J. Ernesto Solanes[1]
Instituto de Diseño y Fabricación,
Universitat Politècnica de València,
Camí de Vera s/n,
València 46022, Spain,
e-mail: esolanes@idf.upv.es

Luis Gracia
Instituto de Diseño y Fabricación,
Universitat Politècnica de València,
Camí de Vera s/n,
València 46022, Spain
e-mail: luigraca@isa.upv.es

Pau Muñoz-Benavent
Instituto de Diseño y Fabricación,
Universitat Politècnica de València,
Camí de Vera s/n,
València 46022, Spain
e-mail: pmunyoz@disca.upv.es

Jaime Valls Miro
Centre for Autonomous Systems,
University of Technology Sydney,
Sydney 2007, NSW, Australia
e-mail: jaime.vallsmiro@uts.edu.au

Carlos Perez-Vidal
Ingeniería de Sistemas y Automática,
Universidad Miguel Hernández,
Avda de la Universidad s/n,
Elche 03202, Spain
e-mail: carlos.perez@umh.es

Josep Tornero
Instituto de Diseño y Fabricación,
Universitat Politècnica de València,
Camí de Vera s/n,
València 46022, Spain
e-mail: jtornero@isa.upv.es

Robust Hybrid Position-Force Control for Robotic Surface Polishing

This work presents a hybrid position-force control of robots for surface polishing using task priority. The robot force control is designed using sliding mode ideas in order to benefit from its inherent robustness and low computational cost. In order to avoid the chattering drawback typically present in sliding mode control, several chattering-free controllers are evaluated and tested. A distinctive feature of the method is that the sliding mode force task is defined using not only equality constraints but also inequality constraints, which are satisfied using conventional and nonconventional sliding mode control, respectively. Moreover, a lower priority tracking controller is defined to follow the desired reference trajectory on the surface being polished. The applicability and the effectiveness of the proposed approach considering the mentioned chattering-free controllers are substantiated by experimental results using a redundant 7R manipulator.
[DOI: 10.1115/1.4041836]

1 Introduction

Continuous advances in the manufacturing sector imply that processes are progressively being partially or fully automated. The inclusion of flexible machinery such as industrial manipulators together with advanced sensing capabilities (cameras, force transducers, etc.) has allowed step improvements in terms of cycle times, operator safety, and comfort, as well as in the quality of the end product over more traditional hand-made methods.

One of the least automated processes is the quality control of surfaces [1]. This is primarily attributed to the fact that automated processes remain elusive in meeting strict requirements when it comes to short cycle time, low cost, and the high quality achieved in other manufacturing industries [2,3]. As such, surface treatment operations and quality control continue to be mainly a manual process being carried out by skilled workers, which give rise to issues such as subjectivity in the evaluation criteria and human errors.

It is the view behind the contribution in this paper that the task of automatically eliminating product anomalies from a given surface can be handled automatically by incorporating robotic systems equipped with the appropriate sensing and intelligent controls. Since the robot tool has to be in contact with the product surface to apply a specific treatment (e.g., polishing, deburring, roughing, and grinding), both the tool position and the exerted forces have to be moderated. Moreover, the tool must be kept perpendicular to the surface at all times to homogenize the pressure on all contact points [4]. Many approaches can be found in the literature tackling this problem using robot manipulators with force feedback, e.g., see Refs. [5] and [6], among others. Other robot force control approaches are based on sliding-mode control (SMC) theory [7,8].

This paper addresses the problem of robot surface treatment using a hybrid position-force approach: force control is performed using SMC in order to benefit from its inherent robustness and low computational cost, whereas position control is performed using a conventional continuous tracking controller. The proposal presents several distinctive features and innovations:

(i) Instead of the classical compliance vector [9], this work uses a task priority scheme to combine the force and position controllers. The advantage of this approach is that it allows the inclusion of other objectives besides tracking reference forces and positions.
(ii) Beyond the more traditional equality constraints being considered in SMC to undertake surface treatment [10]

[1]Corresponding author.
Manuscript received November 14, 2017; final manuscript received October 24, 2018; published online November 28, 2018. Assoc. Editor: Dragan Djurdjanovic.

(i.e., to keep the desired tool pressure on the surface and to keep the tool perpendicular to the surface), an inequality constraint is also incorporated. This inequality constraint becomes active when the robot tool makes contact with obstacles or protrusions on the surface and the sensed forces then exceed user-specified thresholds. Thus, the fulfillment of this inequality constraint allows to compliant the tool position to the obstacles that may be encountered on the surface during the treatment.

(iii) The inclusion of inequality constraints in mathematical and engineering problems is nothing new. For instance, optimization problems typically maximize or minimize an index subject to both equality and inequality constraints. Recently, this type of problems has been solved in the literature using the so-called barrier functions, see Refs. [11] and [12], among others. In this sense, the equality and inequality constraints proposed in this work could be easily solved analytically if a complete and accurate process model is available. However, in this work, it is proposed to use conventional SMC and a novel nonconventional SMC to fulfill these constraints in order to benefit from their inherent robustness and low computational cost, as mentioned earlier.

(iv) One typical problem of conventional SMC is the chattering drawback. Since this phenomenon depends directly on the sampling frequency, one way to alleviate it is to increase the system sampling rate in order to obtain smoother control actions. However, on the one hand, industrial manipulators tend to operate at relatively low control rates, typically over 1 ms, and on the other hand, sensors such as force/torque transducers have significant signal noise which is exacerbated at high sampling rates. Due to these facts, a number of alternatives have been proposed in the literature to obtain a so-called chattering-free SMC without the need to increase the system sampling rate. These approaches are typically based on softening the discontinuous control law or using a dynamical or high-order SMC [13,14]. Therefore, several chattering-free alternatives are evaluated and comparatively tested for the robot task tackled in this work not only for the conventional SMC but also for the nonconventional SMC in order to alleviate the chattering drawback.

It is worth noting that force control tasks have been typically tackled in robotics using the classical impedance/admittance control, see Refs. [15] and [16], among others. These approaches transform the force values measured by the sensor attached to the robot end-effector to a continuous velocity signal that is commanded to the joint controllers of the robot in order to move the robot end-effector. The approach proposed in this work is similar, except that instead of a continuous velocity signal, a discontinuous acceleration signal is considered.

The paper is organized as follows: Sec. 2 introduces some preliminaries, while Sec. 3 presents the conventional and nonconventional SMCs used in this work. The proposed approach is presented in Sec. 4, while its feasibility and effectiveness are substantiated by experimental results in Sec. 5 using a redundant 7R manipulator. Finally, some conclusions are given.

2 Preliminaries

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

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

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

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

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

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

$$\ddot{\mathbf{q}} = \ddot{\mathbf{q}}_c + \mathbf{d}_c \tag{4}$$

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 [17] 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:

$$\mathbf{A}_i\ddot{\mathbf{q}}_c = \mathbf{b}_i, \quad i = 1, \ldots, M \tag{5}$$

where matrix $\mathbf{A}_i$ and vector $\mathbf{b}_i$ of the ith 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 Eq. (5) is given by [18]

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

$$\mathbf{N}_i = \mathbf{N}_{i-1}(\mathbf{I} - (\mathbf{A}_i\mathbf{N}_{i-1})^{\dagger}(\mathbf{A}_i\mathbf{N}_{i-1})), \quad \text{with } i = 1, \ldots, M,\ \ddot{\mathbf{q}}_{c,0} = 0 \text{ and } \mathbf{N}_0 = \mathbf{I} \tag{7}$$

where $\mathbf{I}$ and $\mathbf{0}$ denote the identity matrix and the zero column vector, respectively, of suitable size, superscript † 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

This section presents the SMC algorithms considered in this work: on the one hand, a conventional SMC is used to satisfy equality constraints and, on the other hand, a nonconventional SMC is proposed to satisfy inequality constraints. Figure 1 shows a graphical two-dimensional example to illustrate both approaches.

3.1 Conventional Sliding-Mode Control 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* [19]. 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) [19]. A *continuous equivalent control* [20] can be obtained for the SM phase, i.e., the control required to keep the system on the sliding surface. Hence, SMC produces such control action without explicitly computing it and with low computational cost, which is a typical advantage of SMC strategies [20].

The theorem below presents a conventional SMC designed to satisfy equality constraints, expanded here for completeness:

THEOREM 3.1. *Consider the following dynamical system with n_x states and n_u inputs given by*

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

where $\mathbf{x}(t)$ *is the state vector,* $\mathbf{d}(t)$ *is an unmeasured disturbance or model uncertainty,* $\mathbf{u}(t)$ *is the control input vector (possibly discontinuous),* $\mathbf{f}$ *is the so-called drift vector field, and* $\mathbf{g} = [\mathbf{g}_1 \ldots \mathbf{g}_{n_u}]$*, where* $\mathbf{g}_i$ *is a control vector field.*

Consider also that the system state vector $\mathbf{x}$ *is subject to equality constraints* $\phi_{\text{eq},i}(\mathbf{x}) = 0$, $i = 1, \ldots, N_{\text{eq}}$*, where* $\phi_{\text{eq},i}(\mathbf{x})$ *is the ith equality constraint function. Thus, the region* Φ_{eq} *of the state space compatible with the constraints on state* $\mathbf{x}$ *is given by*

$$\Phi_{\text{eq}} = \{\mathbf{x} | \phi_{\text{eq},i}(\mathbf{x}) = 0\} \tag{9}$$

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

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

$$\mathbf{L}_{\mathbf{g}}\boldsymbol{\phi}_{\text{eq}}\mathbf{u} = -\mathbf{W}_{\text{eq}}\text{sign}(\boldsymbol{\phi}_{\text{eq}})\,u_{\text{eq}}^{+} \tag{10}$$

$$u_{\text{eq}}^{+} > \|L_f\boldsymbol{\phi}_{\text{eq}}\|_1/\text{diag}_{\min}(\mathbf{W}_{\text{eq}}) \tag{11}$$

where $\boldsymbol{\phi}_{\text{eq}}$ *is a column vector with all the constraint functions* $\phi_{\text{eq},i}$*, the scalar* $L_f\phi_{\text{eq},i} = (\partial\phi_{\text{eq},i}^{\text{T}}/\partial\mathbf{x})\mathbf{f}$ *and the row vector* $\mathbf{L}_{\mathbf{g}}\phi_{\text{eq},i} = (\partial\phi_{\text{eq},i}^{\text{T}}/\partial\mathbf{x})\mathbf{g}$ *denote the Lie derivatives of* $\phi_{\text{eq},i}(\mathbf{x})$ *in the direction of* $\mathbf{f}$ *and in the direction of* $\mathbf{g}$*, respectively, column vector* $L_f\boldsymbol{\phi}_{\text{eq}}$ *contains the elements* $L_f\phi_{\text{eq},i}$ *of all equality constraints, matrix* $\mathbf{L}_{\mathbf{g}}\boldsymbol{\phi}_{\text{eq}}$ *contains the row vectors* $\mathbf{L}_{\mathbf{g}}\phi_{\text{eq},i}$ *of all equality constraints, sign(·) 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,* $\mathbf{W}_{\text{eq}}$ *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 Proof 2.1 in Ref. [20] and its generalization. Details omitted for brevity.

