

Document downloaded from:

<http://hdl.handle.net/10251/157497>

This paper must be cited as:

Gracia Calandin, LI.; Solanes Galbis, JE.; Muñoz-Benavent, P.; Valls Miro, J.; Perez-Vidal, C.; Tornero Montserrat, J. (2019). Human-robot collaboration for surface treatment tasks. *Interaction Studies*. 20(1):148-184. <https://doi.org/10.1075/is.18010.gra>



The final publication is available at

<https://doi.org/10.1075/is.18010.gra>

Copyright John Benjamins Publishing Company

Additional Information

## Human-robot collaboration for surface treatment tasks

Luis Gracia<sup>1,\*</sup>, J. Ernesto Solanes<sup>1</sup>, Pau Muñoz-Benavent<sup>1</sup>, Jaime Valls Miro<sup>2</sup>,  
Carlos Perez-Vidal<sup>3</sup> & Josep Tornero<sup>1</sup>

<sup>1</sup>Instituto IDF, Universitat Politècnica de València, 46022 Valencia, Spain

<sup>2</sup>Centre for Autonomous Systems, University of Technology Sydney, NSW 2007 Sydney,  
Australia

<sup>3</sup>Departamento de Ingeniería de Sistemas y Automática, Universidad Miguel Hernández,  
03202 Elche, Spain

### ARTICLE HISTORY

Compiled January 28, 2019

### ABSTRACT

This paper presents a human-robot closely collaborative solution to cooperatively perform surface treatment tasks such as polishing, grinding, finishing, deburring, etc. The proposed scheme is based on task priority and non-conventional sliding mode control. Furthermore, the proposal includes two force sensors attached to the manipulator end-effector and tool: one sensor is used to properly accomplish the surface treatment task, while the second one is used by the operator to guide the robot tool. The applicability and feasibility of the proposed collaborative solution for robotic surface treatment are substantiated by experimental results using a redundant 7R manipulator: the Sawyer collaborative robot.

### KEYWORDS

Cooperative control; robust control; robot system; sliding mode control

---

\*Corresponding author L. Gracia. Email: luigraca@isa.upv.es

## 1. Introduction

The automation of industrial processes has generated great improvements in terms of cost reduction, operator safety and comfort, as well as the quality of the end product over more traditional hand-made methods. However, there are currently many industrial processes within the production of products that are carried out manually due to their complexity. Although it is a true fact that the sensorimotor stability and strength of industrial manipulators are much better than that of humans, nowadays robotic manipulators cannot compete with the flexibility/adaptability of humans. That is why there is currently a strong tendency to combine robots and humans for the accomplishment of complex tasks through collaboration.

One of the least automated processes is the quality control of surfaces [19, 29]. This is primarily attributed to the fact that automated process remain elusive in meeting strict requirements when it comes to short cycle time, low cost and the high quality achieved in other manufacturing industries [20, 24]. As such, surface treatment operations and quality control continues to be mainly a manual process being carried out by skilled workers, which give rise to issues such as subjectivity in the evaluation criteria, human errors and the like [2, 33].

In surface treatment operations the tool has to be in contact with the product surface to apply a specific treatment, e.g., grinding, polishing, deburring, etc. As such, the forces exerted by the tool have to be properly controlled. Moreover, the tool should be kept perpendicular to the surface at all times to homogenize the pressure on all contact points [27, 32]. Due to the complexity of the shape of the surfaces of the product, human operators have difficulties in maintaining a constant pressure and tool perpendicularity to the surface at all times, which causes deficiencies in the surface treatment. In the same way, when the product to be treated is part of a production line where it is in motion, or its morphology is constantly changing —e.g., for the case of car bodies [2, 22] or ceramics [8])— it is difficult for robotic manipulators to carry out automatic surface treatment operations efficiently given the inherent uncertainty regarding the position and velocity of the product being handled.

To mitigate the drawbacks of both, manual and robotic automatic surface treatment,

this work proposes a human-robot closely collaborative solution which adopts the form of a human operator performing the task of guiding the tool along the surface of the object to be treated, together with a robot manipulator to automatically maintain both the pressure of the tool on the surface and the tool perpendicularity to the surface, ensuring a flexible surface treatment. For this purpose, this work uses multi-task, conventional and non-conventional sliding mode control (SMC) and two force/torque sensors attached to the manipulator end-effector and tool.

While solutions of robots working collaboratively alongside humans in polishing tasks exists (e.g., see <https://www.youtube.com/watch?v=kOPVvYapE1Q>, where a Universal Robot is used in the process of loudspeakers polishing), to the best of the authors' knowledge this is the first work proposing a human-robot closely collaborative solution to cooperatively perform surface treatment operations, and also the first work using two force/torque sensors at the end-effector of the robot manipulator. In this sense, one force sensor is used to properly accomplish the surface treatment task, i.e., to attain the desired pressure between the tool and the surface being treated as well as to keep the tool orientation perpendicular to the surface; whereas the second force sensor is used by the human operator to guide the robot tool while the surface treatment is being performed. It is important to 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 would represent a disturbance to properly accomplish the surface treatment task. For instance, if the same sensor is used for both purposes and the operator exerts a force in the negative  $Z$ -axis direction of the robot tool that is larger than the desired pressure for the surface treatment task, the robot would move away the tool from the treated object and the contact would be lost.

It is worthy to mention that the proposed scheme for human-robot collaboration represents a reactive controller with no high-level planning. However, the proposed SMC framework could be combined with other intelligent strategies, such as learning-based control, in order to improve human-robot collaboration in terms of tracking performance, security during the interaction, etc. However, this is out of the scope of this work.

Next, a literature review is presented about the main aspects concerning this work: robot guidance and SMC techniques for robot force control and human-robot collaboration.

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 most commonly used method to convert these measurements into kinematic instructions to the robot is through compliance control, which establishes a direct relationship between the measured forces and the changes in the robot position [6, 15]. Yet other variants and methods can be found in the literature. For instance, in [18] a force tracking method under the impedance control framework was extended to also account for uncertain human limb dynamics. In [36] a decision-and-control architecture was proposed for hand-arm systems with “soft robotics” capabilities via dedicated human-machine interfaces. In [13] a mathematical relation between the velocity of the human-robot interaction point and the force applied by the human operator was established using impedance control for handling tasks.

Other approaches tackling the problems of robot force control and human-robot collaboration are based on *SMC techniques*. Concretely, in [28] SMC is used to suppress impact forces when contacting the environment and be able to continue with a stable robot motion. In [3] a hybrid position/force control scheme was proposed using first- and second-order SMC for position and force control, respectively. In [12] an impedance control structure was proposed for monitoring the contact force between the end-effector and the environment, and a model-free fuzzy SMC strategy was employed to design the position and force controllers. In [9], several methods were developed to control a prosthetic hand and the best results in terms of unwanted force overshoot were obtained using a SMC with force, position and velocity feedback. In [39] a non-singular terminal SMC was developed to ensure trajectory tracking precision for the case of a lower limb rehabilitation parallel robot. In [14] a proxy-based SMC was proposed to obtain effective tracking during normal operations for flexible joint manipulators working close to humans. In [38] a robust SMC was proposed that relied on basic information from the human subject to handle model uncertainties due to biomechanical variation of patients using an upper limb rehabilitation robot. An SMC

consisting of a PID sliding surface and a fuzzy hitting control law was developed in [37] to guarantee robust tracking performance and reduce the chattering effect for a class of robot-assisted therapeutic exoskeleton. A fuzzy SMC was presented in [21] using a non-linear model for trajectory tracking of micro robots in the human vasculature system. Moreover, SMC has been used in the field of robot force control not only to improve controller robustness but also to improve force estimation by means of a sliding perturbation observer to avoid the use of expensive force sensors, e.g., see [26].

It is worthy to mention that human-robot interaction is increasing not only in industrial scenarios but also for social purposes, see [1, 5, 10, 25, 35] among others.

It is interesting to remark that this work about human-robot collaboration for surface treatment tasks is included in a special issue entitled “Human Robot Collaborative Intelligence” and mainly covers the following topics: “collaborative control of human-robot teams” and “applications of human robot collaborative intelligence methods to industrial manufacture”.

The paper is organized as follows: next section introduces some preliminaries, while Section 3 develops the SMC theory used in this work. The proposed collaborative solution for robotic surface treatment is presented in Section 4. A simulation is shown in Section 5 to illustrate the proposed non-conventional SMC and to compare it with conventional SMC. The implementation of the controller is detailed in Section 6. The feasibility of the proposed approach is substantiated by experimental results in Section 7 using a redundant 7R manipulator: the Rethink Sawyer collaborative robot. Finally, conclusions are drawn in Section 8.

## 2. Preliminaries

**Kinematics.** Following the standard notation [4], the robot *pose*  $\mathbf{p}$  depends on the robot *configuration*  $\mathbf{q}$  as follows:

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

where the nonlinear function  $\mathbf{l}$  is called the kinematic function of the robot. The first- and second-order kinematics of the pose vector  $\mathbf{p}$  result in:

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

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

where  $\mathbf{J}$  is the Jacobian matrix of the robot.

**Robot control.** This work assumes the existence of a low-level robot controller in charge of achieving a particular joint acceleration from the commanded acceleration  $\ddot{\mathbf{q}}_c$ , and that its dynamics is fast enough compared to that of  $\ddot{\mathbf{q}}_c$ . Hence, the relationship:

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

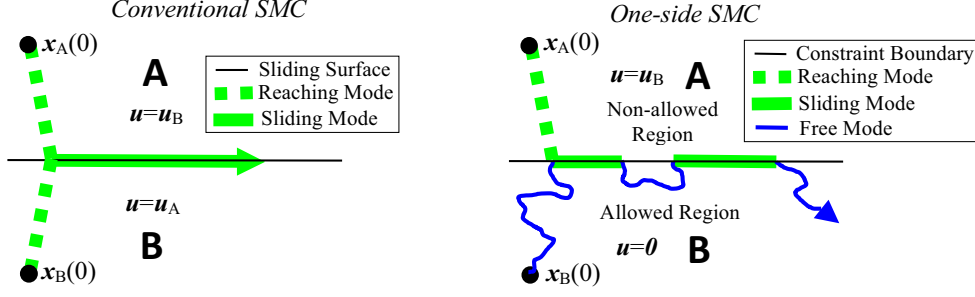
holds approximately true, where  $\mathbf{d}_c$  represents inaccuracies due to disturbances. Note that the *dynamic model* of the robot system should be taken into account to properly design the mentioned underlying joint controller.

**Task-priority scheme.** The task-priority strategy [23] allows to tackle several objectives simultaneously assigning an order of priority to each one. Let us consider  $M$  tasks which consist in calculating the commanded 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, where  $i = 1$  is used for highest priority.

The solution  $\ddot{\mathbf{q}}_{c,M}$  that hierarchically minimizes the error of equations in (5) is given



**Figure 1.** Graphical comparison between conventional SMC (left) and one-side SMC (right).

by the following recursive formulation, proposed in [30]:

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

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

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

where  $\mathbf{I}$  and  $\mathbf{0}$  denote the identity matrix and zero column vector, respectively, of suitable size, 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, respectively, for the set of first  $i$  tasks.

### 3. Sliding Mode Control

The proposed method to accomplish the cooperative surface treatment task using a robotic system is based on satisfying a set of equality and inequality constraints, defined in Section 4. The fulfillment of these constraints will be achieved using SMC theory to benefit from the typical advantages of this type of controllers, such as robustness and low computational cost. However, conventional SMC can only be used to satisfy equality constraints. Hence, this section also presents a non-conventional SMC algorithm, coined as one-side SMC, developed to satisfy inequality constraints. Fig. 1 shows a graphical two dimensional example to illustrate both conventional and one-side SMCs.



