

Elsevier required licence: © <2018>. This manuscript version is made available under the CC-BY-NC-ND 4.0 license <http://creativecommons.org/licenses/by-nc-nd/4.0/>
The definitive publisher version is available online at <https://doi.org/10.1016/j.mechatronics.2018.04.008>

Adaptive Sliding Mode Control for Robotic Surface Treatment Using Force Feedback

Luis Gracia^{*a}, J. Ernesto Solanes^a, Pau Muñoz-Benavent^a,
Jaime Valls Miro^c, Carlos Perez-Vidal^b, Josep Tornero^a

^a*Instituto de Diseño y Fabricación (IDF), Universitat Politècnica de València, Camino de Vera s/n, 46022 Valencia, Spain (e-mail: luigraca@isa.upv.es). *Corresponding author*

^b*Departamento de Ingeniería de Sistemas y Automática (DISA), Universidad Miguel Hernández, Avda de la Universidad s/n, 03202 Elche, Spain*

^c*Centre for Autonomous Systems (CAS), Faculty of Engineering, University of Technology Sydney (UTS), NSW 2007 Sydney, Australia.*

Abstract

This work presents a hybrid position-force control of robots in order to apply surface treatments such as polishing, grinding, finishing, deburring, etc. The robot force control is designed using sliding mode concepts to benefit from robustness. In particular, the sliding mode force task is defined using equality constraints to attain the desired tool pressure on the surface, as well as to keep the tool orientation perpendicular to the surface. In order to deal with sudden changes in material stiffness, which are ultimately transferred to the polishing tool and can produce instability and compromise polishing performance, several adaptive switching gain laws are considered and compared. Moreover, a lower priority tracking controller is defined to follow the desired reference trajectory on the surface being polished. Hence, deviations from the reference trajectory are allowed if such deviations are required to satisfy the constraints mentioned above. Finally, a third-level task is also considered for the case of redundant robots in order to use the remaining degrees of freedom to keep the manipulator close to the home configuration with safety in mind. The main advantages of the method are increased robustness and low computational cost. The applicability and effectiveness of the proposed approach is substantiated by experimental results using a redundant 7R manipulator: the Rethink Robotics Sawyer collaborative robot.

Keywords: Force control, adaptive switching gain, polishing, finishing.

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.) have 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, 2]. 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 [3, 4]. As such, surface treatment operations and quality control continue to be mainly manual processes being carried out by skilled workers, which give rise to issues such as subjectivity in the evaluation criteria, human errors, etc.

Production quality control can be divided in two phases: the first one is related with the detection of anomalies following an industry-set minimum standard, and the second one is the actual elimination of these anomalies. In the case of surface treatment, industrial solutions have been recently developed for anomaly detections in sectors such as automotive manufacturing [5, 6, 7] and ceramics [8, 9]. However, to the best of the authors' knowledge, the second phase remains a manual mechanism in both industries.

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, etc.), 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 [10]. Many approaches can be found in the literature tackling this problem using robot manipulators with force feedback. For instance, in [2] a method for maintaining a constant polishing pressure with a numerical control (NC) polishing system was proposed by controlling the force during the process. In [11] a hybrid force/position control is developed for a compliant rescue manipulator. In [12] a dual force/position control loop based on fuzzy techniques was presented for robotic grinding applications. In [13] an analytically force overshoot-free approach based on impedance control was developed to per-

form force-tracking. In [10] an algorithm was proposed for planning the tool location together with a compliance force control. In [14] a hybrid impedance control of a robot manipulator is developed for wrist and forearm rehabilitation. In [15], a methodology based on sensor-less force control technique and the quarry matrix was proposed to control the x and y trajectories of the tool pose and the polishing force in the z -direction in a parallel machine. Afterwards, in [16] and [17] this method was improved by using information about trajectories and forces applied by skilled workers. Finally, a multisensory approach was used in [18] gathering information from acoustic emissions, forces and scattered light sensors mounted on the manipulator.

Other robot force control approaches are based on sliding-mode control (SMC) theory [19]. Specifically, in [20] the authors developed a nonlinear position and force controller for a two link flexible robot manipulator interacting with a rigid environment. The controller was designed using a first-order SMC to correct position and force tracking errors. Another example can be found in [21] where SMC was used to suppress impact forces when contacting the objects in the environment and to allow the system to continue with a stable robot motion. In [22] a hybrid position/force control scheme was proposed using first- and second-order SMC for position and force control, respectively. In [23] an impedance control structure was proposed for monitoring the contact force between the robot tool and the objects in the environment, and a model-free fuzzy SMC strategy was employed to design the position and force controllers. In [24], a hybrid force-position control is presented based on a dynamic decoupling performed using SMC. In [25], a tracking and force control of a SCARA robot is presented using SMC. In [26], 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 [27], an adaptive hybrid force/position SMC for flexible joint robot was presented. In [28] first- and second-order SMC was used in order for the robot to reach the goal point even in the presence of unknown obstacles. In [29], a hybrid control scheme was proposed using second-order SMC and contact force estimation.