3.2 Nonconventional Sliding-Mode Control to Satisfy Inequality Constraints. The nonconventional 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 **A** and the allowed region **B**, which are separated by the constraint boundary. Similar to conventional SMC, when the system state is in the nonallowed region **A,** the control action $\mathbf{u} = \mathbf{u}_B$ pushes the system into the allowed region **B**. But in contrast to conventional SMC, when the system state is in the allowed region **B**, no control action is applied, i.e., $\mathbf{u} = \mathbf{0}$. Hence, if the system starts in region **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 **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 nonconventional 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 [19].

The theorem below presents the proposed nonconventional SMC in order to satisfy inequality constraints:

Theorem 3.2. *Consider the dynamical system given by Eq. (8) and consider also that the system state vector* $\mathbf{x}$ *is subject to inequality constraints* $\phi_{\text{in},i}(\mathbf{x}) \leq 0$, $i = 1, \ldots, N_{\text{in}}$*, where* $\phi_{\text{in},i}(\mathbf{x})$ *is the ith inequality constraint function. Thus, the region* Φ_{in} *of the state space compatible with the constraints on state* $\mathbf{x}$ *is given by*

$$\Phi_{\text{in}} = \{\mathbf{x} | \phi_{\text{in},i}(\mathbf{x}) \leq 0\} \tag{12}$$

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

Then, assuming that the constraint functions $\phi_{\text{in},i}$ *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:*

$$\text{v2dm}(\text{pos}(\boldsymbol{\phi}_{\text{in}}))\mathbf{L}_{\mathbf{g}}\boldsymbol{\phi}_{\text{in}}\mathbf{u} = -\mathbf{W}_{\text{in}}\,\text{pos}(\boldsymbol{\phi}_{\text{in}})\,u_{\text{in}}^{+} \tag{13}$$

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

where function $\text{v2dm}(\cdot)$ *converts a vector into a diagonal matrix, function pos(·) represents the positive function (i.e., pos(x) is equal to 0 if* $x < 0$ *and equal to 1 if* $x > 0$*),* ϕ_{in} *is a column vector with all the inequality constraint functions* $\phi_{\text{in},i}$*, matrix* $\mathbf{L}_{\mathbf{g}}\boldsymbol{\phi}_{\text{in}}$ *contains the row vectors* $\mathbf{L}_{\mathbf{g}}\phi_{\text{in},i}$ *of all inequality constraints, the scalar* $L_f\phi_{\text{in},i} = (\partial\phi_{\text{in},i}^{\text{T}}/\partial\mathbf{x})\mathbf{f}$ *and the row vector* $\mathbf{L}_{\mathbf{g}}\phi_{\text{in},i} = (\partial\phi_{\text{in},i}^{\text{T}}/\partial\mathbf{x})\mathbf{g}$ *denote the Lie derivatives of the inequality constraints in the direction of* $\mathbf{f}$ *and in the direction of* $\mathbf{g}$*, respectively, positive scalar* u_{in}^{+} *is the switching gain,* $\mathbf{W}_{\text{in}}$ *is a diagonal matrix representing the switching gain weights for the inequality constraints,* $L_f\phi_{\text{in},i}^{a}$ *is used to represent the active inequality constraints, i.e., those with* $\phi_{\text{in},i} \geq 0$*, and* N_a *is the number of active inequality constraints.*

Note that the expression $\text{v2dm}(\text{pos}(\boldsymbol{\phi}_{in}))$ on the left side of Eq. (13) is used to obtain the trivial scalar equation $0 = 0$ for the nonactive inequality constraints (i.e., those with $\phi_{\text{in},i} < 0$) and, hence, no degrees-of-freedom of the system are used by these constraints.

Note also that pos(·) is a *commutation function* that plays the same role as the sign function sign(·) typically used in

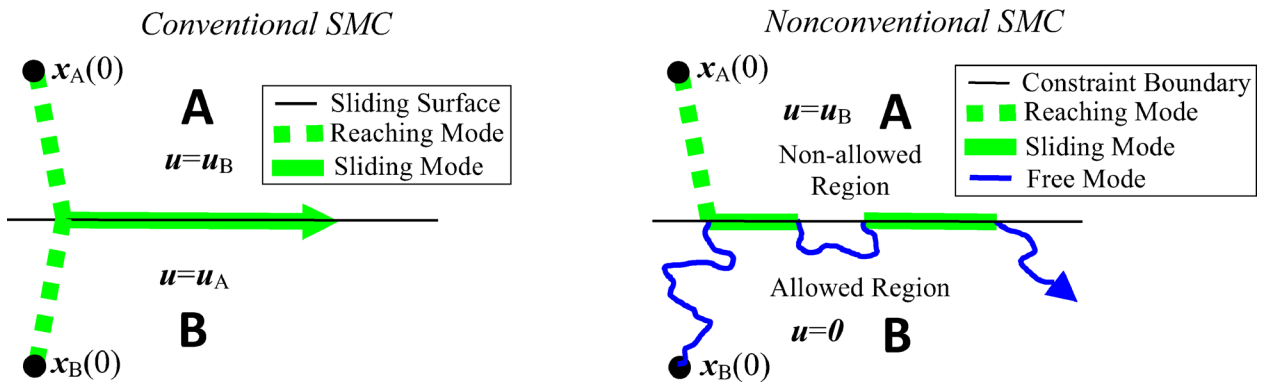

Fig. 1 Graphical comparison between conventional SMC (left) and nonconventional SMC (right)

conventional SMC [21]. Hence, while by definition, the values of pos(x) and sign(x) fall within the intervals [0, 1] and [−1], respectively; at the commutation point $x=0$, they are actually indeterminate. Although it is unlikely that x equals exactly zero, to be conservative, pos(0) = 1 will be used in order to guarantee the fulfillment of the inequality constraints.

Proof. First, the inequality constraint vector is partitioned into two subvectors $\boldsymbol{\phi}_{\text{in}} = [\boldsymbol{\phi}_{\text{in}}^{a\text{T}} \quad \boldsymbol{\phi}_{\text{in}}^{na\text{T}}]^{\text{T}}$, where the first subvector is composed of the N_a active inequality constraints (i.e., those with $\phi_{\text{in},i} \geq 0$) and the second subvector of the remaining $N_{\text{in}} - N_a$ nonactive inequality constraints (i.e., those with $\phi_{\text{in},i} < 0$).

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

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

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

Premultiplying Eq. (15) by v2dm(pos($\boldsymbol{\phi}_{\text{in}}$)) and substituting Eq. (13) yields

$$\text{v2dm}(\mathbf{z}_{\text{in}}) \dot{\boldsymbol{\phi}}_{\text{in}} = \text{v2dm}(\mathbf{z}_{\text{in}}) L_f \boldsymbol{\phi}_{\text{in}} - \mathbf{W}_{\text{in}}\, \mathbf{z}_{\text{in}}\, u_{\text{in}}^{+} \tag{16}$$

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

Let $V_{\text{in}} = \mathbf{z}_{\text{in}}^{\text{T}}\, \text{v2dm}(\mathbf{z}_{\text{in}})\, \boldsymbol{\phi}_{\text{in}}$ be a Lyapunov function candidate. Vector ϕ_{in}^{a} can be generically partitioned into two subvectors $\boldsymbol{\phi}_{\text{in}}^{a} = [\boldsymbol{\phi}_{\text{in}}^{b\text{T}} \quad \boldsymbol{\phi}_{\text{in}}^{nb\text{T}}]^{\text{T}}$, where SM occurs in the manifold given by $\boldsymbol{\phi}_{\text{in}}^{b} = \mathbf{0}$, whereas the components of vector $\boldsymbol{\phi}_{\text{in}}^{nb}$ are greater than zero. Obviously, one of these two subvectors may be empty at a certain time. Since vectors $\mathbf{z}_{\text{in}}^{nb} = 1$ and $\mathbf{z}_{\text{in}}^{na} = \mathbf{0}$ are constant, the time derivative of V results in

$$\dot{V}_{\text{in}} = \frac{d\left(\begin{bmatrix} \mathbf{z}_{\text{in}}^{b} \\ 1 \\ 0 \end{bmatrix}^{\text{T}} \text{v2dm}\left(\begin{bmatrix} \mathbf{z}_{\text{in}}^{b} \\ 1 \\ 0 \end{bmatrix} \right) \right)}{dt} \begin{bmatrix} \mathbf{0} \\ \boldsymbol{\phi}_{\text{in}}^{nb} \\ \boldsymbol{\phi}_{\text{in}}^{na} \end{bmatrix} + \mathbf{z}_{\text{in}}^{\text{T}}\, \text{v2dm}(\mathbf{z}_{\text{in}})\, \dot{\boldsymbol{\phi}}_{\text{in}} = \mathbf{z}_{\text{in}}^{\text{T}}\, \text{v2dm}(\mathbf{z}_{\text{in}})\, \dot{\boldsymbol{\phi}}_{\text{in}} \tag{17}$$

Substituting Eq. (16) in Eq. (17) yields

$$\dot{V}_{\text{in}} = \mathbf{z}_{\text{in}}^{\text{T}}\, \text{v2dm}(\mathbf{z}_{\text{in}})\, L_f \boldsymbol{\phi}_{\text{in}} - \mathbf{z}_{\text{in}}^{\text{T}}\, \mathbf{W}_{\text{in}}\, \mathbf{z}_{\text{in}}\, u_{\text{in}}^{+} \tag{18}$$

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

$$\mathbf{z}_{\text{in}}^{\text{T}}\, \text{v2dm}(\mathbf{z}_{\text{in}})\, L_f \boldsymbol{\phi}_{\text{in}} \leq \sum_{i=1}^{N_a} (\max(L_f \phi_{\text{in},i}^{a}, 0)) \tag{19}$$

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

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

because if vector $\boldsymbol{\phi}_{\text{in}}^{nb}$ is not empty at least one component of vector $\mathbf{z}_{\text{in}}$ is equal to 1.

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

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

Therefore, if u_{in}^{+} fulfills Eq. (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 Sliding-Mode Control to Satisfy Both Equality and Inequality Constraints. Combining the results of Secs. 3.1 and 3.2, the SMC to satisfy simultaneously both equality and inequality constraints is given by

$$\begin{bmatrix} \mathbf{L}_{\mathbf{g}} \boldsymbol{\phi}_{\text{eq}} \\ \text{v2dm}(\text{pos}(\boldsymbol{\phi}_{\text{in}}))\, \mathbf{L}_{\mathbf{g}} \boldsymbol{\phi}_{\text{in}} \end{bmatrix} \mathbf{u} = -\begin{bmatrix} \mathbf{W}_{\text{eq}} & \mathbf{0} \\ \mathbf{0} & \mathbf{W}_{\text{in}} \end{bmatrix} \begin{bmatrix} \text{sign}(\boldsymbol{\phi}_{\text{eq}}) \\ \text{pos}(\boldsymbol{\phi}_{\text{in}}) \end{bmatrix} u^{+}$$
$$\mathbf{L}_{\mathbf{g}} \boldsymbol{\phi}\, \mathbf{u} = -\mathbf{W} \begin{bmatrix} \text{sign}(\boldsymbol{\phi}_{\text{eq}}) \\ \text{pos}(\boldsymbol{\phi}_{\text{in}}) \end{bmatrix} u^{+} \tag{22}$$

where u^{+} is the switching gain for the global SMC, whose lower bound is given by

$$u^{+} > \|L_f \boldsymbol{\phi}_{\text{eq}}\|_1 / \text{diag}_{\min}(\mathbf{W}_{\text{eq}}) + \sum_{i=1}^{N_a} (\max(L_f \phi_{\text{in},i}^{a}, 0)) / \text{diag}_{\min}(\mathbf{W}_{\text{in}}) \tag{23}$$

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

$$\phi_{\text{eq},i} = \sigma_{\text{eq},i} + K_{\text{eq},i} \dot{\sigma}_{\text{eq},i} = 0 \tag{24}$$

$$\phi_{\text{in},i} = \sigma_{\text{in},i} + K_{\text{in},i} \dot{\sigma}_{\text{in},i} \leq 0 \tag{25}$$

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

3.5 Chattering-Free Sliding Mode Control. Discrete-time implementations of the above SMC make the system leave the ideal SM and oscillate with finite frequency and amplitude inside a band around $\boldsymbol{\phi} = \mathbf{0}$, which is called *chattering* [19]. Several approaches can be found in the literature [22,23] to theoretically avoid this drawback, as discussed below. However, it is interesting to remark that these approaches smooth the SMC behavior and, hence, they may bring issues like "drift."

3.5.1 Softened Discontinuous Control. The chattering issue can be reduced by softening the discontinuous control part by a continuous approximation. Two typical options used in the literature for this purpose are the saturation and the hyperbolic tangent functions [19]. In particular, this work considers the following continuous approximations for the discontinuous terms on the right side of Eq. (22):

$$\text{sign}(\boldsymbol{\phi}_{\text{eq}}) \rightarrow \tanh(\boldsymbol{\phi}_{\text{eq}} / K_{ss}) \tag{26}$$

$$\text{pos}(\boldsymbol{\phi}_{\text{in}}) \rightarrow \text{tanhp}(\boldsymbol{\phi}_{\text{in}} / K_{sp}) \tag{27}$$

where $\tanh(\cdot)$ is the hyperbolic tangent function, tanhp($\cdot$) is the positive hyperbolic tangent function, i.e., tanhp(x) is equal to $\tanh(x)$ if $x \geq 0$ and 0 otherwise, and K_{ss} and K_{sp} are the softening parameters that establish the smoothness of the commutation.

3.5.2 Dynamical Sliding Mode Control. The chattering drawback can also be avoided using a dynamical or high-order SMC.

In particular, instead of using the first-order SMC given by Eq. (22) together with the modified constraints given by Eqs. (24) and (25), a second-order SMC can be used. However, this option only applies for the equality constraints since the SMC for the inequality constraints is only active for certain periods (see Fig. 1) and, hence, it makes no sense to use a dynamical control law, i.e., a control law with memory.

This work considers three typical second-order SMCs for the equality constraints: quasi-continuous [24], twisting [25], and super-twisting [26]. In particular, the discontinuous term $\text{sign}(\boldsymbol{\phi}_{\text{eq}})$ of the equality constraints on the right side of Eq. (22) is replaced by

$$\text{sign}\left(\dot{\boldsymbol{\sigma}}_{\text{eq}} + K_{\text{qc}}\sqrt{|\boldsymbol{\sigma}_{\text{eq}}|}\text{sign}(\boldsymbol{\sigma}_{\text{eq}})\right) \quad \text{quasi-continuous} \tag{28}$$

$$\text{sign}(\dot{\boldsymbol{\sigma}}_{\text{eq}}) + K_t\text{sign}(\boldsymbol{\sigma}_{\text{eq}}) \quad \text{twisting} \tag{29}$$

$$\sqrt{|\boldsymbol{\phi}_{\text{eq}}|}\text{sign}(\boldsymbol{\phi}_{\text{eq}}) + K_{\text{st}}\int \text{sign}(\boldsymbol{\phi}_{\text{eq}})dt \quad \text{super-twisting} \tag{30}$$

where K_{qc}, K_t, and K_{st} are the design parameters of the quasi-continuous, twisting, and super-twisting SMC algorithms, respectively.

4 Method

4.1 Overview of the Method. The following three prioritized tasks are considered:

- The first level (high-priority task) includes the equality and inequality *constraints* that must be satisfied at all times to properly accomplish the treatment on the surface with the robot tool. In particular, equality constraints are defined to exert the desired pressure between the tool and the surface being treated, and to keep the tool orientation perpendicular to the surface. Moreover, an inequality constraint is defined to make the tool compliant with respect to any surface obstacles (e.g., protrusions, screws, and nuts) when applying the treatment.
- The second level (medium-priority task) is designed 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 such deviations are required to satisfy the above constraints.
- The third level (low-priority task) is considered only for the case of *redundant* robots (e.g., the one used in the experiments in Sec. 5) in order to keep the robot close to the home configuration.

The input to these tasks is the robot state $\{\mathbf{q}, \dot{\mathbf{q}}\}$, and the vector $\mathbf{F}$ of forces and torques detected by a sensor located at the robot tool, whereas each task gives an acceleration equality whose square error must be minimized.

4.2 Lie Derivatives. The acceleration equality for the first level is obtained below using the SMC presented in Sec. 3. In order to use this theory, a dynamical system in the form of Eq. (8) is considered with the state vector $\mathbf{x} = \begin{bmatrix}\mathbf{q}^{\text{T}} & \dot{\mathbf{q}}^{\text{T}}\end{bmatrix}^{\text{T}}$, 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 Eq. (4), the state equation results in

$$\dot{\mathbf{x}} = \begin{bmatrix}\mathbf{O} & \mathbf{I} \\ \mathbf{O} & \mathbf{O}\end{bmatrix}\mathbf{x} + \begin{bmatrix}\mathbf{0} \\ \mathbf{d}_c\end{bmatrix} + \begin{bmatrix}\mathbf{O} \\ \mathbf{I}\end{bmatrix}\mathbf{u} \tag{31}$$

and, therefore, the Lie derivatives for the constraint function ϕ_i are given by

$$\mathbf{L}_{\mathbf{g}}\phi_i = \nabla\phi_i^{\text{T}}\,\mathbf{g} = (\partial\phi_i/\partial\dot{\mathbf{q}})^{\text{T}} \tag{32}$$

$$L_f\phi_i = \nabla\phi_i^{\text{T}}\,\mathbf{f} = (\partial\phi_i/\partial\mathbf{q})^{\text{T}}\,\dot{\mathbf{q}} + (\partial\phi_i/\partial\dot{\mathbf{q}})^{\text{T}}\,\mathbf{d}_c \tag{33}$$

4.3 Level 1: Constraints

4.3.1 Force Model. The constraints for surface treatment are defined below depending on the vector $\mathbf{F}$ of forces and torques between the tool and the environment, which are measured by a force sensor located at the robot end-effector. In many applications, the interaction forces between the tool and the environment can be represented by

$$\mathbf{F} = \mathbf{K}_s\,\Delta\mathbf{s}(\mathbf{q}, \mathbf{p}_s) + \mathbf{d}_s = \begin{bmatrix}F_x & F_y & F_z & F_\alpha & F_\beta & F_\gamma\end{bmatrix}^{\text{T}} \tag{34}$$

where vector $\mathbf{F}$ is relative to the tool coordinate system, $\mathbf{K}_s$ is a positive definite matrix with the stiffness coefficients, 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, and vector $\mathbf{d}_s$ represents the dynamics not modeled by the classical elastic model [27] given by the first term in Eq. (34).

4.3.2 Equality Constraints. Three equality constraints are defined for the surface treatment as follows:

$$\sigma_{\text{eq},z}(\mathbf{q}, t) = \begin{bmatrix}0 & 0 & 1 & 0 & 0 & 0\end{bmatrix}^{\text{T}}\mathbf{F} - F_{z,\text{ref}} = 0 \tag{35}$$

$$\sigma_{\text{eq},\alpha}(\mathbf{q}, t) = \begin{bmatrix}0 & 0 & 0 & 1 & 0 & 0\end{bmatrix}^{\text{T}}\mathbf{F} = 0 \tag{36}$$

$$\sigma_{\text{eq},\beta}(\mathbf{q}, t) = \begin{bmatrix}0 & 0 & 0 & 0 & 1 & 0\end{bmatrix}^{\text{T}}\mathbf{F} = 0 \tag{37}$$

where the first equality constraint is used to attain the desired force $F_{z,\text{ref}}$ between the tool and the surface being treated in the tool Z-axis (which is longitudinal to the robot end effector), 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 (i.e., F_α and F_β) 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 Eqs. (24), (32), and (34)–(37), the Lie derivative $\mathbf{L}_{\mathbf{g}}\boldsymbol{\phi}_{\text{eq}}$ required for the SMC in Eq. (22) is given by

$$\begin{aligned}\mathbf{L}_{\mathbf{g}}\boldsymbol{\phi}_{\text{eq}} &= \left(\partial\boldsymbol{\phi}_{\text{eq}}/\partial\dot{\mathbf{q}}\right)^{\text{T}} = \mathbf{K}_{\text{eq}}\left(\partial\boldsymbol{\sigma}_{\text{eq}}/\partial\mathbf{q}\right)^{\text{T}} \\ &= \mathbf{K}_{\text{eq}}\begin{bmatrix}0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0\end{bmatrix}\mathbf{K}_s\mathbf{J}_n = \mathbf{K}_{\text{eq}}\,\mathbf{H}_{\text{eq}}\,\mathbf{K}_s\,\mathbf{J}_n\end{aligned} \tag{38}$$

where $\boldsymbol{\sigma}_{eq}$ is a column vector composed of all equality constraints $\sigma_{\text{eq},i}$, $\mathbf{K}_{\text{eq}}$ is a diagonal matrix composed of all parameters $K_{\text{eq},i}$, and $\mathbf{J}_n$ is the geometric Jacobian relative to the tool coordinate system [27], 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.

4.3.3 Inequality Constraint. The following constraint is used to adapt the tool position to the surface obstacles:

$$\sigma_{\text{in},xy}(\mathbf{q}, t) = \sqrt{F_x^2 + F_y^2} - F_{xy,\text{max}} = F_{xy} - F_{xy,\text{max}} \leq 0 \tag{39}$$

where F_{xy} is the linear force measured by the sensor in the tool XY plane and $F_{xy,\text{max}}$ is the maximum allowed value for this force.

Taking into account Eqs. (25), (32), (34), and (39), the Lie derivative $\mathbf{L_g}\boldsymbol{\phi}_{\text{in}}$ for the SMC in Eq. (22) is given by

$$\begin{aligned}\mathbf{L_g}\boldsymbol{\phi}_{\text{in}} &= \left(\partial\phi_{\text{in},xy}/\partial\dot{\mathbf{q}}\right)^{\text{T}} = K_{\text{in}}\left(\partial\sigma_{\text{in},xy}/\partial\mathbf{q}\right)^{\text{T}} \\ &= K_{\text{in}}\begin{bmatrix}\bar{F}_x & \bar{F}_y & 0 & 0 & 0 & 0\end{bmatrix}\mathbf{K}_s\mathbf{J}_n = K_{\text{in}}\mathbf{H}_{\text{in}}\mathbf{K}_s\mathbf{J}_n\end{aligned} \tag{40}$$

where $K_{\text{in}} = K_{\text{in},xy}$ is the approaching parameter to the original constraint in Eq. (39), and $\bar{F}_x = F_x/F_{xy}$ and $\bar{F}_y = F_y/F_{xy}$ represent the normalized linear forces in the tool X- and Y-axes, respectively.

4.3.4 Acceleration Equality for Level 1. The stiffness coefficients of matrix $\mathbf{K}_s$ are typically not known. However, since the nondiagonal elements of $\mathbf{K}_s$ are significantly smaller than those in the main diagonal, it is assumed below that matrix $\mathbf{K}_s$ is diagonal in order to avoid computing these coefficients. The computation of the Lie derivatives $\mathbf{L_g}\boldsymbol{\phi}_{\text{eq}}$ and $\mathbf{L_g}\boldsymbol{\phi}_{\text{in}}$ in Eqs. (38) and (40), respectively, using this assumption implies that the result will not be completely accurate. However, although the Lie derivative $\mathbf{L_g}\phi_i$ provides the gradient vector perpendicular to the sliding surface or constraint boundary given by $\phi_i = 0$, a similar direction can also be used by the SMC to switch the value of the constraint functions ϕ_i from positive to negative or vice versa, see Fig. 1. For instance, experimental results are shown in Ref. [28] for conventional SMC introducing a 30% error in the gradient computation and the control system works properly. Although the minimum required value for the switching gain u^+ is reduced if the actual value of $\mathbf{L_g}\phi_i$ is used.

Under the foregoing assumption, the stiffness coefficients of the diagonal matrix $\mathbf{K}_s$ can be included without loss of generality in the switching gain weight matrix $\mathbf{W}$, so that the SMC given by Eq. (22) is modified as follows:

$$\begin{aligned}&\begin{bmatrix}\mathbf{K}_{\text{eq}}\mathbf{H}_{\text{eq}} \\ \text{v2dm}(\text{pos}(\boldsymbol{\phi}_{\text{in}}))K_{\text{in}}\mathbf{H}_{\text{in}}\end{bmatrix}\mathbf{J}_n\ddot{\mathbf{q}}_c \\ &= -\begin{bmatrix}\bar{\mathbf{W}}_{\text{eq}} & \mathbf{0} \\ \mathbf{0} & \bar{W}_{\text{in}}\end{bmatrix}\begin{bmatrix}\text{sign}(\boldsymbol{\phi}_{\text{eq}}) \\ \text{pos}(\boldsymbol{\phi}_{\text{in}})\end{bmatrix}u^+ = -\bar{\mathbf{W}}\begin{bmatrix}\text{sign}(\boldsymbol{\phi}_{\text{eq}}) \\ \text{pos}(\boldsymbol{\phi}_{\text{in}})\end{bmatrix}u^+ \\ &\rightarrow\ \mathbf{A}_1\ddot{\mathbf{q}}_c = \mathbf{b}_1\end{aligned} \tag{41}$$

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

$$\bar{\mathbf{W}}_{\text{eq}} = \begin{bmatrix}W_{\text{eq},z}/K_{s,z} & 0 & 0 \\ 0 & W_{\text{eq},\alpha}/K_{s,\alpha} & 0 \\ 0 & 0 & W_{\text{eq},\beta}/K_{s,\beta}\end{bmatrix} \tag{42}$$

$$\bar{W}_{\text{in}} = W_{\text{in}}/K_{s,xy} \tag{43}$$

where $\{K_{s,xy}, K_{s,z}\}$ are the stiffness coefficients for the linear X/Y- and Z-axes, respectively, and $\{K_{s,\alpha}, K_{s,\beta}\}$ are the stiffness coefficients for the rotational X- and Y-axes, respectively. Note that it has been assumed the same stiffness coefficient for both linear X- and Y-axes and, hence, when the inequality constraint is active, the XY-component of the tool motion given by Eq. (41) is in the opposite direction to the force vector $\begin{bmatrix}F_x & F_y\end{bmatrix}^{\text{T}}$ detected by the sensor, see $\mathbf{H}_{\text{in}}$ in Eq. (40).

The SMC given by Eq. (41) only requires the control parameters $\{u^+, \bar{W}_{\text{eq},i}, \bar{W}_{\text{in}}, K_{\text{eq},i}, K_{\text{in}}\}$, the robot Jacobian, and the constraint functions $\{\phi_{\text{eq},z}, \phi_{\text{eq},\alpha}, \phi_{\text{eq},\beta}, \phi_{\text{in},xy}\}$, which are computed from the force sensor measurements and their first-order time derivatives.

For the case of the chattering-free alternatives described in Sec. 3.5, the commutation functions $\text{sign}(\boldsymbol{\phi}_{\text{eq}})$ and $\text{pos}(\boldsymbol{\phi}_{\text{in}})$ on the right side of Eq. (41) are replaced by the expressions in Eqs. (26)–(30). Note that, in this case, the following additional control parameters are introduced: $\{K_{ss}, K_{sp}, K_{qc}, K_t, K_{st}\}$.

4.4 Level 2: Reference Tracking. The following equality is considered for this level:

$$\begin{aligned}&\mathbf{J}\ddot{\mathbf{q}}_c = \ddot{\mathbf{p}}_{\text{ref}} + K_{T,v}\dot{\mathbf{e}} + K_{T,p}\mathbf{e} + \text{sign}\left(\dot{\mathbf{e}} + K_{T,p}K_{T,v}^{-1}\mathbf{e}\right)u_T^+ \\ &\rightarrow\ \mathbf{A}_2\ddot{\mathbf{q}}_c = \mathbf{b}_2\end{aligned} \tag{44}$$

where $\mathbf{p}_{\text{ref}}$ is the reference for the tool pose, $\mathbf{e} = \mathbf{p}_{\text{ref}} - \mathbf{p}$ 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_T^+ is the switching gain for the last term, and $\mathbf{A}_2$ and $\mathbf{b}_2$ are the values for the second task in Eq. (5). In this hybrid control law, the switching term represents a conventional SMC used to compensate the term $\dot{J}\dot{q}$ 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.

4.5 Level 3: Home Configuration. This level is considered only for the case of *redundant* robots since otherwise there are no remaining degrees-of-freedom at this level. While there are various options described in the literature [29], this work considers "pushing" the robot to a home configuration $\mathbf{q}_0$ for increasing safety, away from critical areas due to, e.g., joint limits, singular configurations, or possible obstacles in the robot workspace. To achieve this purpose, the following equality is considered:

$$\ddot{\mathbf{q}}_c = -K_{3,v}\dot{\mathbf{q}} + K_{3,p}(\mathbf{q}_0 - \mathbf{q})\ \rightarrow\ \mathbf{A}_3\ddot{\mathbf{q}}_c = \mathbf{b}_3 \tag{45}$$

where $K_{3,v}$ and $K_{3,p}$ are the gains used for the velocity and position corrections, respectively, and $\mathbf{A}_3 = \mathbf{I}$ and $\mathbf{b}_3$ are the values for the third task in Eq. (5).

4.6 Additional Remarks

4.6.1 Control Action. In this work, the joint accelerations are considered as the SM discontinuous control action, which yield two advantages: the joint velocities are continuous (smoother control) and it allows to reach smoothly the boundary of the constraints in the high-priority level. If the actual control action are the joint velocities, a pure integrator can be applied to the discontinuous control signal to compute the actual continuous control action. Similarly, if the actual control action are the joint positions, a double integrator can be applied between both signals.

4.6.2 Nonstatic Constraints. The proposed approach can also be used if the constraints in level 1 are nonstatic, e.g., a *moving target object* for the equality constraints or *moving obstacles* for the inequality constraint. In this case, ϕ_i also depends explicitly on time and, therefore, the derivative of ϕ_i is replaced by $\dot{\phi}_i = \widetilde{L_f\phi}_i + \mathbf{L_g}\phi_i\,\mathbf{u}$, where $\widetilde{L_f\phi}_i$ is equal to $L_f\phi_i + \partial\phi_i/\partial t$, and $\mathbf{L_g}\phi_i$ and $L_f\phi_i$ are given again by Eqs. (32) and (33), respectively. Thus, all developments keep unchanged except for changing $L_f\phi_i$ to $\widetilde{L_f\phi}_i$. Hence, only the value of the lower bound for the switching gain u^+ is changed when non-static constraints are considered and, therefore, the iterative computation of the algorithm remains the same.

4.6.3 Stability. The stability of the SMC in the first task is guaranteed if u^+ fulfills Eq. (23) and matrix $\mathbf{L_g}\boldsymbol{\phi}$ (excluding the nonactive inequality constraints) is full row rank. That is, taking into account Eq. (41), the row rank of the robot Jacobian has to be equal to the number of equality constraints plus the number of active inequality constraints. If this is not satisfied at a certain time, e.g., the current robot configuration is singular, the robot operation should be aborted since the fulfillment of the constraints cannot be guaranteed. For the second and third levels, which represent classical kinematic and inverse kinematic control

Fig. 2 Experimental setup: 7R serial manipulator with a force sensor attached to the robot end-effector, a tool consisting of a cylinder and a flat rectangular plastic object as target

algorithms, the reader is referred to Ref. [30], where the stability of this kind of algorithms is analyzed in a task prioritization framework. Note also that the limitation of degrees-of-freedom for the tracking controller is the common situation in hybrid position-force control [9].

4.6.4 Time Derivatives. The proposed approach requires derivative of the force $\mathbf{F}$ measured by the sensor located at the robot tool. The simplest way to deal with this issue consists in using numerical differentiation, e.g., the well-known backward Euler approximation. However, some kind of filtering should be previously applied to the actual variable when non-negligible noise is present. In particular, in this work, the force signal measured by the sensor is filtered with a discrete first-order low-pass filter, where ω_c is used to represent its cutoff frequency, that is implemented in the sensor electronics (Net F/T interface). Note that the low-pass filters used for noise reduction must not limit the bandwidth of the control law.

4.7 Advantages of the Proposed Method. There are two main advantages of using SMC to satisfy the constraints for the robot task:

- *Robustness*: The SMC algorithm is robust against the Lie derivatives $L_f\phi_i$ since they are collinear [19] with the discontinuous control action. Therefore, it is not affected by the terms included in $L_f\phi_i$, such as the inaccuracies $\mathbf{d}_c$ of the low-level control loop, the position and orientation $\mathbf{p}_s$ of the object being treated by the robot and its derivative, the nonmodeled part $\mathbf{d}_s$ of the force elastic model, the derivative of the Jacobian matrix, and the joint velocities. Even more, as discussed in Sec. 4.3.4, the computation of the Lie derivatives $\mathbf{L_g}\phi_i$ does not need to be completely accurate (e.g., due to considering stiffness $\mathbf{K}_s$ as a diagonal matrix instead of a positive definite matrix, see Sec. 4.3.4), since a similar direction can also be used by the SMC to switch the value of the constraint functions ϕ_i from positive to negative or vice versa, see Fig. 1.
- *Low computational cost*: Only partial information of the system model is used, i.e., the Lie derivatives $L_f\phi_i$ are not needed (see the terms listed above), only the Lie derivatives $\mathbf{L_g}\phi_i$ are required. In particular, the SMC given by Eq. (41)

Table 1 Algorithm executed at sampling time of T_s seconds

$[\mathbf{q}, \dot{\mathbf{q}}, \mathbf{F}] = \texttt{GetRobotStateAndForce;}$	
$\mathbf{p} = \mathbf{l}(\mathbf{q})$;	// Eq. (1)
$\dot{\mathbf{p}} = \mathbf{J}\dot{\mathbf{q}}$;	// Eq. (2)
$\dot{\mathbf{p}}_{\text{ref}} = (\mathbf{p}_{\text{ref}} - \mathbf{p}_{\text{ref,prev}})/T_s$;	// Derivative
$\ddot{\mathbf{p}}_{\text{ref}} = (\dot{\mathbf{p}}_{\text{ref}} - \dot{\mathbf{p}}_{\text{ref,prev}})/T_s$;	// Derivative
$\dot{\mathbf{F}} = (\mathbf{F} - \mathbf{F}_{\text{prev}})/T_s$;	// Derivative
$\boldsymbol{\phi}_{\text{eq}} = \begin{bmatrix} F_z - F_{z,\text{ref}} + K_{\text{eq},z}\dot{F}_z \\ F_\alpha + K_{\text{eq},\alpha}\dot{F}_\alpha \\ F_\beta + K_{\text{eq},\beta}\dot{F}_\beta \end{bmatrix}$;	// Eqs. (24), (35)–(37)
$\phi_{\text{in}} = \sqrt{F_x^2 + F_y^2} - F_{xy,\max} + K_{\text{in}} \frac{\dot{F}_x F_x + \dot{F}_y F_y}{\sqrt{F_x^2 + F_y^2}}$;	// Eqs. (25), (39)
$\mathbf{A}_1 = \begin{bmatrix} \mathbf{K}_{\text{eq}}\mathbf{H}_{\text{eq}} \\ \text{v2dm}(\text{pos}(\boldsymbol{\phi}_{\text{in}}))K_{\text{in}}\mathbf{H}_{\text{in}} \end{bmatrix}\mathbf{J}_n$;	// Eq. (41)
$\mathbf{b}_1 = -\bar{\mathbf{W}}\begin{bmatrix} \text{sign}(\boldsymbol{\phi}_{\text{eq}}) \\ \text{pos}(\boldsymbol{\phi}_{\text{in}}) \end{bmatrix} u^+$, where the commutation functions $\text{sign}(\boldsymbol{\phi}_{eq})$ and $\text{pos}(\boldsymbol{\phi}_{in})$ are replaced with the expressions in {(26), (28), (29), (30)} and (27), respectively, for the chattering-free alternatives;	// Eq. (41)
$\mathbf{A}_2 = \mathbf{J}$; // Eq. (44)	
$\mathbf{b}_2 = \ddot{\mathbf{p}}_{\text{ref}} + K_{T,v}\dot{\mathbf{e}} + K_{T,p}\mathbf{e} + \text{sign}\left(\dot{\mathbf{e}} + K_{T,p}K_{T,v}^{-1}\mathbf{e}\right)u_T^+$;	// Eq. (44)
$\mathbf{A}_3 = \mathbf{I}$;	// Eq. (45)
$\mathbf{b}_3 = -K_{3,v}\dot{\mathbf{q}} + K_{3,p}(\mathbf{q}_0 - \mathbf{q})$;	// Eq. (45)
$\ddot{\mathbf{q}}_{c,1} = \mathbf{A}_1^{\dagger}\mathbf{b}_1$;	// Eq. (6), $i=1$
$\mathbf{N}_1 = \mathbf{I} - \mathbf{A}_1^{\dagger}\mathbf{A}_1$;	// Eq. (7), $i=1$
$\ddot{\mathbf{q}}_{c,2} = \ddot{\mathbf{q}}_{c,1} + (\mathbf{A}_2\mathbf{N}_1)^{\dagger}(\mathbf{b}_2 - \mathbf{A}_2\ddot{\mathbf{q}}_{c,1})$;	// Eq. (6), $i=2$
$\mathbf{N}_2 = \mathbf{N}_1\left(\mathbf{I} - (\mathbf{A}_2\mathbf{N}_1)^{\dagger}(\mathbf{A}_2\mathbf{N}_1)\right)$;	// Eq. (7), $i=2$
$\ddot{\mathbf{q}}_{c,3} = \ddot{\mathbf{q}}_{c,2} + (\mathbf{A}_3\mathbf{N}_2)^{\dagger}(\mathbf{b}_3 - \mathbf{A}_3\ddot{\mathbf{q}}_{c,2})$;	// Eq. (6), $i=3$
$\texttt{SendToJointControllers}(\ddot{\mathbf{q}}_{c,3})$;	
$\mathbf{p}_{\text{ref,prev}} = \mathbf{p}_{\text{ref}}$;	// For next iteration
$\dot{\mathbf{p}}_{\text{ref,prev}} = \dot{\mathbf{p}}_{\text{ref}}$;	// For next iteration
$\mathbf{F}_{\text{prev}} = \mathbf{F}$;	// For next iteration

Table 2 Values of the controller parameters used for the real experimentation

General parameters	$T_s = 20\,\text{ms}$, $\omega_c = 73\,\text{Hz}$
Level 1	$F_{z,\text{ref}} = -15\text{N}$, $F_{xy,\max} = 20\text{N}$, $K_{\text{eq},i} = K_{\text{in}} = 0.15$, $u^+ = 0.06$, $\bar{W}_{\text{eq},z} = 0.33$, $\bar{W}_{\text{eq},\alpha} = \bar{W}_{\text{eq},\beta} = 16$, $\bar{W}_{\text{in}} = 1$
Level 2	$K_{T,v} = 5.0$, $K_{T,p} = 3.0$, $u_T^+ = 0.01$, Reference: $\gamma_{\text{ref}} = -\pi/2$ and a 2D circle (radius = 80 mm, period = 10 s)
Level 3	$K_{3,v} = 0.5$, $K_{3,p} = 1.0$, $\mathbf{q}_0 = [0.003 \quad -0.577 \quad 0.002 \quad 2.044 \quad 0.104 \quad -0.067 \quad 3.382]^{\text{T}}$ rad

Table 3 System performance comparison using the standard and several chattering-free SMC algorithms

Controller		Metrics							
Equalities	Inequalities	TEC	MCA	CCA	CCF	$E_{\text{in},xy}$ (N)	$E_{\text{eq},z}$ (N)	$E_{\text{eq},\alpha\beta}$ (N mm)	E_{RT} (mm)
sign	pos	1531	0.499	0.0349	10.54	1.57	3.18	92.5	1.301
sign	tanhp	1474	0.494	0.0336	10.13	2.38	3.65	87.1	1.470
tanh	pos	794	0.185	0.0213	7.91	1.48	2.65	73.7	1.279
tanh	tanhp	564	0.064	0.0184	7.12	1.26	2.39	70.5	1.239
Quasi-continuous	pos	1178	0.489	0.0297	9.57	1.97	4.89	88.9	1.233
Quasi-continuous	tanhp	1026	0.365	0.0293	9.05	1.92	4.63	72.5	1.248
Twisting	pos	902	0.303	0.0235	8.57	1.76	2.83	81.8	1.218
Twisting	tanhp	762	0.446	0.0237	8.25	1.21	2.56	82.3	1.293
Super-twisting	pos	919	0.232	0.0235	8.58	1.81	4.59	60.7	1.294
Super-twisting	tanhp	693	0.130	0.0221	9.54	1.17	4.38	52.6	1.205

only requires the robot Jacobian and the constraint functions $\phi_{\text{eq},i}$ and $\phi_{\text{in},i}$ which are computed from the force sensor measurement $\mathbf{F}$ and its time derivative. Hence, the proposed approach only requires a few program lines and has reduced computation time, see Sec. 5.1.

Like in other SMC applications, the main disadvantage of the proposed method is the *chattering* drawback and, hence, several options are considered in the real experimentation of next section to alleviate this issue, see Sec. 3.5.

5 Real Experimentation

The setup used for the experiments consists of (Fig. 2) a Sawyer collaborative robot, a force/torque sensor Nano25 SI-25-25 attached to the robot end-effector, a tool consisting of a cylinder of $43 \times 43 \times 10\,\text{mm}$ attached to the sensor to emulate a surface preparation operation, and a flat rectangular plastic object 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 robot, force sensor, and external PC are connected to a router and communicate via UDP protocol.

5.1 Controller Implementation. The pseudo-code of the proposed method is shown in Table 1. The algorithm is executed at T_s seconds sampling time and incorporates the following auxiliary functions: kinematic function $\mathbf{l}(\mathbf{q})$; 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 [31] in order to avoid extremely large values for the commanded accelerations; *GetRobotStateAndForce*, which returns the current robot state $\{\mathbf{q}, \dot{\mathbf{q}}\}$ and force vector $\mathbf{F}$, which has been already filtered by the sensor electronics; and *SendToJointControllers* $(\ddot{\mathbf{q}}_c)$, which sends the current commanded joint acceleration vector to the joint controllers. The computation time per iteration of the algorithm in a modern computer using compiled C code was around 15 μs for the experiment in Sec. 5.

5.2 Guidelines to Design the Control Parameters. The procedure followed in the real experimentation to design the control parameters of the conventional and nonconventional SMCs in level 1 is very similar. In particular, the steps to design the control parameters of the proposed control method are as follows. First, the cut-off frequency of the force sensor filter has to be selected to effectively remove the measurement noise. Then, the control period T_s must be selected guaranteeing that the SM frequency $f_{\text{SM}} = (2Ts)^{-1}$ is lower than the bandwidths of the sensor filter 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 $K_{\text{eq},i}$ and K_{in}), level 2 (given by $K_{T,p}$ and $K_{T,v}$), and level 3 (given by

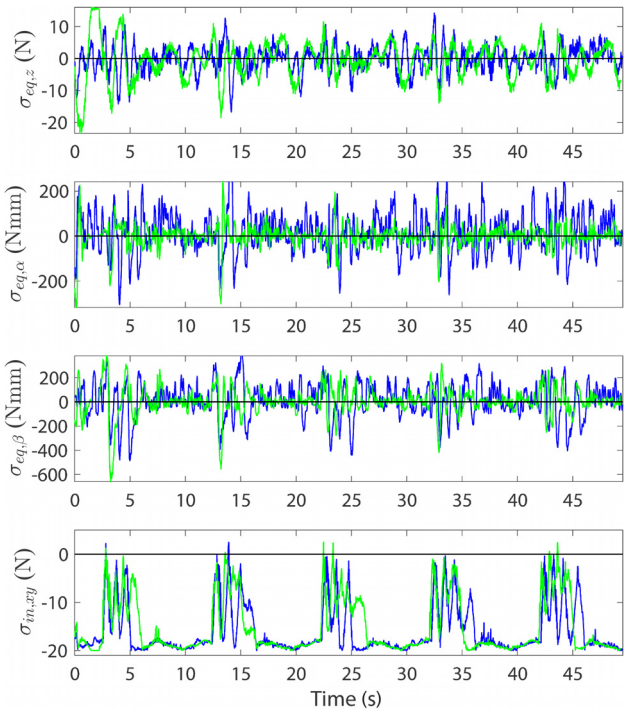

Fig. 3 Comparison of the constraint functions σ_i for the standard (dark blue) and chattering-free (light green) SMCs

$K_{3,v}$ and $K_{3,p}$) 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. Reference force $F_{z,\text{ref}}$ can then be established depending on the requirements of the actual surface treatment task. Moreover, force $F_{xy,\max}$ can be established as small as possible to be sensitive to the surface obstacles but not too small to avoid the accidental activation of the inequality constraint due to the pressure between the tool and the surface and the movement of the tool on the surface (in practice, one-time accidental activation is admissible, full-time accidental activation is not). Finally, the switching gains $\{u^+, u_T^+\}$, weights $\{\bar{W}_{\text{eq},i}, \bar{W}_{in}\}$, and chattering-free parameters $\{K_{ss}^{-1}, K_{sp}^{-1}, K_{qc}, K_t, K_{st}\}$ can be empirically tuned to be as small as possible in order to reduce the chattering effect while guaranteeing that the SM behavior of the control action remains effective for the task at hand.

5.3 Experiment Conditions. The commanded joint accelerations $\ddot{\mathbf{q}}_c$ computed by the proposed algorithm are integrated to obtain the commanded joint velocities $\dot{\mathbf{q}}_c$ sent to the robot controller. Moreover, the parameter values used for the real experimentation are shown in Table 2. Furthermore, for the SMC in level 1, which is given by Eq. (41), several implementations have been considered for the experiments:

- Five options have been tested for the conventional SMC (which is used to satisfy the equality constraints): standard SMC, i.e., using the commutation function $\text{sign}(\boldsymbol{\phi}_{\text{eq}})$ on the right side of Eq. (41) and other four chattering-free SMCs, which consist in replacing the term $\text{sign}(\boldsymbol{\phi}_{\text{eq}})$ with the expressions in Eqs. (26) and (28)–(30).
- Two options have been tested for the nonconventional SMC (which is used to satisfy the inequality constraint): standard SMC, i.e., using the commutation function $\text{pos}(\boldsymbol{\phi}_{\text{in}})$ on the right side of Eq. (41) and another chattering-free SMC, which consists in replacing the term $\text{pos}(\boldsymbol{\phi}_{\text{in}})$ with the expressions in Eq. (27).

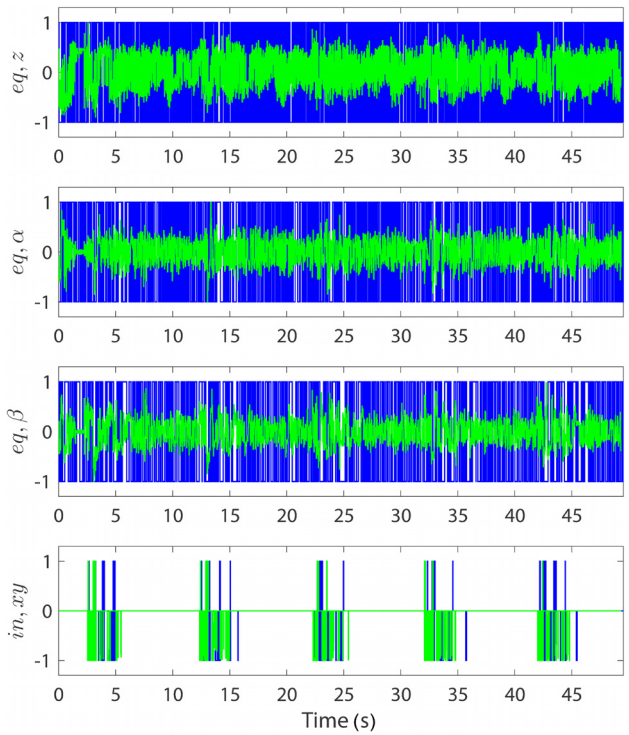

Fig. 4 Comparison of the commutation functions for the standard (dark blue) and chattering-free (light green) SMCs

Therefore, ten combinations have been tested for the SMC in level 1, whose parameters have been empirically tuned to obtain a good performance in each case. The complete list of parameter values used for each case is omitted for brevity.

5.4 Remark About the Comparison Study. The experimental results shown below for the proposed SMC approach present a comparison between several chattering-free options available in the literature. The objective of this study is not to establish the best alternative to alleviate the chattering issue. In fact, a fair comparison cannot be easily guaranteed since the results are very dependent on the controller parameters, the reference trajectory, the robot task, etc. Probably, none of the tested methods would get the best results for all possible cases and tasks. Therefore, the objective of this study is just to show how the chattering drawback can be partially mitigated by slightly modifying the standard SMC.

5.5 Results. The first comparison experiment consists in tracking a circular trajectory with a fixed obstacle that limits the progress of the tool along the reference trajectory in the X-axis at a certain point. The following metrics will be used to study the quality and robustness of all the control cases:

(a) *Total energy consumption* (TEC): $\int_0^{t_{\text{end}}} \mathbf{u}^T \mathbf{u}\, dt$.
(b) *Maximum control action* (MCA): Average value of $\|\mathbf{u}\|_\infty$.
(c) *Magnitude of the chattering for the control action* (CCA): Taking into account that the chattering frequency is equal to $(2T_s)^{-1} = 25\text{Hz}$, which also corresponds to the Nyquist frequency of the controlled system, it will be obtained computing the average value of the magnitude of the harmonics ranging from frequency 5 Hz to 25 Hz, i.e., $20^{-1}\int_5^{25} \|\mathbf{h}_{\mathbf{u},F}\|_2 dF$, where $\mathbf{h}_{\mathbf{u},F}$ is the vector magnitude of the harmonics with frequency F for the vector signal $\mathbf{u}$. Note that the harmonics of a signal, i.e., its frequency components, can be readily obtained using the fast Fourier transform.
(d) *Magnitude of the chattering for the constraint function* ϕ (CCF): Similar to the previous case, it will be obtained computing $20^{-1}\int_5^{25} h_{\phi,F}\, dF$, where $h_{\phi,F}$ is the magnitude of the harmonic with frequency F for the signal ϕ.
(e) *Error of the inequality constraint* ($E_{\text{in},xy}$): Average error of the inequality constraint $\sigma_{\text{in},xy}$ when it is unfulfilled.

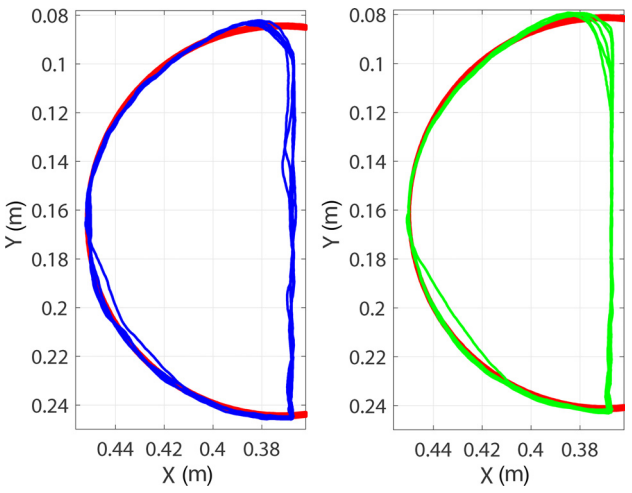

Fig. 5 Trajectory followed by the robot tool and circular reference trajectory (thick red line): left, standard SMC (dark blue); right, chattering-free SMC (light green)

(f) *Error of the linear equality constraint* ($E_{\text{eq},z}$): average error of the equality constraint $\sigma_{\text{eq},z}$.
(g) *Error of the Angular Equality Constraints* ($E_{\text{eq},\alpha\beta}$): Average error of the equality constraints $\sigma_{\text{eq},\alpha}$ and $\sigma_{\text{eq},\beta}$.
(h) *Error of the reference tracking* (E_{RT}): Average error of the reference tracking when the reference trajectory is not in the obstacle area.

Table 3 shows the results obtained for each control case. From a global point of view, the worst case is given by the standard SMC, i.e., using the commutation functions sign(·) and pos(·) for the equality and inequality constraints, respectively, followed by the second case, which uses the commutation functions sign(·) and tanhp(·) for the equality and inequality constraints, respectively.

On the one hand, the commutation functions tanh(·) and tanhp(·) for the equality and inequality constraints, respectively, provide the best performance in terms of control effort and chattering reduction: the metrics TEC, MCA, CCA, and CCF are reduced to about a 63%, 84%, 47%, and 33%, respectively, with respect to the standard SMC. On the other hand, in general, the super-twisting algorithm for the equalities together with the commutation function tanhp(·) for the inequality provides the best performance in terms of errors: the metrics $E_{\text{in},xy}$, $E_{\text{eq},\alpha\beta}$, and E_{RT} are reduced to about a 26%, 43%, and 8%, respectively, with respect to the standard SMC.

From a reliability and stability point of view, it is always preferable to minimize any mechanical chattering that may be present in the system in order to reduce undesirable stresses in the physical components, hence the chattering-free SMC used below for level 1 is the hyperbolic tangent approximation for the equalities together with the positive hyperbolic tangent approximation for the inequality. However, it is important to remark that this does

Fig. 6 Frames of the video of the dynamic experiment: the time instant is indicated for each frame: (*a*) video: 0 m 33 s; graph: 21 s, (*b*) video: 1 m 15 s; graph: 63 s, (*c*) video: 2 m 02 s; graph: 110 s, (*d*) video: 2 m 06 s; graph: 114 s, (*e*) video: 4 m 13 s; graph: 241 s, and (*f*) video: 4 m 17 s; graph: 245 s

not mean that this case gets the best results from a global point of view (e.g., see the metrics in Table 3); rather, this case is suitable to show how the chattering problem is partially mitigated, which is the objective of the comparison study presented in this work.

Figures 3–5 show a comparison between the standard SMC and the chattering-free SMC (approximation tanh for the equalities and tanhp for the inequality) in the first experiment. In particular, Fig. 3 shows the constraint functions, where it can be seen that the inequality constraint is fulfilled, i.e., $\max(\sigma_{\text{in},xy}) \leq 0$, and that it becomes active during five phases, which is due to the five times that the circular reference trajectory (period = 10 s) reaches the obstacle area during 50 s of the experiment. However, the main

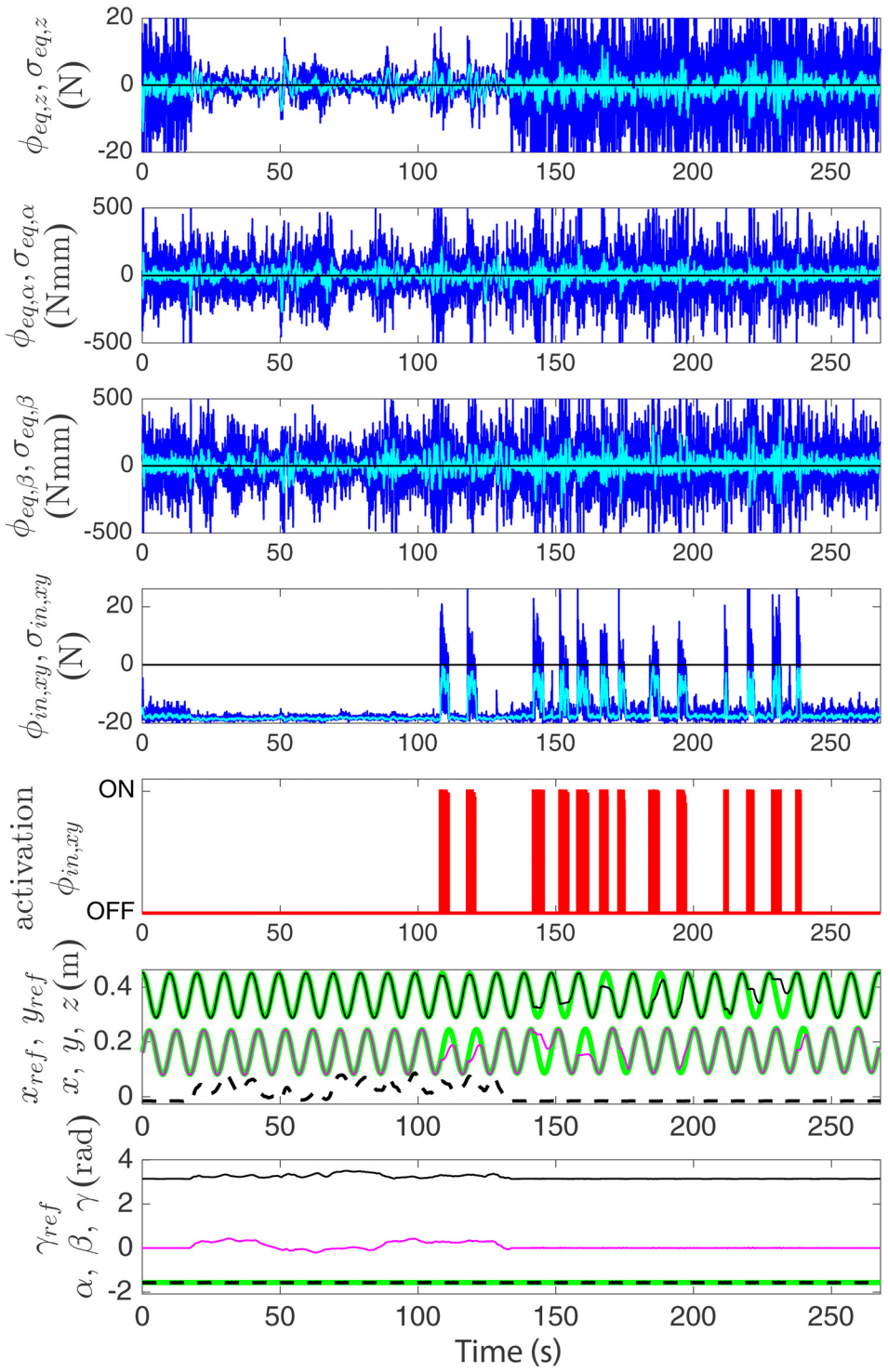

Fig. 7 Signals in the dynamic experiment as a function of time. First to fourth plots: original and modified constraint functions (σ_i in light cyan and ϕ_i in dark blue). Fifth plot: activation of the inequality constraint. Sixth plot: tool position {*x*, *y*, *z*} and reference signals {x_{ref}, y_{ref}}. Seventh plot: tool orientation {α, β, γ} in roll-pitch-yaw angles and reference signal γ_{ref}. The reference signals are in solid thick line, {*x*, α} in thin dark line, {*y*, β} in thin light line, and {*z*, γ} in dashed line.

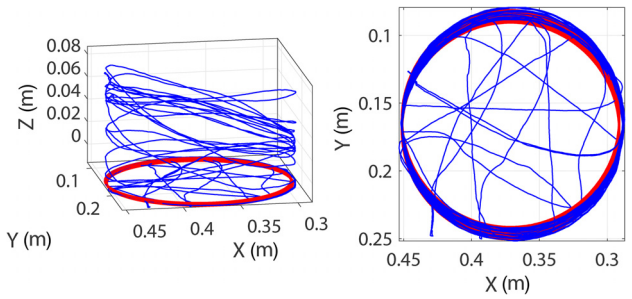

Fig. 8 Trajectory followed by the robot tool (thin blue line) and circular reference trajectory (thick red line): left, 3D representation; right, top view

purpose of this figure is to show that the chattering is significantly smaller for the chattering-free SMC, especially for the equality constraints, which gives rise to a smoother behavior as can be seen in the commutation functions depicted in Fig. 4 and in the trajectory followed by the tool shown in Fig. 5. A video of this comparative experiment can be played at the website link.[2]

In order to verify the robustness and performance of the proposed approach, a more dynamic experiment has been conducted introducing different perturbations during the tracking task: changes in the object position and orientation together with dynamic obstacles along the trajectory. The video of this experiment can be played at the website link,[2] whereas Fig. 6 shows several instants of the experiment: Figs. 6(*a*), and 6(*b*) (interval 0 m 33 s–1 m 15 s) show the system behavior when surface variations are present; Figs. 6(*c*) and 6(*d*) (interval 2 m 02 s–2 m 06 s) show the robot behavior when the obstacle comes upon and the flat object is held above the table; and Figs. 6(*e*) and 6(*f*) (interval 4 m 13 s–4 m 17 s) show how the tool skirts the corner of the obstacle and returns to the reference path once it has been overcome.

Figure 7 shows the performance of the system in terms of constraints. It can be seen that the inequality constraint is fulfilled, i.e., $\max(\sigma_{\text{in},xy}) \leq 0$, and that it becomes active during 13 phases. In this regard, it is important to consider the following rationale. The appearance of obstacles in a totally random way along the reference trajectory is a relevant issue from a practical point of view. When the tool comes upon an obstacle (e.g., a protrusion on the surface), the inequality constraint becomes active and the tool does not continue in the same direction to prevent the tool breakage. However, it neither stops the execution of the task but continues in a direction similar to that given by the reference path and satisfying the inequality constraint. This behavior is reflected in the inequality constraint functions shown in Fig. 7. For instance, at 110 seconds, an obstacle has come up. However, $\sigma_{\text{in},xy}$ does not overcome the zero value, meaning that the inequality defined in Eq. (39) is fulfilled and, hence, the force exerted never exceeds the maximum value $F_{xy,\max}$ chosen by the user, protecting thus the integrity of the tool. This behavior is accomplished during the whole experiment, even when the flat object is held above the table.

Moreover, it can also be seen in Fig. 7 that the equality constraints are switching around zero as expected. This means that the surface treatment is being done properly: the tool orientation is perpendicular to the surface and the pressure with the tool on the surface is being kept around the desired value regardless of changes in the surface position or orientation and when obstacles are introduced along the reference trajectory. This can be better perceived in the videos of the experiments referenced earlier in this section.

[2]https://media.upv.es/player/?id=1b0e16b0-7ea2-11e8-bed3-8f377bbbf659

Figures 7 and 8 show the Cartesian trajectory followed by the reference pose and the tool pose as a function of time and the corresponding 3D representation, respectively. Note that the tracking error for the reference circular trajectory is not zero when an obstacle comes upon the trajectory (see the video), although the robot tool retrieves the reference when the obstacle is overcome.

In order to show the adaptation capability of the proposed approach, another experiment has been conducted introducing quick changes in the position and orientation of the target object. The video of this experiment can be played at the website link,[2] where it can be seen that the controller is able to dynamically adapt the tool position and orientation for wide range of values: six snapshots of this experiment are shown in Fig. 9 to illustrate that the robot tool is kept perpendicular to the surface in very challenging cases. This experiment clearly shows that the robot does not have prior knowledge about the shape and movements of the surface, and must infer them from the forces and torques measured by the sensor in order to properly accomplish the robot task. In fact, this moving target object gives rise to nonstatic equality constraints (see Sec. 4.6) that are successfully fulfilled by the proposed SMC. Obviously, the limit for the moving constraints that can be properly handled by the SMC is given by the value of the switching gain u^+.

6 Conclusions

A hybrid position-force control approach has been developed in this work for polishing tasks using task priority and sliding mode control. In particular, conventional and nonconventional sliding mode controls have been used to satisfy equality and inequality constraints, respectively, in order to perform the surface polishing task. In order to avoid the chattering drawback typically present in sliding mode control, several chattering-free controllers have been considered. Furthermore, two more tasks have been considered: a medium-priority task defined to track the desired reference trajectory on the surface being polished and, for the case of redundant robots, a low-priority task used to keep the manipulator close to the home configuration.

A significant contribution of this research is the development of a novel nonconventional sliding mode control to fulfill inequality constraints. In this regard, the inclusion of inequality constraints in mathematical and engineering problems is nothing new, e.g., optimization problems typically maximize or minimize an index subject to both equality and inequality constraints (recently, this type of problems has been solved in the literature using barrier functions). In fact, the inequality constraints proposed in this work could be easily solved analytically if a complete and accurate process model is available. Therefore, the novelty of the proposed approach consists in using sliding mode control to fulfill inequality constraints in order to benefit from its inherent robustness and low computational cost.

The applicability and effectiveness of the proposed approach have been substantiated by experimental results using a redundant 7R manipulator. In particular, an experimental comparison has been conducted among all the chattering-free controllers to evaluate their performance. However, it is important to remark that the objective of this comparison study was not to establish the best alternative (note that a fair comparison cannot be easily guaranteed); rather, the objective was to show how the chattering drawback can be partially mitigated by slightly modifying the standard SMC.

Funding Data

- Spanish Government, Agencia Estatal de Investigación (DPI2017-87656-C2-1-R).
- Generalitat Valenciana, Conselleria d'Educación, Investigación, Cultura i Esport (VALi+d APOSTD/2016/044).

Fig. 9 Frames of the video of the adaptation experiment: the time instant is indicated for each frame: (*a*) video: 0 m 19 s, (*b*) video: 0 m 55 s, (*c*) video: 1 m 22 s, (*d*) video: 1 m 33 s, (*e*) video: 1 m 40 s, and (*f*) video: 1 m 59 s

References

[1] Martínez, S. S., Ortega, J. G., García, J. G., García, A. S., and Estévez, E. E., 2013, "An Industrial Vision System for Surface Quality Inspection of Transparent Parts," Int. J. Adv. Manuf. Technol., **68**(5–8), pp. 1123–1136.

[2] Molina, J., Solanes, J. E., Arnal, L., and Tornero, J., 2017, "On the Detection of Defects on Specular Car Body Surfaces," Rob. Comput.-Integr. Manuf., **48**, pp. 263–278.

[3] Cen, L., Melkote, S., Castle, J., and Appelman, H., 2018, "A Method for Mode Coupling Chatter Detection and Suppression in Robotic Milling," ASME J. Manuf. Sci. Eng., **140**(8), p. 081015.

[4] Roswell, A., Xi, F. J., and Liu, G., 2006, "Modelling and Analysis of Contact Stress for Automated Polishing," Int. J. Mach. Tools Manuf., **46**(3–4), pp. 424–435.

[5] Shi, Y., Zheng, D., Hu, L., Wang, Y., and Wang, L., 2012, "Nc Polishing of Aspheric Surfaces Under Control of Constant Pressure Using a Magnetorheological Torque Servo," Int. J. Adv. Manuf. Technol., **58**(9–12), pp. 1061–1073.

[6] Oba, Y., and Kakinuma, Y., 2017, "Simultaneous Tool Posture and Polishing Force Control of Unknown Curved Surface Using Serial-Parallel Mechanism Polishing Machine," Precis. Eng., **49**, pp. 24–32.

[7] Massimiliano Capisani, L., and Ferrara, A., 2012, "Trajectory Planning and Second-Order Sliding Mode Motion/Interaction Control for Robot Manipulators in Unknown Environments," IEEE Trans. Ind. Electron., **59**(8), pp. 3189–3198.

[8] Zhao, Y., Lin, Y., Xi, F., Guo, S., and Ouyang, P., 2017, "Switch-Based Sliding Mode Control for Position-Based Visual Servoing of Robotic Riveting System," ASME J. Manuf. Sci. Eng., **139**(4), p. 041010.

[9] Raibert, M., and Craig, J., 1981, "Hybrid Position/Force Control of Manipulators," ASME J. Dyn. Syst., Meas., Control, **102**(2), pp. 126–133.

[10] Tian, F., Li, Z., Lv, C., and Liu, G., 2016, "Polishing Pressure Investigations of Robot Automatic Polishing on Curved Surfaces," Int. J. Adv. Manuf. Technol., **87**(1–4), pp. 639–646.
[11] Armand, P., and Omheni, R., 2017, "A Mixed Logarithmic Barrier-Augmented Lagrangian Method for Nonlinear Optimization," J. Optim. Theory Appl., **173**(2), pp. 523–547.
[12] Tiwari, S., Reddy, T., and Upadhyay, P., 2017, "Error Performance Optimization Using Logarithmic Barrier Function in Molecular Nanonetworks," IEEE Commun. Lett., **21**(11), pp. 2408–2411.
[13] Bartolini, G., Ferrara, A., Levant, A., and Usai, E., 1999, *On Second Order Sliding Mode Controllers*, Springer, London, pp. 329–350.
[14] Utkin, V., 2016, "Discussion Aspects of High-Order Sliding Mode Control," IEEE Trans. Autom. Control, **61**(3), pp. 829–833.
[15] Lopes, A., and Almeida, F., 2008, "A Force-Impedance Controlled Industrial Robot Using an Active Robotic Auxiliary Device," Robot. Comput. Integr. Manuf., **24**(3), pp. 299–309.
[16] Yao, B., Zhou, Z., Wang, L., Xu, W., Liu, Q., and Liu, A., 2018, "Sensorless and Adaptive Admittance Control of Industrial Robot in Physical Human-Robot Interaction," Robot. Comput. Integr. Manuf., **51**(1), pp. 158–168.
[17] Nakamura, Y., Hanafusa, H., and Yoshikawa, T., 1987, "Task-Priority Based Redundancy Control of Robot Manipulators," Int. J. Rob. Res., **6**(2), pp. 3–15.
[18] Siciliano, B., and Slotine, J., 1991, "A General Framework for Managing Multiple Tasks in Highly Redundant Robotic Systems," Fifth International Conference on Advanced Robotics (ICAR), Pisa, Italy, June 19–22, pp. 1211–1216.
[19] Edwards, C., and Spurgeon, S., 1998, *Sliding Mode Control: Theory and Applications*, 1st ed., Taylor & Francis, London.
[20] Utkin, V., Guldner, J., and Shi, J., 2009, *Sliding Mode Control in Electro-Mechanical Systems*, 2nd ed., Taylor & Francis, London.
[21] Shtessel, Y., Edwards, C., Fridman, L., and Levant, A., 2014, "Introduction: Intuitive Theory of Sliding Mode Control," *Sliding Mode Control and Observation*, Springer, New York, pp. 1–42.
[22] Aghababa, M., 2018, "Stabilization of Canonical Systems Via Adaptive Chattering Free Sliding Modes With No Singularity Problems," IEEE Trans. Syst., Man, Cybern.: Syst. (in press).
[23] Shaer, B., Kenné, J.-P., Kaddissi, C., and Fallaha, C., 2018, "A Chattering-Free Fuzzy Hybrid Sliding Mode Control of an Electrohydraulic Active Suspension," Trans. Inst. Meas. Control, **40**(1), pp. 222–238.
[24] Levent, A., 2005, "Quasi-Continuous High-Order Sliding-Mode Controllers," IEEE Trans. Autom. Control, **50**(11), pp. 1812–1816.
[25] Levant, A., 1993, "Sliding Order and Sliding Accuracy in Sliding Mode Control," Int. J. Control, **58**(6), pp. 1247–1263.
[26] Levant, A., 2003, "Higher-Order Sliding Modes, Differentiation and Output-Feedback Control," Int. J. Control, **76**(9–10), pp. 924–941.
[27] Siciliano, B., Sciavicco, L., Villani, L., and Oriolo, G., 2009, *Robotics: Modelling, Planning and Control*, Springer-Verlag, London.
[28] Muñoz Benavent, P., Gracia, L., Solanes, J., Esparza, A., and Tornero, J., 2018, "Sliding Mode Control for Robust and Smooth Reference Tracking in Robot Visual Servoing," Int. J. Robust Nonlinear Control, **28**(5), pp. 1728–1756.
[29] Huo, L., and Baron, L., 2008, "The Joint-Limits and Singularity Avoidance in Robotic Welding," Ind. Rob.: An Int. J., **35**(5), pp. 456–464.
[30] Antonelli, G., 2009, "Stability Analysis for Prioritized Closed-Loop Inverse Kinematic Algorithms for Redundant Robotic Systems," IEEE Trans. Rob., **25**(5), pp. 985–994.
[31] Golub, G., and Van Loan, C., 1996, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, MD.