### 3.1. Conventional SMC to satisfy equality constraints

For conventional SMC (see Fig. 1-left) the state space of the system is divided into two regions,  $\mathbf{A}$  and  $\mathbf{B}$ , separated by the sliding surface. The value of the control action  $\mathbf{u}$  when the system state is in region  $\mathbf{A}$  is such that it “pushes” the system into region  $\mathbf{B}$ , namely  $\mathbf{u}_B$ . Analogously, when the system state is in region  $\mathbf{B}$ , the value of control action is such that it pushes the system into region  $\mathbf{A}$ , namely  $\mathbf{u}_A$ . Hence, regardless of whether the system starts in region  $\mathbf{A}$  -  $\mathbf{x}_A(0)$ , or  $\mathbf{B}$  -  $\mathbf{x}_B(0)$ , it evolves to the sliding surface in what is referred as *reaching mode* [7]. Once the system has reached the sliding surface, the system is kept on it by a control action  $\mathbf{u}$  that switches between  $\mathbf{u}_A$  and  $\mathbf{u}_B$  at a theoretically infinite frequency, which is known as *sliding mode* (SM) [7]. A *continuous equivalent control* [34] can be obtained for the SM phase, i.e., the control required to keep the system on the sliding surface. Therefore, SMC produces such control action without explicitly computing it and with low computational cost, which is a typical advantage of SMC strategies [34].

The theorem below presents a conventional SMC designed to satisfy equality constraints. This theorem is given for completeness and for comparison with the proposed one-side SMC, which is designed below to satisfy inequality constraints.

**Theorem 3.1.** *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 of dimension  $n_x$ ,  $\mathbf{d}(t)$  is an unmeasured disturbance or model uncertainty of dimension  $n_d$ ,  $\mathbf{u}(t)$  is the control input vector (possibly discontinuous) of dimension  $n_u$ ,  $\mathbf{f}$  is a vector field and  $\mathbf{g}$  is a set of vector fields.

Consider also that the system 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  $\Phi_{eq}$  of the state space compatible with the constraints on state  $\mathbf{x}$  is given by:

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

with  $i = 1, \dots, N_{eq}$ .

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  $\Phi_{eq}$  in finite time and remains there henceforth:

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

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

where  $\phi_{eq}$  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  $\mathbf{L}_{\mathbf{g}}\phi_{eq,i} = \frac{\partial\phi_{eq,i}}{\partial\mathbf{x}}\mathbf{g}$  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  $\mathbf{L}_{\mathbf{g}}\phi_{eq}$  contains the row vectors  $\mathbf{L}_{\mathbf{g}}\phi_{eq,i}$  of all equality constraints,  $\text{sign}(\cdot)$  represents the sign function (typically used in SMC), positive scalar  $u_{eq}^+$  is the so-called switching gain, which can be either constant or varying in time,  $\mathbf{W}_{eq}$  is a diagonal matrix representing the switching gain weights for the constraints,  $\|\cdot\|_1$  represents the 1-norm (also known as the Taxicab norm) and function  $\text{diag}_{\min}(\cdot)$  computes the minimum value of the diagonal elements of a matrix.

**Proof.** The proof can be obtained straightforward from the Proof 2.1 in [34] and its generalization. Details omitted for brevity.  $\square$

### 3.2. One-side SMC to satisfy inequality constraints

The one-side SMC proposed in this work is graphically represented in Fig. 1-right and is used to satisfy inequality constraints. In this case, the state space of the system is divided into the non-allowed region  $\mathbf{A}$  and the allowed region  $\mathbf{B}$ , which are separated by the constraint boundary. Similarly to conventional SMC, when the system state is in the non-allowed region  $\mathbf{A}$  the control action  $\mathbf{u} = \mathbf{u}_B$  pushes the system into the allowed region  $\mathbf{B}$ . But in contrast to conventional SMC, when the system state is in the allowed region  $\mathbf{B}$ , no control action is applied, i.e.,  $\mathbf{u} = \mathbf{0}$ . Hence, if the system starts in region  $\mathbf{A}$ , i.e.,  $\mathbf{x}_A(0)$ , it evolves in reaching mode to the sliding surface. Nevertheless,

when the system starts in the allowed region  $\mathbf{B}$ , the system state can “freely” evolve according to some other criterion, e.g., a control law for reference tracking. Therefore, only when the state trajectory tries by itself to leave the allowed region, the one-side SMC will make  $\mathbf{u}$  switch between  $\mathbf{0}$  and  $\mathbf{u}_B$  at a theoretically infinite frequency, which can be seen as an ideal SM behavior [7].

The theorem below presents the proposed one-side SMC in order to satisfy inequality constraints.

**Theorem 3.2.** *Consider the dynamical system given by (8) and consider also that the system 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 inequality constraint function. Thus, the region  $\Phi_{in}$  of the state space compatible with the constraints on state  $\mathbf{x}$  is given by:*

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

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

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  $\Phi_{in}$  in finite time and remains there henceforth:

$$\mathbf{v}2\mathbf{m}(\text{pos}(\boldsymbol{\phi}_{in})) \mathbf{L}_{\mathbf{g}}\boldsymbol{\phi}_{in} \mathbf{u} = -\mathbf{W}_{in} \text{pos}(\boldsymbol{\phi}_{in}) u_{in}^+ \quad (13)$$

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

where function  $\mathbf{v}2\mathbf{m}(\cdot)$  converts a vector into a diagonal matrix, function  $\text{pos}(\cdot)$  represents the positive function (i.e.,  $\text{pos}(x)$  is equal to 0 if  $x < 0$  and equal to 1 if  $x > 0$ ),  $\boldsymbol{\phi}_{in}$  is a column vector with all the inequality constraint functions  $\phi_{in,i}$ , matrix  $\mathbf{L}_{\mathbf{g}}\boldsymbol{\phi}_{in}$  contains the row vectors  $\mathbf{L}_{\mathbf{g}}\phi_{in,i}$  of all inequality constraints, the scalar  $L_f \phi_{in,i} = \frac{\partial \phi_{in,i}^T}{\partial \mathbf{x}} \mathbf{f}$  and the row vector  $\mathbf{L}_{\mathbf{g}}\phi_{in,i} = \frac{\partial \phi_{in,i}^T}{\partial \mathbf{x}} \mathbf{g}$  denote the Lie derivatives of the inequality 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 for the inequality constraints

and  $n_a$  is the number of active inequality constraints, i.e., those with  $\phi_{in,i} \geq 0$ .

Note that, the expression  $\text{v2m}(\text{pos}(\phi_{in}))$  on the left-side of (13) is used to obtain the trivial scalar equation  $0 = 0$  for the non-active inequality constraints (i.e., those with  $\phi_{in,i} < 0$ ) and, hence, no degrees of freedom of the system are used by these constraints.

**Proof.** Firstly, the inequality constraint vector is partitioned into two subvectors  $\phi_{in} = [\phi_{in}^{n_a \text{ T}} \quad \phi_{in}^{N_{in}-n_a \text{ T}}]^\text{T}$ , where the first subvector is composed of the  $n_a$  active inequality constraints (i.e., those with  $\phi_{in,i} \geq 0$ ) and the second subvector of the remaining non-active inequality constraints (i.e., those with  $\phi_{in,i} < 0$ ).

Assuming that  $\phi_{in}^{n_a}(0) > \mathbf{0}$ , the goal of this proof is to show that convergence to point  $\phi_{in}^{n_a} = \mathbf{0}$  is achieved in finite time.

The column vector  $\dot{\phi}_{in}$  composed of the constraint function derivatives  $\dot{\phi}_{in,i}$  is given by

$$\dot{\phi}_{in} = \frac{\partial \phi^\text{T}}{\partial \mathbf{x}} \mathbf{f}(\mathbf{x}, \mathbf{d}) + \frac{\partial \phi^\text{T}}{\partial \mathbf{x}} \mathbf{g}(\mathbf{x}) \mathbf{u} = L_f \phi_{in} + \mathbf{L}_g \phi_{in} \mathbf{u}. \quad (15)$$

Premultiplying (15) by  $\text{v2m}(\text{pos}(\phi_{in}))$  and substituting (13) yields:

$$\text{v2m}(\mathbf{z}_{in}) \dot{\phi}_{in} = \text{v2m}(\mathbf{z}_{in}) L_f \phi_{in} - \mathbf{W}_{in} \mathbf{z}_{in} u_{in}^+, \quad (16)$$

where  $\mathbf{z}_{in}$  is a column vector with the  $i$ th-component  $z_{in,i} = 1$  if  $\phi_{in,i} > 0$  and  $z_{in,i} = 0$  if  $\phi_{in,i} < 0$ .

Let  $V_{in} = \mathbf{z}_{in}^\text{T} \text{v2m}(\mathbf{z}_{in}) \phi_{in}$  be a Lyapunov function candidate. Vector  $\phi_{in}^{n_a}$  can be generically partitioned into two subvectors  $\phi_{in}^{n_a} = [\phi_{in}^{b \text{ T}} \quad \phi_{in}^{n_a-b \text{ T}}]^\text{T}$ , where SM occurs in the manifold given by  $\phi_{in}^b = \mathbf{0}$ , whereas the components of vector  $\phi_{in}^{n_a-b}$  are greater than zero. Obviously, one of these two subvectors may be empty at a certain time. Since vectors  $\mathbf{z}_{in}^{n_a-b} = \mathbf{1}$  and  $\mathbf{z}_{in}^{N_{in}-n_a} = \mathbf{0}$  are constant, the time derivative of  $V$  results

in:

$$\begin{aligned} \dot{V}_{in} &= \frac{d \left( \begin{bmatrix} \mathbf{z}_{in}^b \\ \mathbf{1} \\ \mathbf{0} \end{bmatrix}^\top \text{v2m} \left( \begin{bmatrix} \mathbf{z}_{in}^b \\ \mathbf{1} \\ \mathbf{0} \end{bmatrix} \right) \right)}{dt} \begin{bmatrix} \mathbf{0} \\ \phi_{in}^{n_a-b} \\ \phi_{in}^{N_{in}-n_a} \end{bmatrix} + \mathbf{z}_{in}^\top \text{v2m}(\mathbf{z}_{in}) \dot{\phi}_{in} \\ &= \mathbf{z}_{in}^\top \text{v2m}(\mathbf{z}_{in}) \dot{\phi}_{in}. \end{aligned} \quad (17)$$

Substituting (16) in (17) yields:

$$\dot{V}_{in} = \mathbf{z}_{in}^\top \text{v2m}(\mathbf{z}_{in}) L_f \phi_{in} - \mathbf{z}_{in}^\top \mathbf{W}_{in} \mathbf{z}_{in} u_{in}^+. \quad (18)$$

Since  $\mathbf{z}_{in}^{N_{in}-n_a} = \mathbf{0}$  and the components of vector  $\mathbf{z}_{in}^{n_a}$  range from 0 to 1, the upper bound of the first term in (18) is given by  $z_{in,i}^{n_a} = 1$  when  $L_f \phi_{in,i}^{n_a} > 0$  and  $z_{in,i}^{n_a} = 0$  when  $L_f \phi_{in,i}^{n_a} < 0$ , that is:

$$\mathbf{z}_{in}^\top \text{v2m}(\mathbf{z}_{in}) L_f \phi_{in} \leq \sum_{i=1}^{n_a} (\max(L_f \phi_{in,i}, 0)). \quad (19)$$

Since  $u_{in}^+$  is a positive scalar and matrix  $\mathbf{W}_{in}$  is positive definite, the second term in (18) is negative and its upper bound is given by:

$$-\mathbf{z}_{in}^\top \mathbf{W}_{in} \mathbf{z}_{in} u_{in}^+ \leq -\text{diag}_{\min}(\mathbf{W}_{in}) \|\mathbf{z}_{in}\|_2^2 u_{in}^+, \quad \text{where } \|\mathbf{z}_{in}\|_2 \geq 1 \ \forall \ \phi_{in} > \mathbf{0}, \quad (20)$$

because if vector  $\phi_{in}^{n_a-b}$  is not empty at least one component of vector  $\mathbf{z}_{in}$  is equal to 1.

From (19) and (20), the upper bound of the time derivative of the Lyapunov function  $V$  results in:

$$\dot{V}_{in} \leq \sum_{i=1}^{n_a} (\max(L_f \phi_{in,i}, 0)) - \text{diag}_{\min}(\mathbf{W}_{in}) u_{in}^+. \quad (21)$$

Therefore, if  $u_{in}^+$  fulfills (14) the Lyapunov function decays at a finite rate, it vanishes and collective SM in the intersection of the  $n_a$  active inequality constraints occurs after a finite time interval.

□

### 3.3. Modified constraints

Approaching the constraints at high speed is not advisable because, in general, large joint accelerations  $\ddot{\mathbf{q}}$  would be required to slow down the robot motion in order to keep it on the constraint manifold. Therefore, the actual constraints  $\sigma_{eq,i}$  and  $\sigma_{in,i}$  will be modified to include the speed of movement as follows:

$$\phi_{eq,i} = \sigma_{eq,i} + K_{eq,i} \dot{\sigma}_{eq,i} = 0 \quad (22)$$

$$\phi_{in,i} = \sigma_{in,i} + K_{in,i} \dot{\sigma}_{in,i} \leq 0, \quad (23)$$

where  $K_{eq,i}$  and  $K_{in,i}$  are free design parameters that determine the rate of approach to the equality constraint manifold and to the boundary of the inequality constraint, respectively.

### 3.4. Chattering

Discrete-time implementations of the proposed SMC makes the system leave the ideal SM and oscillate with finite frequency and amplitude inside a band around  $\phi = \mathbf{0}$ , which is called *chattering* [7]. The upper bound for the chattering band  $\Delta\phi$  of the proposal can be obtained using the Euler-integration of the discontinuous control action given by Eq. (10), that is:

$$\Delta\phi = T_s |\mathbf{L}_g \phi \mathbf{u}| = T_s u^+ \text{diag}(\mathbf{W}), \quad (24)$$

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.

## 4. Proposed approach

### 4.1. Overview of method

The objective of this work is to obtain a robot control using the theory in Section 3 so that robot and human operator cooperatively perform a surface treatment task. For this purpose, some coordinates of the robot pose are controlled automatically to maintain the desired pressure between the robot tool and the object's surface as well as to assure that the tool is perpendicular to this surface by using a force sensor, namely *treatment sensor*, attached to the end of the robot tool. Meanwhile, other coordinates of the robot pose are controlled by the human operator in order to guide the tool along the object's surface using a second force sensor, namely *guide sensor*, attached to some place of the robot end-effector.

Fig. 2 shows the block diagram of the proposed method. Four tasks with different priority levels are considered. Level 1 (high-priority task) is developed using the SMC theories described in Section 3 in order to fulfill a set of *equality constraints* to properly accomplish the treatment on the surface with the robot tool using the treatment sensor. Level 2 (medium-high-priority task) is also developed using the SMC described in Section 3 in order to fulfill the *inequality constraint* required for the robot to track the human operator's forces using the guide sensor. Level 3 (medium-priority task) is used to reduce to zero the speed of the robot tool when no operator's forces are detected by the force guide sensor. Finally, Level 4 (low-priority task) is considered for the case of redundant robots to keep the robot close to the so-called *home* configuration.

The input to all four levels is the robot state  $\{\mathbf{q}, \dot{\mathbf{q}}\}$  obtained from the robot controller. Moreover, Level 1 and Level 2 have also as inputs the vectors  $\mathbf{F}_t$  and  $\mathbf{F}_g$ , respectively, of forces and torques measured by the treatment and guide sensors, which have already been filtered by the sensor electronics. Each level must satisfy an acceleration equality of the form  $\mathbf{A}_i \ddot{\mathbf{q}}_c = \mathbf{b}_i$  (5) whose square error must be minimized. For this purpose, the task priority redundancy resolution given by Eqs. (6) and (7) is used to obtain the commanded acceleration  $\ddot{\mathbf{q}}_{c,4}$ , which is integrated and sent to the robot controller. Finally, the robot controller performs a low-level control loop to track the commanded velocity  $\dot{\mathbf{q}}_c$  using the current angles  $\mathbf{q}$  and torques  $\boldsymbol{\tau}$  measured by the

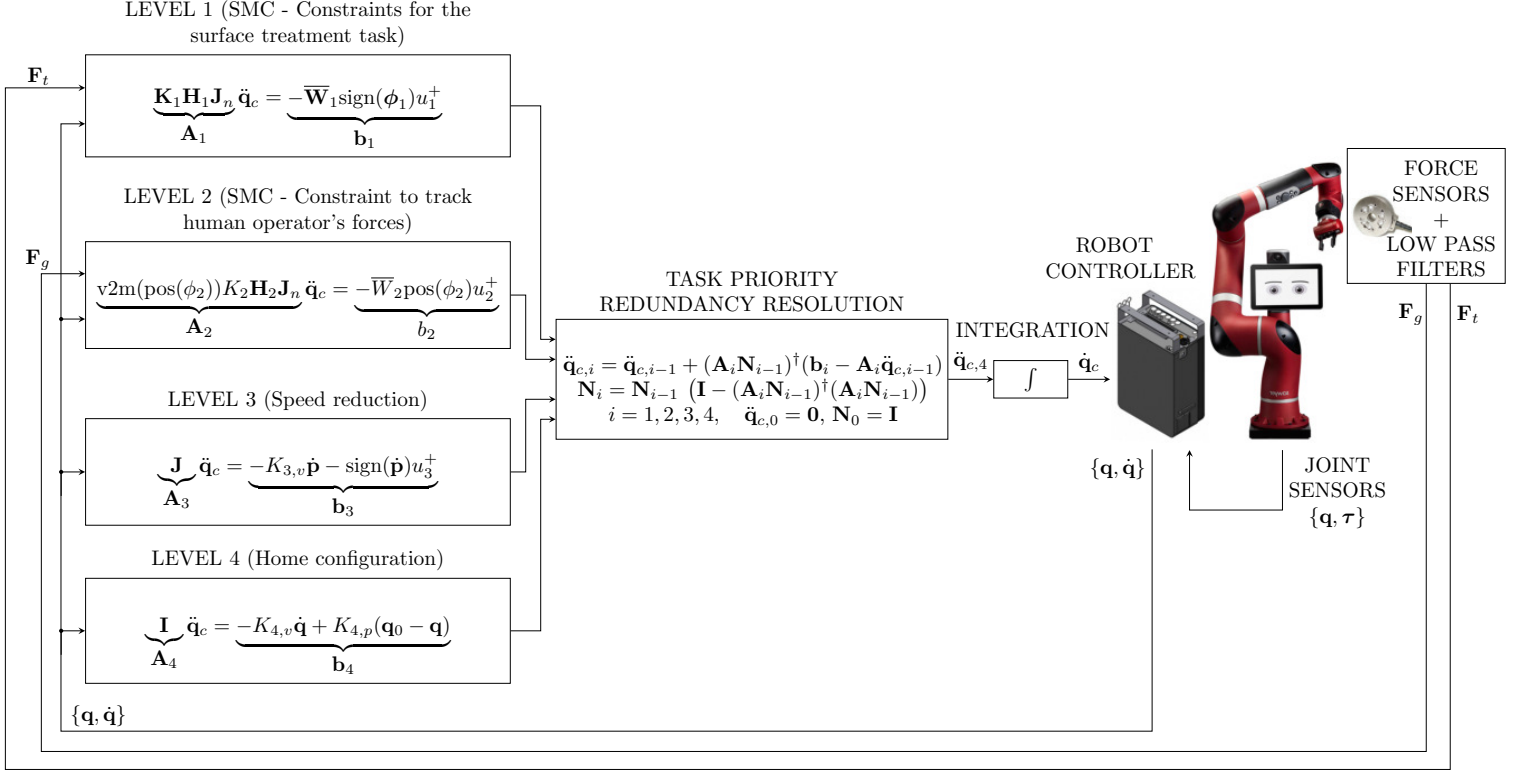


Figure 2. Block diagram of the method.

joint sensors.

#### 4.2. Lie derivatives

In order to use the theory in Section 3, a dynamical system in the form of Eq. (8) is considered with the state vector  $\mathbf{x} = [\mathbf{q}^T \quad \dot{\mathbf{q}}^T]^T$ , the disturbance vector  $\mathbf{d} = \mathbf{d}_c$  and the input vector  $\mathbf{u} = \ddot{\mathbf{q}}_c$ . Hence, the model is a double integrator, and from (4) the state equation results in:

$$\dot{\mathbf{x}} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{x} + \begin{bmatrix} \mathbf{0} \\ \mathbf{d}_c \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix} \mathbf{u}, \quad (25)$$



and, therefore, the Lie derivatives for the constraint function  $\phi_i$  are given by:

$$\mathbf{L}_g \phi_i = \nabla \phi_i^T \mathbf{g} = (\partial \phi_i / \partial \dot{\mathbf{q}})^T \quad (26)$$

$$L_f \phi_i = \nabla \phi_i^T \mathbf{f} = (\partial \phi_i / \partial \mathbf{q})^T \dot{\mathbf{q}} + (\partial \phi_i / \partial \dot{\mathbf{q}})^T \mathbf{d}_c. \quad (27)$$

### 4.3. Force model

Level 1 and Level 2 include the equality and inequality constraints required to properly accomplish the surface treatment and to guide the robot, respectively. These constraints 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 [31]:

$$\mathbf{F}(\mathbf{q}, t) = \mathbf{K}_s \Delta \mathbf{s}(\mathbf{q}, \mathbf{p}_s) = \begin{bmatrix} F_x & F_y & F_z & F_\alpha & F_\beta & F_\gamma \end{bmatrix}^T, \quad (28)$$

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. Note that, in general, both  $\mathbf{K}_s$  and  $\mathbf{p}_s$  are variable.

### 4.4. Level 1: Constraints for the surface treatment task

Three equality constraints are defined for the surface treatment as follows:

$$\sigma_{1,z}(\mathbf{F}_t) = \sigma_{1,z}(\mathbf{q}, t) = F_{t,z} - F_{z,ref} = 0 \quad (29)$$

$$\sigma_{1,\alpha}(\mathbf{F}_t) = \sigma_{1,\alpha}(\mathbf{q}, t) = F_{t,\alpha} = 0 \quad (30)$$

$$\sigma_{1,\beta}(\mathbf{F}_t) = \sigma_{1,\beta}(\mathbf{q}, t) = F_{t,\beta} = 0, \quad (31)$$

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 angular forces (i.e., 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 equality constraint is used to attain the desired force  $F_{z,ref}$  between the tool and the surface, whereas the last two equality constraints are used to keep the tool orientation perpendicular to the surface, since the torques in the tool  $X$ - and  $Y$ -axes are zero if the tool is perfectly perpendicular to the surface. Note that the torque in the  $Z$ -axis is not constrained and can be used for the specific treatment application: polishing, grinding, etc.

Taking into account (22), (26) and (28)–(31), the Lie derivative  $\mathbf{L}_g\phi_1$  required for the SMC in (10) is given by:

$$\begin{aligned}\mathbf{L}_g\phi_1 &= (\partial\phi_1/\partial\dot{\mathbf{q}})^T = \mathbf{K}_1(\partial\sigma_1/\partial\mathbf{q})^T \\ &= \mathbf{K}_1 \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \mathbf{K}_{t,s}\mathbf{J}_n = \mathbf{K}_1\mathbf{H}_1\mathbf{K}_{t,s}\mathbf{J}_n,\end{aligned}\quad (32)$$

where  $\sigma_1$  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 (22),  $\mathbf{K}_{t,s}$  is the stiffness diagonal matrix for the treatment sensor and  $\mathbf{J}_n$  is the geometric Jacobian relative to the tool coordinate system [31], 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  $\mathbf{L}_g\phi_1$  (32) 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:

$$\mathbf{K}_1\mathbf{H}_1\mathbf{J}_n\ddot{\mathbf{q}}_c = -\overline{\mathbf{W}}_1\text{sign}(\phi_1)u_1^+ \rightarrow \mathbf{A}_1\ddot{\mathbf{q}}_c = \mathbf{b}_1,\quad (33)$$

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

$$\overline{\mathbf{W}}_1 = \begin{bmatrix} W_{1,z}/K_{t,s,z} & 0 & 0 \\ 0 & W_{1,\alpha}/K_{t,s,\alpha} & 0 \\ 0 & 0 & W_{1,\beta}/K_{t,s,\beta} \end{bmatrix} = \begin{bmatrix} \overline{W}_{1,z} & 0 & 0 \\ 0 & \overline{W}_{1,\alpha} & 0 \\ 0 & 0 & \overline{W}_{1,\beta} \end{bmatrix}. \quad (34)$$

#### 4.5. Level 2: Constraint to track human operator's forces

The following inequality constraint is proposed to track the human operator's forces:

$$\sigma_2(\mathbf{F}_g) = \sigma_2(\mathbf{q}, t) = \sqrt{F_{g,x}^2 + F_{g,y}^2} - F_{l,0} = F_l - F_{l,0} \leq 0, \quad (35)$$

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 in the direction of the detected forces in order to fulfill the constraint. Note that the smaller is threshold  $F_{l,0}$  the less effort is for the human operator to move the robot tool.

Taking into account (23), (26) and (35), the Lie derivative  $\mathbf{L}_g\phi_2$  for the above constraint, which is required for the one-side SMC in (13), is given by:

$$\mathbf{L}_g\phi_2 = K_2(\partial\sigma_2/\partial\mathbf{q})^T = K_2 \begin{bmatrix} \overline{F}_{g,x} & \overline{F}_{g,y} & 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{K}_{g,s} \mathbf{J}_n = K_2 \mathbf{H}_2 \mathbf{K}_{g,s} \mathbf{J}_n, \quad (36)$$

where  $K_2$  is the approaching parameter to the original constraint (35), see (23),  $\overline{F}_{g,i} = F_{g,i}/F_l$  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:

$$\mathbf{v}2\mathbf{m}(\text{pos}(\phi_2)) \mathbf{L}_g\phi_2 \ddot{\mathbf{q}}_c = -W_2 \text{pos}(\phi_2) u_2^+, \quad (37)$$

where  $W_2$  and  $u_2^+$  are the switching gain weight and switching gain, respectively, for the second level (in this scalar case the switching gain weight could be omitted).

Since the stiffness coefficients  $\mathbf{K}_{g,s}$  in  $\mathbf{L}_g\phi_2$  (36) may not be known, they can be included in the switching gain weight  $W_2$ , so that the SMC given by (37) is modified as follows:

$$v2m(\text{pos}(\phi_2)) K_2 \mathbf{H}_2 \mathbf{J}_n \ddot{\mathbf{q}}_c = -\overline{W}_2 \text{pos}(\phi_2) u_2^+ \rightarrow \mathbf{A}_2 \ddot{\mathbf{q}}_c = b_2, \quad (38)$$

where  $\mathbf{A}_2$  and  $b_2$  are the values for the second task in (5) and:

$$\overline{W}_2 = W_2/K_{s,l}, \quad (39)$$

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 (38) is in the same direction as the human operator's forces detected by the guide sensor.

#### 4.6. Level 3: Speed reduction

The following equality is considered to reduce to zero the Cartesian speed of the robot tool:

$$\mathbf{J}\ddot{\mathbf{q}}_c = -K_{3,v}\dot{\mathbf{p}} - \text{sign}(\dot{\mathbf{p}})u_3^+ \rightarrow \mathbf{A}_3\ddot{\mathbf{q}}_c = \mathbf{b}_3, \quad (40)$$

where the Cartesian speed of the robot tool  $\dot{\mathbf{p}}$  is obtained from the first-order kinematics (2),  $K_{3,v}$  is the velocity correction gain of the continuous term in the above control law,  $u_3^+$  is the switching gain of the conventional SMC used in the second term, and  $\mathbf{A}_3$  and  $\mathbf{b}_3$  are the matrix and vector for the third task in (5). Basically, the above control law uses the first continuous term for speed reduction when  $\dot{\mathbf{p}} \neq \mathbf{0}$  and the second switching term to compensate the term  $\dot{\mathbf{J}}\dot{\mathbf{q}}$  of the robot second-order kinematics (3) when  $\dot{\mathbf{p}} = \mathbf{0}$ . This hybrid control law has several advantages: the time derivative of the robot Jacobian is not required; the switching gain  $u_3^+$  can be relatively small, reducing the chattering effects, while a fast speed convergence to zero is obtained due to the continuous correction term.

#### 4.7. Level 4: Home configuration

This level is considered only for the case of *redundant* robots (e.g., the one used in the experiments in Section 7) since otherwise there are no remaining degrees of freedom at this level. Among the different available options in literature, this work considers “pushing” the robot to a home configuration  $\mathbf{q}_0$  for increasing safety, in order to avoid a bias robot self-motion which may lead achieving critical areas due to, for instance, joint limits, singular configurations or possible obstacles in the robot workspace. To accomplish this purpose, the following equality is considered:

$$\ddot{\mathbf{q}}_c = -K_{4,v}\dot{\mathbf{q}} + K_{4,p}(\mathbf{q}_0 - \mathbf{q}) \rightarrow \mathbf{A}_4\ddot{\mathbf{q}}_c = \mathbf{b}_4, \quad (41)$$

where  $K_{4,v}$  and  $K_{4,p}$  are the gains used for the velocity and position corrections, respectively, and  $\mathbf{A}_4$  and  $\mathbf{b}_4$  are the matrix and vector for the fourth task in (5).

#### 4.8. Additional remarks

*Control action.* In this work the joint accelerations are 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 boundary of the constraints. 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 derivatives  $\dot{\mathbf{F}}_t$  and  $\dot{\mathbf{F}}_g$  for the SMC in Level 1 and Level 2, respectively. The simplest way to deal with this issue consists in using numerical differentiation, although some kind of filtering should be previously applied to the actual variable when non-negligible noise is present. However, the low-pass filter used for noise reduction must not limit the bandwidth of the control law.

#### 4.9. Guidelines to design the control parameters

The procedure to design the control parameters of the conventional and one-side SMCs in Level 1 and Level 2, respectively, is very similar. In particular, the steps to design

the control parameters of the proposed control method are as follows. Firstly, the cut-off frequency of the force sensor filters has to be selected to effectively remove the measurement noise. Then, the control period  $T_s$  must be selected guaranteeing that the SM frequency  $f_{SM} = (2T_s)^{-1}$  is lower than the bandwidths of the sensor filters and low-level joint controllers, otherwise changes in the SM control action would not be properly “followed”. Moreover, the bandwidth of the kinematic control performed in Level 1 (given by  $K_{1,i}$ ), Level 2 (given by  $K_2$ ), Level 3 (given by  $K_{3,v}$ ) and Level 4 (given by  $K_{4,p}$  and  $K_{4,v}$ ) should be significantly lower than the SM frequency  $f_{SM}$  for stability reasons. Note that the bandwidth of the controlled system corresponds to the bandwidth of the aforementioned kinematic controller. Forces  $F_{z,ref}$  and  $F_{l,0}$  can then be established depending on the requirements of the actual surface treatment task and the desired guiding sensitivity, respectively. Finally, the switching gains  $\{u_1^+, u_2^+, u_3^+\}$  and weights  $\{\bar{W}_{1,i}, \bar{W}_2\}$  can be empirically tuned to be as small as possible in order to reduce the chattering effect whilst guaranteeing that the SM behavior of the control action remains effective for the task at hand.

## 5. Evaluation of the proposed control method

In this section, the main features of the proposed conventional and one-side SMCs are illustrated by simulating a DC motor model, as it allows to better highlighting the merits of the proposed controller with a traditional example. Note that this model represents one joint of the robot system and can be easily extrapolated to a given number of joints, as considered in the real experimentation of Section 7. The simulation results presented in this section were obtained using MATLAB<sup>®</sup>.

The DC motor model is given by:

$$\dot{\omega} = (K_{m,t} i_m - b_m \omega) / J_m \quad (42)$$

$$\dot{i}_m = (-R_m i_m + V_m - K_{m,e} \omega) / L_m, \quad (43)$$

where  $\omega$  is the rotor angular velocity,  $i_m$  the armature current,  $V_m$  the voltage source,  $J_m$  the moment of inertia of the rotor,  $b_m$  the motor viscous friction constant,  $K_{m,e}$

the electromotive force constant,  $K_{m,t}$  the motor torque constant,  $R_m$  the electric resistance and  $L_m$  the electric inductance. For this model, the output of the system will be the rotor speed  $\omega$  and the input or control action the voltage  $V_m$ .

In order to track the reference velocity  $\omega_{ref}$  the equality and inequality constraints for the conventional and one-side SMC are defined in terms of the error  $e = \omega_{ref} - \omega$  as follows:

$$\sigma_{eq,m}(\omega) = e = 0, \quad \phi_{eq,m}(\omega, \dot{\omega}) = \sigma_{eq,m} + K_{eq,m} \dot{\sigma}_{eq,m} = 0 \quad (44)$$

$$\sigma_{in,m}(\omega) = |e| - e_{\max} \leq 0, \quad \phi_{in,m}(\omega, \dot{\omega}) = \sigma_{in,m} + K_{in,m} \dot{\sigma}_{in,m} \leq 0, \quad (45)$$

where  $K_{eq,m}$  and  $K_{in,m}$  are the approaching parameters to the original constraints and  $e_{\max}$  is the maximum allowed tracking error considered for the inequality constraint. Note that the relative degree between the constraint functions  $\{\phi_{eq,m}, \phi_{in,m}\}$  and the control action  $V_m$  is equal to one, as required by SM control theory.

Therefore, the Lie derivative  $L_g \phi$  and the control action  $V_m$  for the conventional and one-side SMCs are given by:

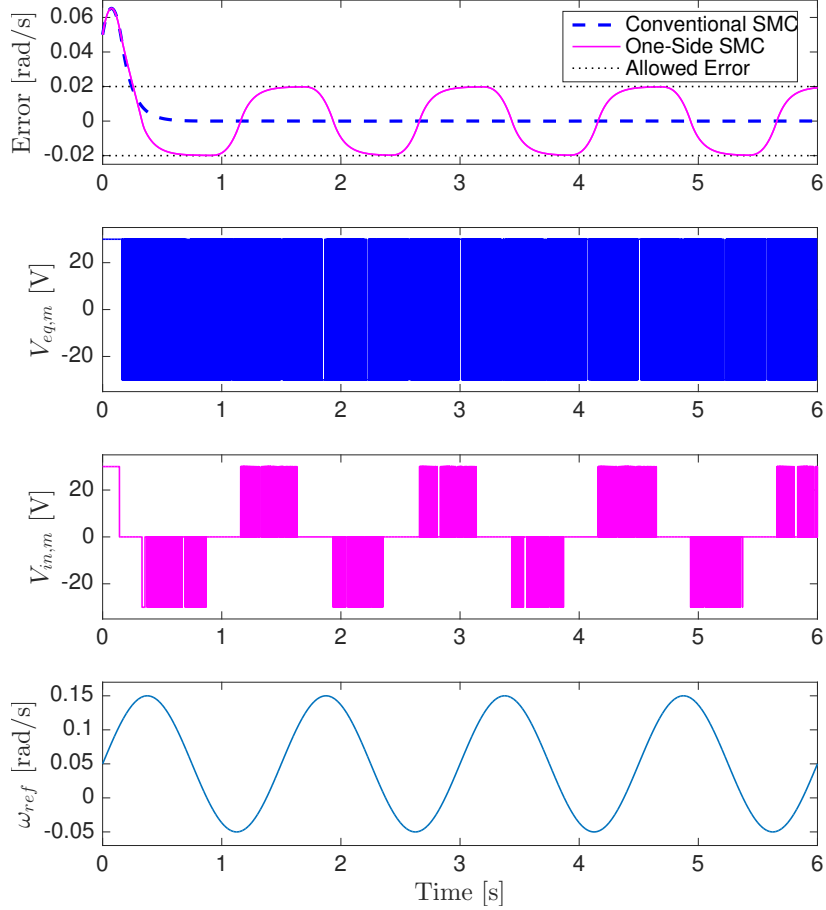
$$L_g \phi_{eq,m} = -(K_{eq,m} K_{m,t} / L_m), \quad V_{eq,m} = \text{sign}(\phi_{eq,m}) u^+ \quad (46)$$

$$L_g \phi_{in,m} = -(K_{in,m} K_{m,t} / L_m) \text{sign}(e), \quad V_{in,m} = \text{sign}(e) \text{pos}(\phi_{in,m}) u^+, \quad (47)$$

where the switching gain  $u^+$  includes all the constants of the control action.

The simulation was run under the following conditions:  $T_s = 1\text{ms}$ ,  $J_m = 0.1 \text{ kg.m}^2$ ,  $b_m = 0.1 \text{ N.m.s}$ ,  $K_{m,e} = 0.01 \text{ V.s}$ ,  $K_{m,t} = 0.01 \text{ N.m/A}$ ,  $R_m = 1 \text{ } \Omega$ ,  $L_m = 0.5 \text{ H}$ ,  $K_{eq,m} = K_{in,m} = 0.1 \text{ s}$ ,  $e_{\max} = 0.02 \text{ rad/s}$ ,  $u^+ = 30$  and the reference velocity is given by the sinusoidal wave  $\omega_{ref}(t) = 0.05 + 0.1 \sin(1.33\pi t) \text{ rad/s}$ .

Fig. 3 shows the simulation results for both conventional and one-side SMCs. It can be seen that, once the initial error has been corrected (reaching mode), the conventional SMC keeps the error signal at zero (top graph) switching the control action value from positive to negative and vice versa (second graph), which is known as SM. In contrast, the one-side SMC only applies a correction (third graph), i.e., the value of the control signal is not zero, when the error signal is about to leave the allowed region (top



**Figure 3.** Simulation of a DC motor control using conventional and proposed one-side SMC with a sinusoidal reference velocity. Top graph: errors. Second and third graphs: control action for the conventional and one-side SMC, respectively. Bottom graph: reference velocity.

graph) in order to keep the system within the limits. Therefore, it can be concluded that the proposed one-side SMC has a good performance and is especially suitable for non-negligible allowed regions, since otherwise the behavior is very similar to that obtained with conventional SMC. It is important to remark that when the one-side SMC applies no correction, the control signal can be used to achieve other goals of lower-priority tasks, see Section 2.

## 6. Controller implementation

The pseudo-code of the proposed method is shown below. The algorithm is executed at  $T_s$  seconds sampling time, and incorporates the following auxiliary functions:

- Kinematic function  $\mathbf{l}(\mathbf{q})$  and 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 [11].
- Robot sensors: *GetRobotStateAndForces*, which returns the current robot state  $\{\mathbf{q}, \dot{\mathbf{q}}\}$  and the current force vectors  $\mathbf{F}_t$  and  $\mathbf{F}_g$  measured by the treatment and guide sensors, respectively (it is assumed that the force signals have been already filtered by the sensor electronics).
- Actuators: *SendToJointControllers*( $\dot{\mathbf{q}}_c$ ), which sends the current commanded joint velocity vector to the joint controllers.

The computation time per iteration of the algorithm using compiled C code in a computer with Intel Core i5-3470 processor at 3.2 GHz clock frequency was around 15 microseconds for the experiment in Section 7.

## 7. Real experimentation

### 7.1. Setup

The setup used for the experiment consists of (see Fig. 4): a Sawyer cobot (collaborative robot); a guiding handle consisting of a T-shaped plastic rigid object composed of two prismatic portions of dimensions 148x30x6mm and 30x36x6mm; a force/torque sensor Nano25 (guide sensor) located between the last link of the robot and the guiding handle; another force/torque sensor Nano25 (treatment sensor) located between the guiding handle and the tool; a tool consisting of a cylinder of 43x43x10mm; and a flat rectangular plastic object of 190x95x3mm as target.

The controller is implemented in an external PC (Intel Core i5-3470 processor at 3.2GHz) using *Ubuntu 16.04* as O.S., *ROS Lunar* distribution, *Intera 5 SDK* from Rethink Robotics, and the *netft\_rdt\_driver* ROS package provided by ATI Industrial Automation. All Sawyer cobot, force sensor and external PC are connected to a router and communicate via UDP protocol.

---

Algorithm executed at sampling time of  $T_s$  seconds

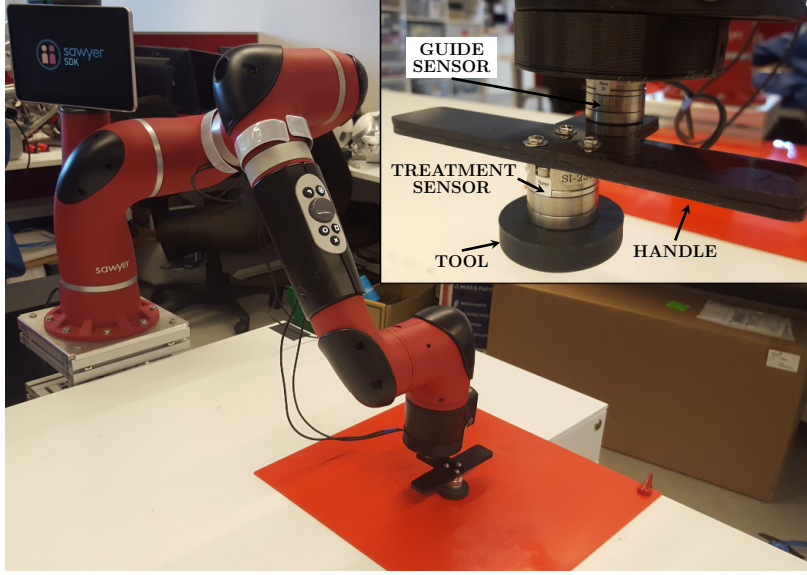
---

```

1  $[\mathbf{q}, \dot{\mathbf{q}}, \mathbf{F}_t, \mathbf{F}_g] = \text{GetRobotStateAndForces};$ 
2  $\mathbf{p} = \mathbf{l}(\mathbf{q});$  // Eq. (1)
3  $\dot{\mathbf{p}} = \mathbf{J}\dot{\mathbf{q}};$  // Eq. (2)
4  $\dot{\mathbf{F}}_t = (\mathbf{F}_t - \mathbf{F}_{t,prev})/T_s;$  // Derivative
5  $\dot{\mathbf{F}}_g = (\mathbf{F}_g - \mathbf{F}_{g,prev})/T_s;$  // Derivative
6  $\phi_1 = \begin{bmatrix} F_{t,z} - F_{z,ref} + K_{1,z}\dot{F}_{t,z} \\ F_{t,\alpha} + K_{1,\alpha}\dot{F}_{t,\alpha} \\ F_{t,\beta} + K_{1,\beta}\dot{F}_{t,\beta} \end{bmatrix};$  // Eqs. (22),(29)-(31)
7  $\phi_2 = \sqrt{F_{g,x}^2 + F_{g,y}^2} - F_{l,0} + K_2 \frac{\dot{F}_{g,x}F_{g,x} + \dot{F}_{g,y}F_{g,y}}{\sqrt{F_{g,x}^2 + F_{g,y}^2}};$  // Eqs. (23),(35)
8  $\mathbf{A}_1 = \mathbf{K}_1\mathbf{H}_1\mathbf{J}_n;$  // Eq. (33)
9  $\mathbf{b}_1 = -\bar{\mathbf{W}}_1 \text{sign}(\phi_1) u_1^+;$  // Eq. (33)
10  $\mathbf{A}_2 = v2m(\text{pos}(\phi_2)) K_2\mathbf{H}_2\mathbf{J}_n;$  // Eq. (38)
11  $\mathbf{b}_2 = -\bar{W}_2 \text{pos}(\phi_2) u_2^+;$  // Eq. (38)
12  $\mathbf{A}_3 = \mathbf{J};$  // Eq. (40)
13  $\mathbf{b}_3 = -K_{3,v}\dot{\mathbf{p}} - \text{sign}(\dot{\mathbf{p}})u_3^+;$  // Eq. (40)
14  $\mathbf{A}_4 = \mathbf{I};$  // Eq. (41)
15  $\mathbf{b}_4 = -K_{4,v}\dot{\mathbf{q}} + K_{4,p}(\mathbf{q}_0 - \mathbf{q});$  // Eq. (41)
16  $\ddot{\mathbf{q}}_{c,1} = \mathbf{A}_1^\dagger \mathbf{b}_1;$  // Eq. (6),  $i = 1$ 
17  $\mathbf{N}_1 = \mathbf{I} - \mathbf{A}_1^\dagger \mathbf{A}_1;$  // Eq. (7),  $i = 1$ 
18  $\ddot{\mathbf{q}}_{c,2} = \ddot{\mathbf{q}}_{c,1} + (\mathbf{A}_2\mathbf{N}_1)^\dagger (\mathbf{b}_2 - \mathbf{A}_2\ddot{\mathbf{q}}_{c,1});$  // Eq. (6),  $i = 2$ 
19  $\mathbf{N}_2 = \mathbf{N}_1(\mathbf{I} - (\mathbf{A}_2\mathbf{N}_1)^\dagger (\mathbf{A}_2\mathbf{N}_1));$  // Eq. (7),  $i = 2$ 
20  $\ddot{\mathbf{q}}_{c,3} = \ddot{\mathbf{q}}_{c,2} + (\mathbf{A}_3\mathbf{N}_2)^\dagger (\mathbf{b}_3 - \mathbf{A}_3\ddot{\mathbf{q}}_{c,2});$  // Eq. (6),  $i = 3$ 
21  $\mathbf{N}_3 = \mathbf{N}_2(\mathbf{I} - (\mathbf{A}_3\mathbf{N}_2)^\dagger (\mathbf{A}_3\mathbf{N}_2));$  // Eq. (7),  $i = 3$ 
22  $\ddot{\mathbf{q}}_{c,4} = \ddot{\mathbf{q}}_{c,3} + (\mathbf{A}_4\mathbf{N}_3)^\dagger (\mathbf{b}_4 - \mathbf{A}_4\ddot{\mathbf{q}}_{c,3});$  // Eq. (6),  $i = 4$ 
23  $\dot{\mathbf{q}}_c = \dot{\mathbf{q}}_{c,prev} + T_s \ddot{\mathbf{q}}_{c,4};$  // Integration
24  $\text{SendToJointControllers}(\dot{\mathbf{q}}_c);$ 
25  $[\dot{\mathbf{q}}_{c,prev}, \mathbf{F}_{t,prev}, \mathbf{F}_{g,prev}] = [\dot{\mathbf{q}}_c, \mathbf{F}_t, \mathbf{F}_g];$  // For next iteration

```

---



**Figure 4.** Experimental setup: 7R serial manipulator with two force sensors (guide and treatment sensors), a guiding handle consisting of a T-shaped plastic rigid object (black), a tool consisting of a cylinder (black) and a flat rectangular plastic object as target (red).

## 7.2. Experiment conditions

- i) 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.
- ii) The control period  $T_s$  is set to 20 milliseconds.
- iii) Parameters used for Level 1 (Section 4.4):  $F_{z,ref} = -10\text{N}$ ,  $u_1^+ = 0.06$ ,  $\bar{W}_{1,z} = 0.75$ ,  $\bar{W}_{1,\alpha} = \bar{W}_{1,\beta} = 12$ , and  $K_{1,i} = 0.15$ .
- iv) Parameters used for Level 2 (Section 4.5):  $F_{l,0} = 3\text{N}$ ,  $u_2^+ = 0.2$  and  $K_2 = 0.075$ .
- v) Parameters used for Level 3 (Section 4.6):  $K_{3,v} = 0.01$  and  $u_3^+ = 1.1$ .
- vi) Parameters used for Level 4 (Section 4.7):  $K_{4,v} = 4.5$ ,  $K_{4,p} = 0.75$  and  $\mathbf{q}_0 = \left[ 1.04^\circ \ -36.41^\circ \ -1.39^\circ \ 95.24^\circ \ 2.08^\circ \ 30.19^\circ \ 188.29^\circ \right]^T$ .

## 7.3. Results

In order to verify the performance of the proposed approach an experiment has been conducted, where the human operator moves the robot tool using the guiding handle while the target object to be treated either rests on a table or is held above the table. The video of this experiment can be played at <https://media.upv.es/player/?id=e7df1870-1371-11e8-9032-77826e6b3a7e>

and the results for this experiment are depicted at different figures as discussed below.

Fig. 5 shows several frames from the video: Fig. 5a and 5b (interval 19s–25s in the video of the experiment) show how the user guides the robot tool when the flat object (i.e., the target to be treated) is resting on the table; Fig. 5c and 5d (interval 1m27s–1m30s in the video of the experiment) show how the robot tool smoothly stops when the user leaves the guiding handle<sup>1</sup>; and Fig. 5e and 5f (interval 1m41s–1m49s in the video of the experiment) and Fig. 5g and 5h (interval 2m06s–2m14s in the video of the experiment) show how the user can guide the robot tool or leave it motionless, respectively, even when the flat object changes its position, orientation and stiffness, i.e., when the flat object is held above the table.

Fig. 6 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. This means that the surface treatment is being done properly: the tool orientation is perpendicular to the object surface and the pressure with the tool on the surface is being kept around the desired value regardless the changes of the object position, orientation and stiffness. Note that the amplitude of the chattering band for the equality constraint  $\phi_{1,z}$  is significantly reduced for the time interval 86s–164s in the graph (1m40s–2m58s in the video), which corresponds to the phase when the flat object is held above the table. This is because the stiffness coefficient  $K_{t,s,z}$  is significantly reduced when the flat object has no support and, since the used value for  $\overline{W}_{1,z}$  remains the same, it means that the actual value for  $W_{1,z}$  is also reduced, see (34). Thus, resulting in a lower chattering band, see (24), as can be clearly seen in the figure.

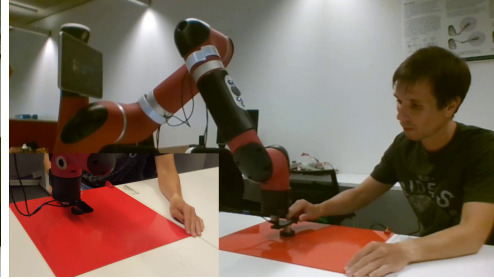
Fig. 6 also shows the behavior for the constraint in Level 2, which is used to guide the robot tool. In particular, it can be seen how the constraint of Level 2  $\phi_2$  is activated, see the fifth plot, when small force magnitudes are detected, see the fourth plot. Note that the main objective of Level 2 is to be very sensitive to the human operator’s forces in order to track its movements, which is successfully attained as shown in the video:

---

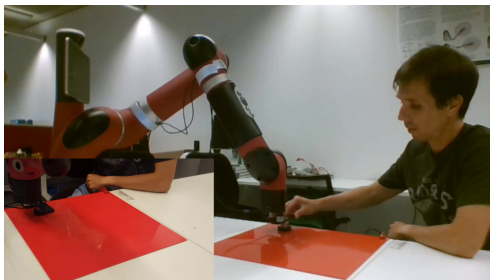
<sup>1</sup>A smooth stop has been considered for the robot tool, e.g., this is useful for textile industry where an abrupt stop is not advisable in order to avoid ruining the fabric. However, the desired speed reduction to stop the robot tool can be selected in Level 3.



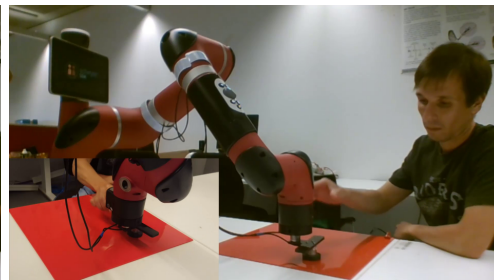
(a) video: 0m 19s; graph: 5s



(b) video: 0m 25s; graph: 11s



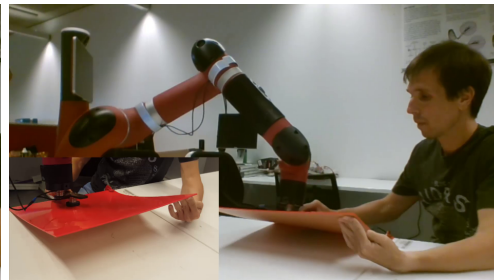
(c) video: 1m 27s; graph: 73s



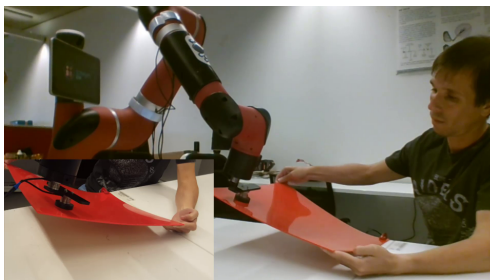
(d) video: 1m 30s; graph: 76s



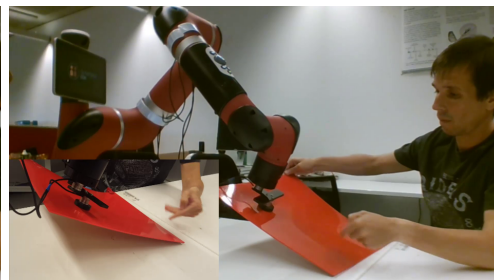
(e) video: 1m 41s; graph: 87s



(f) video: 1m 49s; graph: 95s

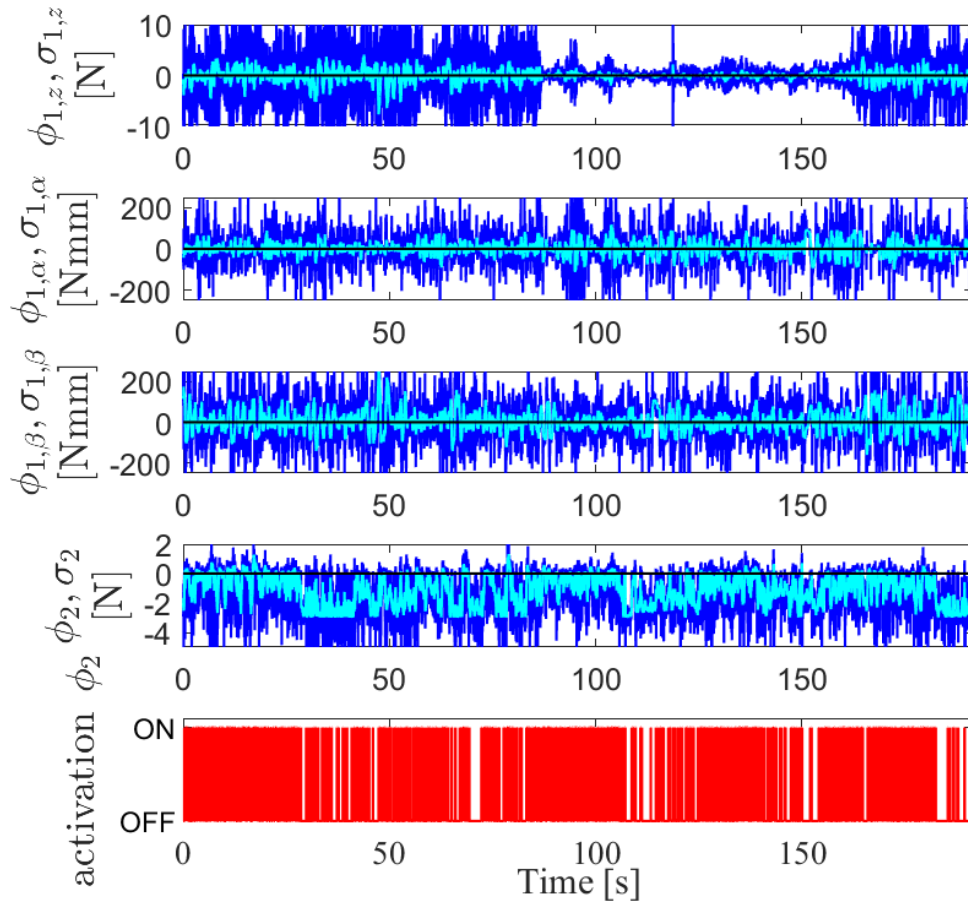


(g) video: 2m 06s; graph: 112s

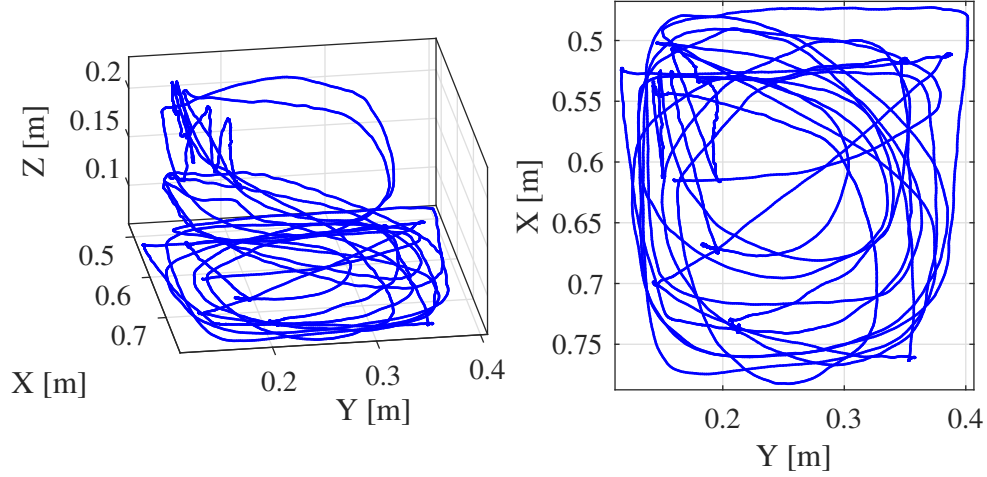


(h) video: 2m 14s; graph: 120s

**Figure 5.** Frames of the video of the experiment. The time instant is indicated for each frame.



**Figure 6.** Behavior of the constraints in the experiment. First to fourth plots: constraint functions for Level 1 and Level 2, the modified constraint function  $\phi_i$  is in dark-blue, whereas the original constraint function  $\sigma_i$  is in light-cyan. Fifth plot: activation of the inequality constraint in Level 2.

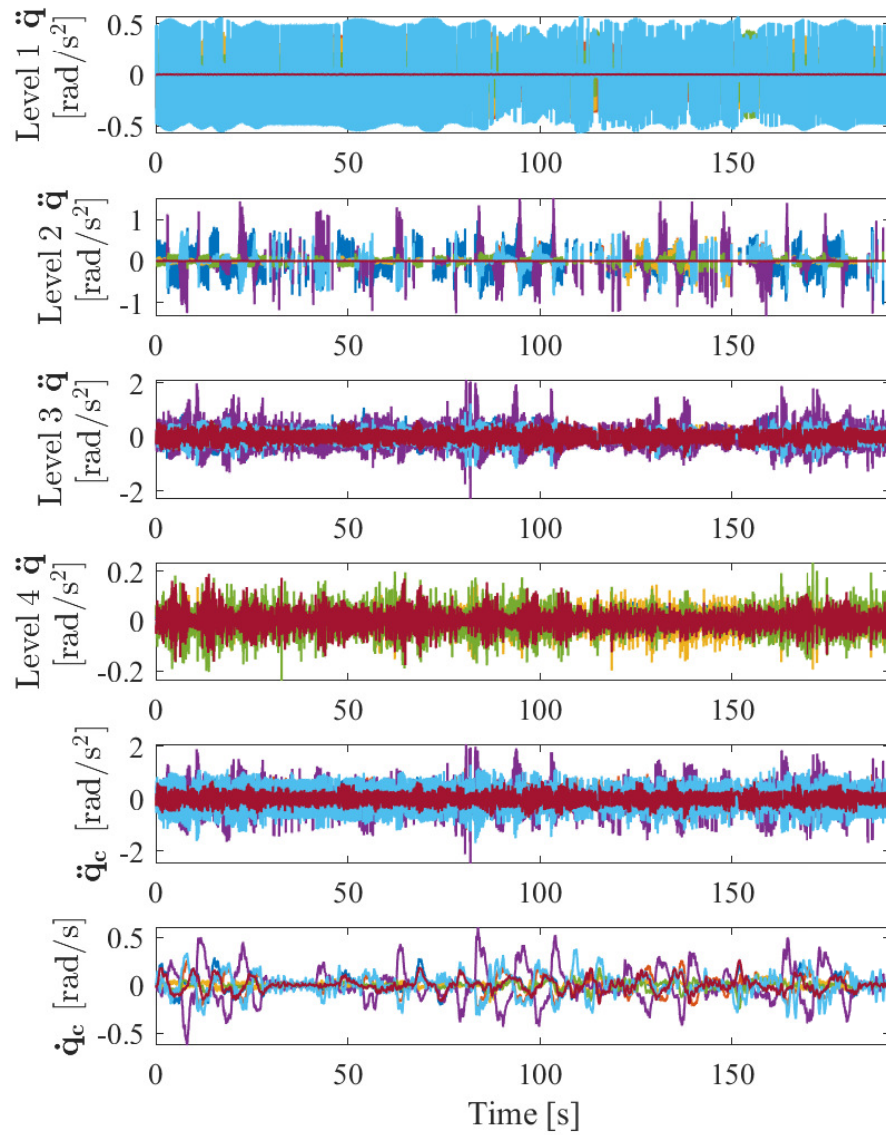


**Figure 7.** Trajectory followed by the robot tool in the experiment. Left: 3D view. Right: top view

the robot tool is guided by the human operator using just one finger and very small forces. However, it is interesting to remark that, if a low threshold  $F_{l,0}$  is considered to activate the inequality constraint used to guide the robot, the constraint could be accidentally activated by the weight supported by the guide sensor when the slope of the surface being treated is large, see the intervals 2m14s–2m17s and 2m39s–2m43s of the video. Obviously, this undesired movement can be avoided if the user simply handles the guiding element (in fact, the usual case is that the human operator guides the tool during the whole surface treatment task), see the intervals 2m17s–2m20s and 2m43s–2m46s of the video. In practice, to avoid gravity perturbation it is convenient to minimize the weight supported by the guide sensor, e.g., the sensor could be attached at a light handle of the guiding element that is used by the operator to move the robot.

Fig. 7 shows the trajectory followed by the robot tool, where it can be seen that a large area has been covered by the tool during the experiment: around 0.3 meters in both  $X$ - and  $Y$ -axes and 0.2 meters in  $Z$ -axis. Note that, although there are reference values for the forces in Level 1, there are no reference values for the tool trajectory in Level 2. In particular, the robot tool is guided by the human operator’s forces and, hence, the so-called “tracking errors” of the tool trajectory do not apply for the proposed human-robot collaborative solution for surface treatment tasks.

Fig. 8 shows the control commands computed during the experiment, where it can be seen that all four levels contribute to the commanded joint accelerations. Note that



**Figure 8.** Control actions for the experiment: contribution of each priority level to the commanded joint accelerations in the first four plots, the commanded joint accelerations in the fifth plot and commanded joint velocities to be sent to the robot controller in the bottom plot.



the control commands computed in Level 4 are small compared to those computed at the higher levels. However, Level 4 commands cannot be neglected since they allow to avoid a bias self-motion for the redundant robot which may lead achieving a critical region, e.g., the joint limits.

It is interesting to remark that robot behavior with the proposed SMC approach is relatively smooth. This is mainly due to two reasons. Firstly, the constraint modification introduced in Section 3.3 increases the order of the SM control action, which yields that the discontinuous control action is the joint acceleration vector and, hence, the joint velocities (and positions) are smooth, see Fig. 8. Secondly, the robot dynamics and low-level joint controllers smooth to a certain extent the commanded discontinuous control action.

It is worth noting that, in general, the hardness and stiffness of the target object affect the performance of the proposed method for surface treatment tasks and, hence, the parameters of the algorithm should be selected depending on these characteristics. In this sense, the proposed approach could benefit from adaptive methods for online self-tuning of the control parameters. For instance, adaptive switching gain (ASG) methods could be used to compute online the switching gain parameter  $u^+$  in order to improve the performance and adaptability of the control algorithm.

#### ***7.4. Chattering-free methods***

A new experiment has been conducted to evaluate several methods that reduce the chattering effect mentioned in Section 3.4. The experiment consists in tracking a circular reference trajectory with the flat target resting on the table and a fixed obstacle at the mid point of the circle. For this experiment Level 2 and Level 3 in Section 4.5 and Section 4.6, respectively, have been replaced by a conventional reference tracking controller in order to obtain repeatability in the experiments, i.e., the online guiding behavior is replaced by an a priori known reference trajectory. Moreover, the following inequality constraint is also included in Level 1 to adapt the tool position to the surface obstacles:  $\sigma_{1,xy} = \sqrt{F_{t,x}^2 + F_{t,y}^2} - F_{xy,\max} \leq 0$ , where  $F_{xy,\max} = 15$  N is the maximum allowed value for the linear force measured by the treatment sensor in the tool  $XY$  plane. This inequality constraint is fulfilled using the proposed one-side SMC. Details

omitted for brevity.

The following options available in the literature have been tested to reduce the chattering effect of the proposed SMC: hyperbolic tangent function [7] as commutation function, quasi-continuous SMC [17] and twisting and super-twisting SMCs [16]. Among all of them, the best results have been obtained using the super-twisting SMC. Fig. 9 shows a comparison experiment between the standard and the chattering-free SMCs. It can be seen in Fig. 9a that the tool trajectory is smoother for the chattering-free SMC, whereas Fig. 9b shows that the frequency harmonics of the chattering-free SMC are lower, particularly for the rotation  $\alpha$  in the  $X$ -axis, i.e., in the direction of the obstacle.

## 8. Conclusions

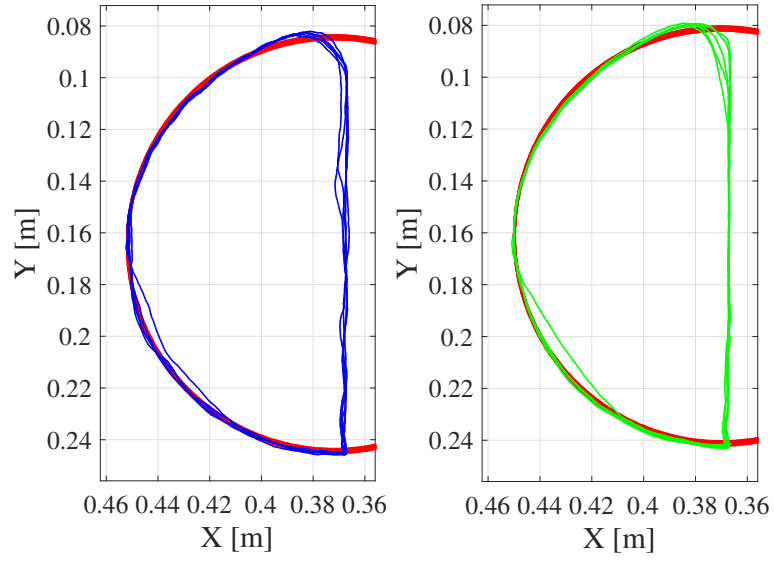
While solutions of robots working collaboratively alongside humans in polishing tasks exists, to the best of the authors' knowledge this is the first work that has proposed a human-robot closely collaborative solution to cooperatively perform surface treatment tasks such as polishing, grinding, finishing, deburring, etc. 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 force sensor is used by the human operator to guide the robot tool.

A distinctive feature of the proposed approach is that the controller was developed using not only conventional sliding mode control but also a non-conventional sliding mode control. The main advantages of the method are robustness and low computational cost.

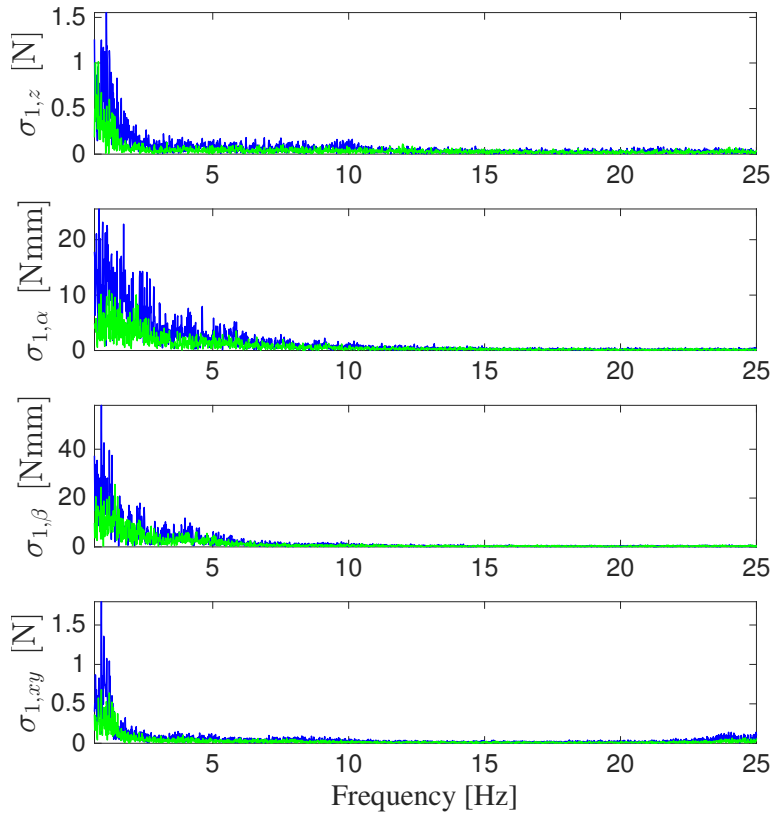
The effectiveness of the proposed approach has been shown experimentally using a redundant 7R manipulator: the Rethink Sawyer collaborative robot.

## Acknowledgements

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



(a) Left: standard SMC; Right: chattering-free SMC



(b) Constraint functions as a function of frequency

**Figure 9.** Comparison with chattering-free SMC. Top: trajectory followed by the tool (thin line) and circular reference (thick line). Bottom: constraint functions for the standard SMC (dark-blue) and chattering-free SMC (light-green).

## References

- [1] J. Angel-Fernandez and A. Bonarini. Robots showing emotions. *Interaction Studies*, 17(3):408–437, 2016.
- [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*, 45:306–321, 2017.
- [3] E. Bassi, F. Benzi, L. M. Capisani, D. Cuppone, and A. Ferrara. Hybrid position/force sliding mode control of a class of robotic manipulators. In *Proceedings of the 48th IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pages 2966–2971, 2009.
- [4] S. Chiaverini, G. Oriolo, and I. Walker. Kinematically redundant manipulators. *Springer Handbook of Robotics*, pages 245–268, 2008.
- [5] M. de Graaf, S. Allouch, and J. van Dijk. Long-term evaluation of a social robot in real homes. *Interaction Studies*, 17(3):461–490, 2016.
- [6] F. Dimeas and N. Aspragathos. Online stability in human-robot cooperation with admittance control. *IEEE Transactions on Haptics*, 9(2):267–278, 2016.
- [7] C. Edwards and S. Spurgeon. *Sliding Mode Control: Theory and Applications*. Taylor & Francis, UK, 1st edition, 1998.
- [8] H. Elbehriy, A. Hefnawy, and M. Elewa. Surface defects detection for ceramic tiles using image processing and morphological techniques. *International Journal of Computer and Information Engineering*, 1(5):1488–1492, 2007.
- [9] E. Engeberg, S. Meek, and M. Minor. Hybrid force-velocity sliding mode control of a prosthetic hand. *IEEE Transactions on Biomedical Engineering*, 55(5):1572–1581, 2008.
- [10] A. Etzioni and O. Etzioni. The ethics of robotic caregivers. *Interaction Studies*, 18(2):174–190, 2017.
- [11] G. Golub and C. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, MD, 3rd edition, 1996.

- [12] 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*, 7(7):1446 – 1453, 2013.
- [13] S. Jlassi, S. Tliba, and Y. Chitour. An event-controlled online trajectory generator based on the human-robot interaction force processing. *Industrial Robot: An International Journal*, 41(1):15–25, 2014.
- [14] N. Kashiri, N. G. Tsagarakis, M. Van Damme, B. Vanderborght, and D. G. Caldwell. *Proxy-Based Sliding Mode Control of Compliant Joint Manipulators*, pages 241–257. Springer International Publishing, Cham, 2016.
- [15] A. M. Khan, D.-w. Yun, K. M. Zuhaib, J. Iqbal, R.-J. Yan, F. Khan, and C. Han. Estimation of desired motion intention and compliance control for upper limb assist exoskeleton. *International Journal of Control, Automation and Systems*, 15(2):802–814, 2017.
- [16] A. Levant. Higher-order sliding modes, differentiation and output-feedback control. *Int. Journal of Control*, 76(9-10):924–941, 2003.
- [17] A. Levent. Quasi-continuous high-order sliding-mode controllers. *IEEE Transactions on Automatic Control*, 50(11):1812–1816, 2005.
- [18] Y. Li and S. S. Ge. Force tracking control for motion synchronization in human-robot collaboration. *Robotica*, 34(6):1260–1281, 2016.
- [19] S. S. Martínez, J. G. Ortega, J. G. García, A. S. García, and E. E. Estévez. An industrial vision system for surface quality inspection of transparent parts. *The International Journal of Advanced Manufacturing Technology*, 68(5):1123–1136, 2013.
- [20] A. T. Massoud, H. A. El Maraghy, and T. Lahdhiri. On the robust nonlinear motion position and force control of flexible joints robot manipulators. *Journal of Intelligent and Robotic Systems*, 25(3):227–254, 1999.
- [21] A. Mitra and L. Behera. Development of a fuzzy sliding mode controller with adaptive tuning technique for a mri guided robot in the human vasculature. In *2015 IEEE 13th International Conference on Industrial Informatics (INDIN)*, pages 370–377, 2015.

- [22] J. Molina, J. E. Solanes, L. Arnal, and J. Tornero. On the detection of defects on specular car body surfaces. *Robotics and Computer-Integrated Manufacturing*, 48:263 – 278, 2017.
- [23] Y. Nakamura, H. Hanafusa, and T. Yoshikawa. Task-priority based redundancy control of robot manipulators. *The Int. Journal of Robotics Research*, 6(2):3–15, 1987.
- [24] G. Orta, A. S. Bilgi, K. Tasdemir, and H. Kalkan. A hyperspectral imaging based control system for quality assessment of dried figs. *Computers and Electronics in Agriculture*, 130:38 – 47, 2016.
- [25] F. Papadopoulos, D. Kuster, L. Corrigan, A. Kappas, and G. Castellano. Do relative positions and proxemics affect the engagement in a human-robot collaborative scenario? *Interaction Studies*, 17(3):321–347, 2016.
- [26] N. Rahman and M. C. Lee. Reaction force separation method of surgical tool from unknown dynamics and disturbances by fuzzy logic and perturbation observer of smcspo algorithm. In *The SICE Annual Conference 2013*, pages 2536–2541, 2013.
- [27] A. Roswell, F. J. Xi, and G. Liu. Modelling and analysis of contact stress for automated polishing. *International Journal of Machine Tools and Manufacture*, 46(3):424 – 435, 2006.
- [28] S. Sakaino and K. Ohnishi. Sliding mode control based on position control for contact motion applied to hopping robot. In *2006 IEEE International Conference on Industrial Technology*, pages 170–175, 2006.
- [29] Y. Shi, D. Zheng, L. Hu, Y. Wang, and L. Wang. Nc polishing of aspheric surfaces under control of constant pressure using a magnetorheological torque servo. *The International Journal of Advanced Manufacturing Technology*, 58(9):1061–1073, 2012.
- [30] 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)*, pages 1211–1216, 1991, Pisa, Italy, 1991.
- [31] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo. *Robotics: Modelling, Planning and Control*. Springer-Verlag, London, UK, 2009.
- [32] 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*, 87(1):639–646, 2016.
- [33] J. Tornero, L. Armesto, M. C. Mora, N. Montés, A. Herráez, and J. Asensio. Detección de defectos en carrocerías de vehículos basado en visión artificial: Diseño e implantación. *Revista Iberoamericana de Automática e Informática Industrial RIAI*, 9(1):93–104, 2012.
- [34] V. Utkin, J. Guldner, and J. Shi. *Sliding Mode Control in Electro-Mechanical Systems*. Taylor & Francis, London, 2nd edition, 2009.
- [35] E. Vlachos, E. Jochum, and L.-P. Demers. The effects of exposure to different social robots on attitudes toward preferences. *Interaction Studies*, 17(3):390–404, 2016.
- [36] 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*, 34(6):763–780, 2015.
- [37] Q. Wu, X. Wang, F. Du, and Q. Zhu. Fuzzy sliding mode control of an upper limb exoskeleton for robot-assisted rehabilitation. In *2015 IEEE International Symposium on Medical Measurements and Applications (MeMeA) Proceedings*, pages 451–456, 2015.
- [38] D. Yun, A. M. Khan, R.-J. Yan, Y. Ji, H. Jang, J. Iqbal, K. M. Zuhaib, J. Y. Ahn, J. Han, and C. Han. Handling subject arm uncertainties for upper limb rehabilitation robot using robust sliding mode control. *International Journal of Precision Engineering and Manufacturing*, 17(3):355–362, 2016.
- [39] J. Zhou, Z. Zhou, and Q. Ai. Impedance control of the rehabilitation robot based on sliding mode control. In X. Li, editor, *Mechanical Engineering and Control Systems (MECS2015)*, pages 135–140, 2016.

## *Authors' addresses*

Luis Gracia

Department of Systems Engineering and Control

Universitat Politècnica de València

Camino de Vera s/n

46022 Valencia, Spain

luigraca@isa.upv.es

J. Ernesto Solanes

Design and Manufacturing Institute

Universitat Politècnica de València

Camino de Vera s/n

46022 Valencia, Spain

esolanes@idf.upv.es

Pau Muñoz-Benavent

Design and Manufacturing Institute

Universitat Politècnica de València

Camino de Vera s/n

46022 Valencia, Spain

pmunyo@disca.upv.es

Jaime Valls Miro

Centre for Autonomous Systems

University of Technology Sydney

NSW 2007 Sydney, Australia



Jaime.VallsMiro@uts.edu.au

Carlos Perez-Vidal

Departamento de Ingeniería de Sistemas y Automática

Universidad Miguel Hernández

Avda de la Universidad s/n

03202 Elche, Spain

carlos.perez@umh.es

Josep Tornero

Department of Systems Engineering and Control

Universitat Politècnica de València

Camino de Vera s/n

46022 Valencia, Spain

jtornero@isa.upv.es

### *Biographical notes*

**Luis Gracia** received the B.Sc. degree in electronic engineering, the M.Sc. degree in control systems engineering, and the Ph.D. in automation and industrial computer science from the Technical University of Valencia (UPV), Spain, in 1998, 2000, and 2006, respectively. He is currently Associate Professor at the Department of Systems Engineering and Control (DISA) of the UPV. His research interests include mobile robots, robotic manipulators, and system modeling and control.

**J. Ernesto Solanes** received his B.S. Degree in Industrial Electronics Engineering

(2007), B.S. Degree in Industrial Automatics (2009), M.S. Degree in Automatics and Industrial Informatics (2011) and the Ph.D. in Robotics, Automatics and Industrial Informatics (2015), all of them at the Technical University of Valencia (UPV), Spain. He is currently assistant professor at the Department of Systems Engineering and Control (DISA) of the UPV. His research interests include nonlinear and robust control, computer vision and robotics.

**Pau Muñoz-Benavent** received the B.Sc. degree in industrial engineering, the M.Sc. degree in control systems engineering and the Ph.D. in automation and industrial computer science from the Technical University of Valencia (UPV), Spain, in 2008, 2010 and 2017, respectively. He is currently researcher at the Design and Manufacturing Institute (IDF) of the UPV. His research interests include visual servoing, computer vision systems and robotic systems.

**Jaime Valls Miro** received the B.Eng. and M.Eng. degrees in Computer Science (Systems Engineering) from the Technical University of Valencia (UPV), Valencia (Spain), in 1990 and 1993 respectively. He received his Ph.D. degree from Middlesex University, London (UK) in 1998. His thesis examined the use of full dynamics for trajectory planning and optimal control of industrial manipulators. He worked for 5 years as a software and control systems analyst for a London-based company designing ROVs (remotely operated underwater vehicles). Currently, he is Associate Professor at the Centre for Autonomous Systems of the University of Technology Sydney (Australia).

**Carlos Perez-Vidal** is Associate Professor of Control and Systems Engineering at the Miguel Hernandez University (UMH), Elche (Spain), and researcher at the Systems Engineering and Automatic Control Division. He holds a B.Sc. in Industrial Engineering (1998), M.Sc. in Control Engineering (2000) and Ph.D. in Industrial Technologies (2008) from the Technical University of Valencia and UMH, Spain. His current research interests are robotics, visual servoing, automation/control and medical applications.

**Josep Tornero** received the MS Degree in Systems and Control from the University of Manchester in 1982, and the PhD in Electrical Engineering at the Technical University of Valencia (UPV) in 1985. He has been Visiting Professor at: the CIRSSE

(NASA Center for Intelligent Robotics Systems for Space Exploration); the Rensselaer Polytechnic Institute at Troy (New York); and at the Department of Mechanical Engineering at the University of California (Berkeley). He is Professor at the DISA of the UPV and responsible for the Design and Manufacturing Institute (IDF) at the UPV. He is interested in control of robot systems and multirate sampled data systems.