One typical problem of SMC is related to the controller switching gain. High values of the controller switching gain increase the *chattering* band, which is a well known issue to be solved in SMC techniques [30, 31, 32]. On the contrary, adjusting the switching gain to minimize the chattering band at a certain operating point may cause the control to become unstable for another operating point. For instance, in the specific problem treated in this

paper (the surface treatment using the force sensor feedback) this problem is present due to changes on the stiffness along the surface, e.g., hood or doors on car bodies, textile materials on car seats, etc.

To overcome this problem, Adaptive SMC (ASMC) solutions have been proposed in the literature, i.e., SMC approaches with an adaptive switching gain (ASG). For instance, in [33] an ASMC was developed using an integral/exponential adaptation law with boundary-layer in order to reduce the switching gain overestimation while simultaneously speeding up the system response to the uncertainties. In [34] two fault-tolerant control schemes for spacecraft attitude stabilization with external disturbances were proposed, where a fault-tolerant SMC was incorporated with an adaptive technique to accommodate actuator faults in order to relax the required boundary information. In [35] a high-order ASMC was proposed based on the concepts of integral sliding mode and real high-order sliding mode detector. In [36] the over-adaptation problem in ASMC for rigid spacecraft attitude maneuvers was investigated. For more SMCs with ASG solutions, the reader is referred to [37, 38, 39], among others.

The problem of robot surface treatment is addressed in this work using a hybrid force/position approach. In particular, force control is performed using SMC in order to benefit from robustness, whereas position control is performed using a conventional continuous tracking controller. The proposal presents several distinctive features and innovations:

1. Instead of the classical compliance vector [40], 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.
2. SMC is used to satisfy the equality constraints required to undertake the surface treatment, i.e., to attain the desired tool pressure on the surface and to keep the tool orientation perpendicular to the surface.
3. Several ASG algorithms are evaluated and compared to determine their performance.
4. A third low-priority task is also considered for the case of redundant robots (such as the one used in the experiments of this work) where the remaining degrees of freedom of the robot are exploited to keep it as close as possible to the home configuration for increased safety.

The paper is organized as follows. Next section introduces some preliminaries, while Section 3 presents the SMC used in this work. The proposed

approach for robotic polishing is presented in Section 4, while some important remarks about the method are given in Section 4.4. A simulation is presented in Section 5 in order to evaluate and compare several ASG laws. The implementation of the proposed robot controller is detailed in Section 7. The feasibility and robustness of the proposed approach is substantiated by experimental results in Section 8 using a redundant 7R manipulator: the Sawyer collaborative robot. Finally, some conclusions are given.

Nomenclature

a_i	i -component of vector \mathbf{a}	\mathbf{K}_c	Approaching parameter matrix
$\bar{\mathbf{a}}$	DH parameters (translation in X -axis)	$K_{c,m}$	Motor approaching parameter
\mathbf{A}_i	Matrix for the i -th task	K_{int}	Speed adaptation parameter of continuous ASG methods
\mathbf{b}_i	Vector for the i -th task	$K_{m,e}$	Motor electromotive force constant
b_m	Motor viscous friction	$K_{m,t}$	Motor torque constant
\mathbf{d}	Unmeasured disturbance	\mathbf{K}_s	Stiffness coefficient matrix
$\bar{\mathbf{d}}$	DH parameters (translation in Z -axis)	$K_{T,p}$	Position correction gain in Level 2
\mathbf{d}_c	Inaccuracy of the low-level control	$K_{T,v}$	Velocity correction gain in Level 2
\mathbf{e}	Error of the robot tool pose	\mathbf{l}	Robot kinematic function
\mathbf{f}	Vector field	$L_f\phi$	Lie derivative of ϕ in the \mathbf{f} -direction
f_{FC}	Bandwidth of the sensor filter	$\mathbf{L}_g\phi$	Lie derivative of ϕ in the \mathbf{g} -direction
$f_{KC,i}$	Bandwidth of the kinematic control in Level i	L_m	Motor electric inductance
f_{RC}	Bandwidth of the robot controller	M	Number of tasks
f_{SM}	Frequency of the SM control action	N_{eq}	Number of equality constraints
\mathbf{F}	Vector of forces and torques	\mathbf{N}_i	Null-space projection matrix for the first i tasks
$F_{z,ref}$	Desire force in Z -axis	\mathbf{p}	Robot pose
\mathbf{g}	Set of vector fields	$\ddot{\mathbf{p}}_c$	Commanded acceleration for the robot tool pose
i_m	Armature current of the motor	\mathbf{p}_{ref}	Reference pose
\mathbf{J}	Robot Jacobian matrix	\mathbf{p}_s	Surface position and orientation
J_m	Moment of inertia of the motor	\mathbf{q}	Robot configuration
\mathbf{J}_n	Geometric Jacobian relative to the tool coordinate system	$\bar{\mathbf{q}}$	Quaternion vector
$K_{3,p}$	Position correction gain in Level 3	\mathbf{q}_0	Home configuration
$K_{3,v}$	Velocity correction gain in Level 3	$\dot{\mathbf{q}}_c$	Commanded velocity
		$\ddot{\mathbf{q}}_c$	Commanded acceleration

$\ddot{\mathbf{q}}_{c,i}$	Solution of the commanded acceleration for the first i tasks	η	Configuration parameter of continuous ASG methods
R_m	Motor electric resistance	γ	Yaw angle
T_s	Control period	μ	Speed adaptation parameter of discrete ASG method
\mathbf{u}	Control input	ω	Motor angular velocity
u^+	Switching gain	ω_{ref}	Reference for the motor velocity
V_m	Motor input voltage	ϕ	Modified constraint functions
\mathbf{W}	Switching gain weights	ϕ_m	Motor modified constraint function
$\overline{\mathbf{W}}$	Modified switching gain weights	Φ	Allowed space
\mathbf{x}	State vector	ψ	Configuration parameter of continuous ASG methods
α	Roll angle	σ	Original constraint functions
$\overline{\alpha}$	DH parameters (rotation in X -axis)	σ_m	Motor original constraint function
β	Pitch angle	τ	Joint torques
$\Delta\phi$	Chattering band of ϕ	$\overline{\theta}$	DH parameters (rotation in Z -axis)
$\Delta\mathbf{s}$	Deformation of the sensor		

2. Preliminaries

Kinematics. Following the standard notation [41], the robot *pose* \mathbf{p} depends on the robot *configuration* \mathbf{q} as follows:

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

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. Obviously, for stability reasons, the bandwidth of this underlying robot control should be faster than that of the used kinematic control.

Task-priority based redundancy resolution. It is useful to consider the task-priority strategy [41] to tackle several (possibly incompatible) objectives simultaneously assigning an order of priority to each one. Thus, a lower-priority task is satisfied only by using the degrees of freedom in the null space of the higher-priority ones [42]. When an exact solution is not possible for a given task at a particular priority level, its error is minimized. The formulation for this approach is detailed below. Let us consider M tasks which consist on calculating a command vector $\ddot{\mathbf{q}}_c$ (i.e., the commanded joint acceleration vector) in order to fulfill the following acceleration equality constraint:

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

where matrix \mathbf{A}_i and vector \mathbf{b}_i of the i th task are assumed known and index i represents the priority order: $i = 1$ for highest priority and $i = M$ to lowest.

The solution $\ddot{\mathbf{q}}_{c,M}$ that hierarchically minimizes the error of equations in (5) is given by the following recursive formulation, proposed in [43]:

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

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

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

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

Theorem 1. *Consider the dynamic system given by:*

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

where $\mathbf{x}(t)$ is the state vector, $\mathbf{d}(t)$ is an unmeasured disturbance or model uncertainty, $\mathbf{u}(t)$ is the control input vector (possibly discontinuous), \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 $\phi_i(\mathbf{x}) = 0$, $i = 1, \dots, N_{eq}$, where $\phi_i(\mathbf{x})$ is the i th equality constraint function. Thus, the manifold Φ of the state space compatible with the constraints on state \mathbf{x} is given by:

$$\Phi = \{\mathbf{x} \mid \phi_i(\mathbf{x}) = 0\}, \quad (9)$$

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

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

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

$$u^+ > \|\mathbf{L}_f\phi\|_1 / \text{diag}_{\min}(\mathbf{W}), \quad (11)$$

where ϕ is a column vector with all the constraint functions ϕ_i , the scalar $L_f\phi_i = \frac{\partial\phi_i^T}{\partial\mathbf{x}}\mathbf{f}$ and the row vector $\mathbf{L}_{\mathbf{g}}\phi_i = \frac{\partial\phi_i^T}{\partial\mathbf{x}}\mathbf{g}$ denote the Lie derivatives of $\phi_i(\mathbf{x})$ in the direction of vector field \mathbf{f} and in the direction of the set of vector fields \mathbf{g} , respectively, column vector $L_f\phi$ contains the elements $L_f\phi_i$ of all equality constraints, matrix $\mathbf{L}_{\mathbf{g}}\phi$ contains the row vectors $\mathbf{L}_{\mathbf{g}}\phi_i$ of all equality constraints, $\text{sign}(\cdot)$ represents the sign function (typically used in SMC), positive scalar u^+ is the so-called switching gain, which can be either constant or varying in time, \mathbf{W} 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 [44] and its generalization. Details omitted for brevity. \square

3.1. Chattering

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:

$$\Delta\phi = T_s |\mathbf{L}_g \phi \mathbf{u}| = T_s u^+ \text{diag}(\mathbf{W}), \quad (12)$$

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.

3.2. Adaptive sliding mode control

Since the switching gain u^+ can be varying in time, a common option consists in using an *adaptive switching gain* (ASG) in order to minimize its value online and, thus, the control effort and chattering amplitude are reduced. Next, some relevant ASG laws in the literature are detailed:

- Method in [45, 46] (M1):

$$u^+(t) = u^+(t_0) + K_{int} \int_{\tau=t_0}^t \|\phi\| d\tau, \quad (13)$$

where $u^+(t_0)$ is the initial SMC switching gain and K_{int} is a positive configuration parameter that determines the speed of the adaptation. Note that this ASG method is based on the fact that when the system reaches the SM behavior the constraint function ϕ is equal to zero. However, this method has the disadvantage that the switching gain value cannot decrease due to the integral term. In fact, from a practical point of view, the chattering phenomenon mentioned in Section 3.1 makes the switching gain increase indefinitely with this method, even if the chattering band is small.

- Method in [47, 48] (M2):

$$u^+(t) = \left| u^+(t_0) + K_{int} \int_{\tau=t_0}^t (\|\phi\| - \psi) d\tau \right|, \quad (14)$$

where ψ is a positive constant lower than the SM band $\Delta\phi$. Note that this method improves the aforementioned method M1 including the parameter ψ in the integral term in order to allow decreasing the switching gain.

- Method in [37, 33] (M3):

$$u^+(t) = \eta (e^{\|\phi\|} - 1) + K_{int} \int_{\tau=t_0}^t (\|\phi\| - \psi) d\tau, \quad (15)$$

where η is a positive configurable parameter. Note that this method is similar to method M2 but introduces an exponential term that tends to zero as the system tends to the SM surface given by $\phi = 0$.

- Method in [38, 33] (M4):

$$u^+(t) = \eta (e^{\|\phi\|} - 1) + K_{int} \int_{\tau=t_0}^t \|\phi\| \text{sign}(\|\phi\| - \psi) d\tau. \quad (16)$$

Note that this method is similar to method M3 but modifies the integral term using the sign function.

- Discrete method in [49, 50] (M5):

$$u^+(k) = |u^+(k-1) + T_s \mu \text{sign}(\phi(k)) \text{sign}(\phi(k-1))|, \quad (17)$$

where $u^+(k)$ and $u^+(k-1)$ are the values of the switching gain for the current and the previous time steps, respectively, $\phi(k)$ and $\phi(k-1)$ are values of the constraint function for the current and the previous time steps, respectively, and μ is a positive configuration parameter that determines the speed of the adaptation. Since the SMC consists in a first-order control law, the SM surface should be crossed in every successive time step and, hence, the ASG law in (17) increases the switching gain if the SM surface has not been crossed and decreases it otherwise. Thus, the method aims to lead the system to cross the SM surface while minimizing the switching gain. This discrete approach works properly for ideal or simple systems. Nevertheless, its performance is compromised in real complex systems due to the unmodeled dynamics of the system (nonlinearities, delays, noisy signals, etc.), which could make it difficult to cross the SM surface in just one time step. Therefore, it is proposed in this work to “relax” the commutation condition for real systems, i.e., to consider a few time steps (e.g., two or three) instead of just one time step to evaluate if the SM surface has been successfully crossed or not in order to decrease or increase the switching gain.

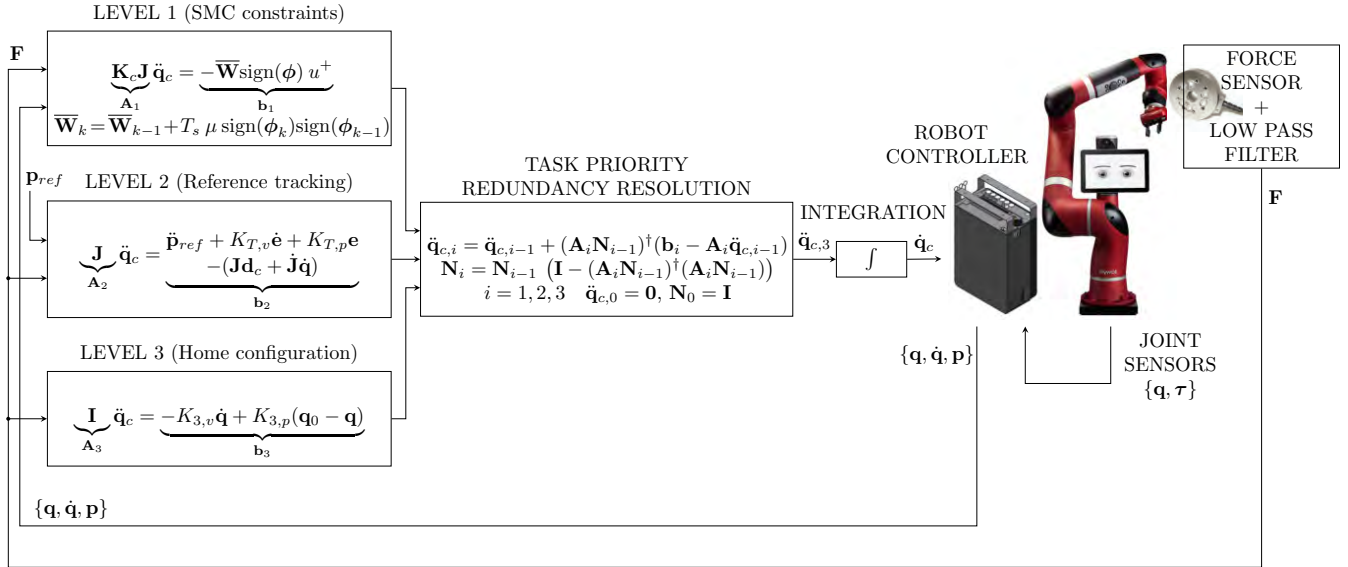


Figure 1. Block diagram of the method.

4. Proposed approach

Fig. 1 shows the block diagram of the proposed method. Three 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 the equality constraints required to accomplish the surface treatment task with the robot tool. Level 2 (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. Level 3 (low-priority task) is considered for the case of redundant robots to keep the robot close to the home configuration.

The input to these levels are: the robot state $\{\mathbf{q}, \dot{\mathbf{q}}\}$ and pose vector \mathbf{p} obtained from the robot controller; the vector \mathbf{F} of forces and torques measured by a sensor located at the robot tool, which has already been filtered by the sensor electronics; and the reference pose \mathbf{p}_{ref} . 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 $\ddot{\mathbf{q}}_{c,3}$, 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 joint sensors.

4.1. Level 1: Constraints

The acceleration equality for Level 1 is obtained below using the SMC presented in Section 3. In particular, the following steps must be followed to apply Theorem 1:

- 1) Define a dynamic system in the form of Eq. (8).
- 2) Define the equality constraints $\phi_i(\mathbf{x}) = 0$, $i = 1, \dots, N_{eq}$ to be satisfied.
- 3) Compute the lie derivatives $\mathbf{L}_g \phi_i$.
- 4) Compute the control action \mathbf{u} from equality (10), where the switching gain u^+ and switching gain weights \mathbf{W} are the control parameters.

For the first step 1), a dynamic system is considered with state vector $\mathbf{x} = [\mathbf{q}^T \quad \dot{\mathbf{q}}^T]^T$, disturbance vector $\mathbf{d} = \mathbf{d}_c$ and input vector $\mathbf{u} = \ddot{\mathbf{q}}_c$. Hence, the model is a double integrator, and from (4) the state equation results in:

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

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

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

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

The remaining steps 2), 3) and 4) are tackled below.

4.1.1. Force model

The constraints required to accomplish the surface treatment are defined below depending on the 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 approximated by the ideal elastic model below [51]:

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

where \mathbf{F} is the force vector relative to the tool coordinate system, \mathbf{K}_s is a diagonal matrix with the stiffness coefficients for each tool axis and vector $\Delta \mathbf{s}$

is the mechanical deformation of the sensor relative to the tool coordinate system, which depends on the robot configuration \mathbf{q} and the position and orientation \mathbf{p}_s of the surface of the environment. Note that, in general, both \mathbf{K}_s and \mathbf{p}_s are variable.

4.1.2. 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 Φ . