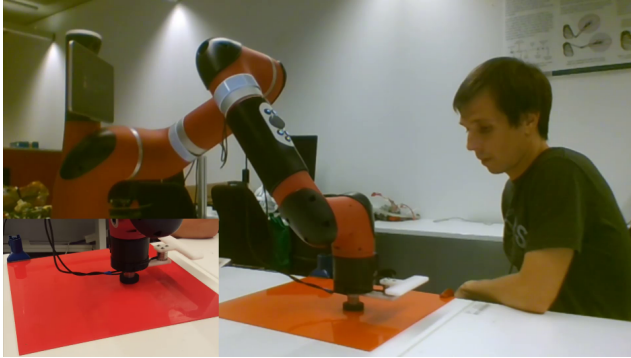
- The signals of both force sensors are filtered using a first-order low-pass filter with a cutoff frequency of 73 Hz, which is implemented in the sensor electronics.
- The control period T_s is set to 20 milliseconds and, hence, the frequency of the SM control action is equal to $(2 \cdot 0.02)^{-1} = 25$ Hz, which is lower than the filter cutoff frequency mentioned above, as required.
- The reference joint accelerations $\ddot{\mathbf{q}}_c$ are integrated to obtain the joint velocities $\dot{\mathbf{q}}_c$ sent to the robot controller.
- The switching gains u_i^+ and weights $\bar{\mathbf{W}}_i$ of the SMC have been empirically tuned as small as possible to reduce the chattering effect but guaranteeing that the SM behavior of



(a) video: 0m 29s; graph: 17s



(b) video: 0m 34s; graph: 22s



(c) video: 0m 50s; graph: 38s

Fig. 2. Frames taken from the first experiment video: time instant is indicated for each frame.

- the control action remains effective for the task at hand.
- e) Level 1: $F_{z,ref} = -10\text{N}$, $u_1^+ = 0.06$, $\bar{W}_{1,z} = 0.75$, $\bar{W}_{1,\alpha} = \bar{W}_{1,\beta} = 12$, and $K_{1,i} = 0.15$.
 - f) Level 2: $F_{l,0} = 3\text{N}$, $u_2^+ = 0.2$ and $K_2 = 0.075$.
 - g) Level 3: $K_{T,v} = 4.5$, $K_{T,p} = 2.0$, $e_{c,max} = 0.025$, $u_3^+ = 0.01$ and a circular reference trajectory of radius 80mm and period 10 seconds. The motion of the reference point is stopped when the tracking error is above a threshold, which allows to resume the reference tracking at the same point of the circular trajectory where it was previously interrupted by the user.
 - h) Level 4: $K_{4,v} = 4.5$, $K_{4,p} = 0.75$ and $\mathbf{q}_0 = [1.04^\circ \ -36.41^\circ \ -1.39^\circ \ 95.24^\circ \ 2.08^\circ \ 30.19^\circ \ 188.29^\circ]^\text{T}$.
- Two experiments were then carried out with increasing

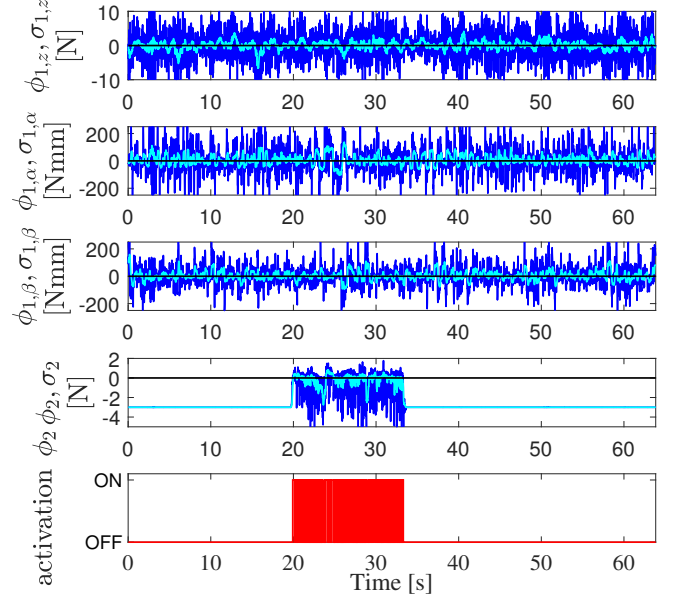


Fig. 3. Behavior of the constraints in the first experiment. First to fourth plots: constraint functions for Level 1 and Level 2, the modified constraint function ϕ_i is in dark-blue, whereas the original constraint function σ_i is in light-cyan. Fifth plot: activation of the inequality constraint in Level 2.

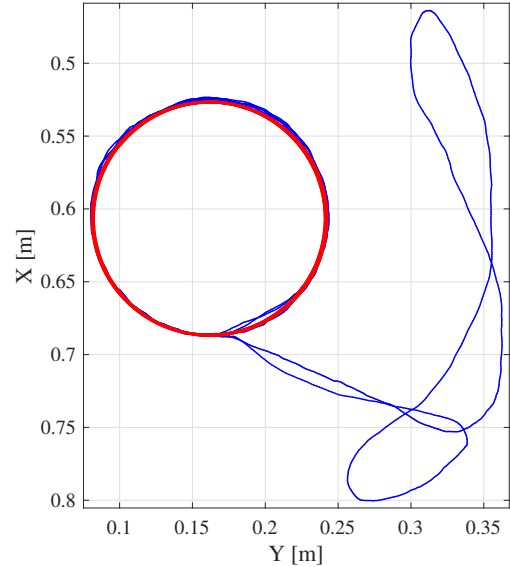


Fig. 4. Trajectory followed by the tool (thin-blue line) in the first experiment and circular reference trajectory (thick-red line).

levels of complexity in the collaborative task being undertaken, one with a static flat surface and the other with a dynamically changing surface to operate on.

B. Results

The first experiment was carried out with a flat target object resting on a table (please refer to the video at <https://media.upv.es/player/?id=019da250-1b7d-11e8-b454-edb485e1cbe7>). Fig. 2 shows several frames extracted from the video: Fig. 2a (time instant 29s in the video) shows how the robot motion is automatically tracking the circular reference trajectory;

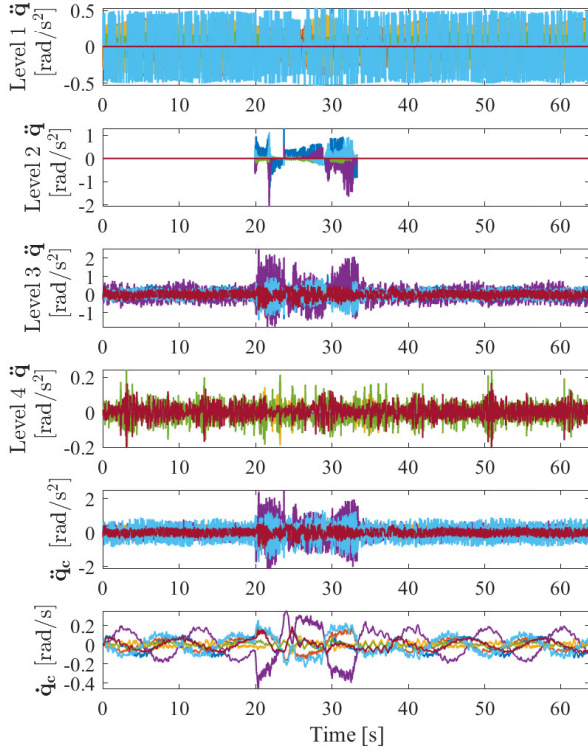
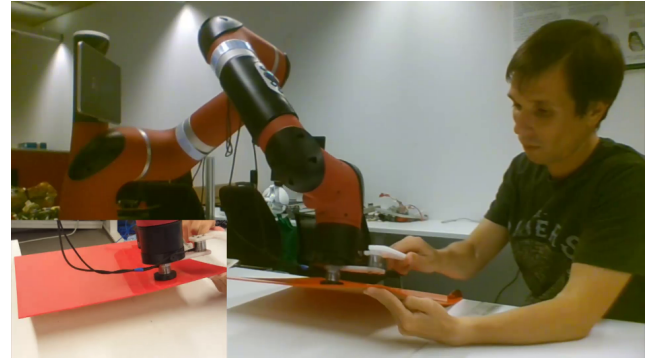


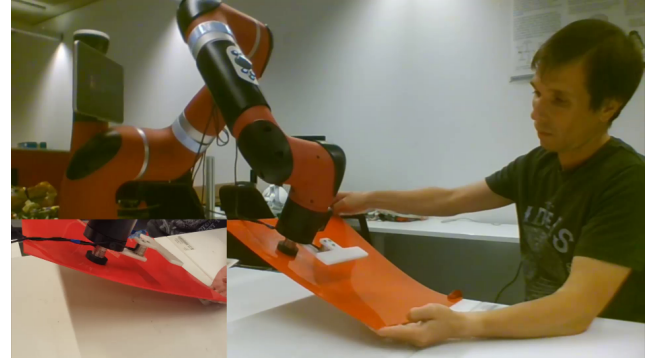
Fig. 5. Control actions for the first experiment: contribution of each priority level to the reference joint accelerations in the first four plots, reference joint accelerations in the fifth plot and reference joint velocities to be sent to the robot controller in the bottom plot.

Fig. 2b (time instant 34s) shows how the robot motion switches its behavior when the tool is guided by the user; and Fig. 2c (time instant 50s) shows how the robot smoothly returns to track the circular reference trajectory after the user has stopped guiding the tool. Fig. 3 shows the performance of the current approach in terms of constraint functions and activation of the inequality constraint used to guide the robot tool. In particular, it can be seen that the equality constraints $\{\sigma_{1,z}, \sigma_{1,\alpha}, \sigma_{1,\beta}\}$ are switching around zero as expected, indicating a proper surface treatment operation: the tool orientation is kept perpendicular to the object surface and the tool pressure on the surface is maintained. It can also be seen in the fifth plot how the Level 2 constraint ϕ_2 , which is used to guide the robot tool, is activated when small force magnitudes are detected (as shown in the fourth plot), effectively meaning the tool can be guided by the operator using slight indicative actions.

Fig. 4 shows the trajectory followed by the robot tool, where two distinctive paths are clearly differentiated: an automatic mode path (circular), and the arbitrary path followed by the robot under user guidance. Note how the robot tool automatically gets back to the same point where the reference tracking was previously interrupted by the user and the circular reference motion is resumed. Fig. 5 shows the



(a) video: 1m 55s; graph: 103s



(b) video: 2m 36s; graph: 144s

Fig. 6. Frames of the video of the second experiment: the time instant is indicated for each frame.

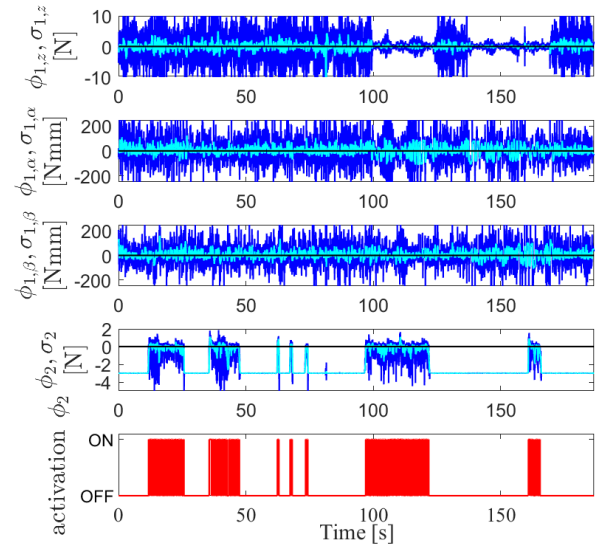


Fig. 7. Behavior of the constraints in the second experiment. Top four plots: constraint functions for Level 1 and Level 2, the modified constraint function ϕ_i is in dark-blue, whereas the original constraint function σ_i is in light-cyan. Bottom plot: activation of the inequality constraint in Level 2.

control commands computed during the experiment, where it can be seen that all levels contribute to the reference joint accelerations. Although the commands computed in Level 4

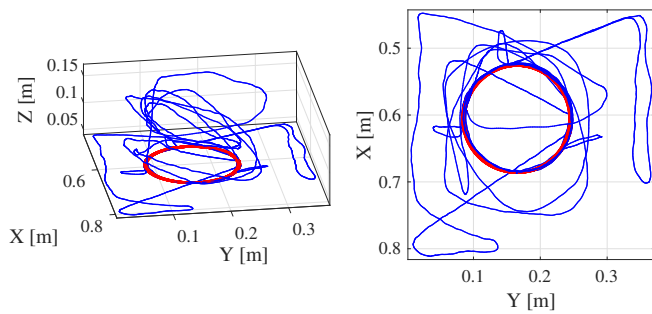


Fig. 8. Trajectory followed by the tool (thin-blue line) in the second experiment and circular reference trajectory (thick-red line).

are small compared to those computed at the higher levels, they cannot be neglected since they allow to avoid a bias self-motion for a redundant robot which may lead the robot towards a critical region, e.g., joint limits.

A second experiment was conducted where the position and orientation of the flat target object was changed dynamically in order to show the adaptability and robustness of the proposed method. The video can be played at <https://media.upv.es/player/?id=e9b73f50-1b7e-11e8-b454-edb485e1cbe7> and Fig. 6 shows several frames from the video: Fig. 6a and Fig. 6b (interval 1m55s–2m36s) show how the user guides the robot tool or leaves it in automatic mode when the flat object changes its position, orientation and stiffness, i.e., when the flat object is held above the table. It is interesting to remark that the tool is kept perpendicular to the object surface despite the changes in the object position and orientation, as can be seen in the video. Fig. 7 shows the constraint functions and activation of the inequality constraint used to guide the robot tool. In addition to the comments made for the first experiment, in this experiment it is worth noting that the amplitude of the chattering for the equality constraint $\phi_{1,z}$ is significantly reduced for the time intervals 101s–128s and 139s–173s in the graph (1m53s–2m20s and 2m31s–3m05s in the video), which corresponds to the phase when the flat surface is held above the table. This is because the stiffness coefficient $K_{t,s,z}$ decreases substantially when the flat object has no support, and since the value used for $\bar{W}_{1,z}$ remains the same, from (27) it means that the actual value for $W_{1,z}$ is also reduced. Thus, resulting in a lower chattering amplitude, see (17). Therefore, it is suggested that an adaptive switching gain [14] could be considered for the proposed SMC in order to automatically adapt the switching gain to the perceived changes in the target stiffness. Fig. 8 depicts the trajectory followed by the robot tool for the second experiment, while the magnitudes of control commands are similar to those shown above for the previous experiment.

VI. CONCLUSIONS

This work has proposed an original control framework for human-robot collaboration to cooperatively perform surface treatment tasks such as polishing, grinding, etc. The shared strategy effectively couples the human operator natural abilities and fine skills in guiding as needed through the user

interactions, while the autonomous physical agent is able to maintain desired operating procedures under strict constraints to guarantee the success of the task. For this purpose, two force sensors attached to the manipulator end-effector and tool have been considered: one sensor is used to properly accomplish the surface treatment task, while the second one is used by the operator to arbitrarily guide the robot tool. The effectiveness of the method has been shown experimentally using a redundant 7R manipulator in a simulated surface polishing exercise successfully working in real time.

As further work, several improvements could be considered for the proposed approach: using an adaptive SMC in order to automatically adapt the switching gain to the perceived changes; replacing the discontinuous commutation functions $\text{sign}(\cdot)$ and $\text{pos}(\cdot)$ of the SMC by continuous approximations (e.g., the hyperbolic tangent function and positive hyperbolic tangent function, respectively) in order to reduce the chattering issue; using a dynamical or high-order SMC to reduce the chattering drawback; etc.

REFERENCES

- [1] L. Pilny and G. Bissacco, “Development of on the machine process monitoring and control strategy in robot assisted polishing,” *CIRP Annals - Manufacturing Technology*, vol. 64, no. 1, pp. 313 – 316, 2015.
- [2] L. Arnal, J. E. Solanes, J. Molina, and J. Tornero, “Detecting dings and dents on specular car body surfaces based on optical flow,” *Journal of Manufacturing Systems*, vol. 45, pp. 306–321, 2017.
- [3] F. Tian, Z. Li, C. Lv, and G. Liu, “Polishing pressure investigations of robot automatic polishing on curved surfaces,” *The International Journal of Advanced Manufacturing Technology*, vol. 87, no. 1, pp. 639–646, 2016.
- [4] F. Dimeas and N. Aspragathos, “Online stability in human-robot cooperation with admittance control,” *IEEE Transactions on Haptics*, vol. 9, no. 2, pp. 267–278, 2016.
- [5] J. Vogel, S. Haddadin, B. Jarosiewicz, J. Simeral, D. Bacher, L. Hochberg, J. Donoghue, and P. van der Smagt, “An assistive decision-and-control architecture for force-sensitive hand-arm systems driven by human-machine interfaces,” *The International Journal of Robotics Research*, vol. 34, no. 6, pp. 763–780, 2015.
- [6] S.-J. Huang, Y.-C. Liu, and S.-H. Hsiang, “Robotic end-effector impedance control without expensive torque/force sensor,” *International Journal of Mechanical, Aerospace, Industrial, Mechatronic and Manufacturing Engineering*, vol. 7, no. 7, pp. 1446 – 1453, 2013.
- [7] E. Engeberg, S. Meek, and M. Minor, “Hybrid force-velocity sliding mode control of a prosthetic hand,” *IEEE Transactions on Biomedical Engineering*, vol. 55, no. 5, pp. 1572–1581, 2008.
- [8] C. Edwards and S. Spurgeon, *Sliding Mode Control: Theory and Applications*, 1st ed. UK: Taylor & Francis, 1998.
- [9] Y. Nakamura, H. Hanafusa, and T. Yoshikawa, “Task-priority based redundancy control of robot manipulators,” *The Int. Journal of Robotics Research*, vol. 6, no. 2, pp. 3–15, 1987.
- [10] B. Siciliano and J. Slotine, “A general framework for managing multiple tasks in highly redundant robotic systems,” in *Proceedings of the Fifth Int. Conference on Advanced Robotics (ICAR’91)*, Pisa, Italy, 1991, pp. 1211–1216, 1991.
- [11] V. Utkin, J. Guldner, and J. Shi, *Sliding Mode Control in Electro-Mechanical Systems*, 2nd ed. London: Taylor & Francis, 2009.
- [12] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*. London, UK: Springer-Verlag, 2009.
- [13] G. Golub and C. Van Loan, *Matrix Computations*, 3rd ed. Baltimore, MD: The Johns Hopkins University Press, 1996.
- [14] F. Plestan, Y. Shtessel, V. Bregeault, and A. Poznyak, “New methodologies for adaptive sliding mode control,” *International Journal of Control*, vol. 83, no. 9, pp. 1907–1919, 2010.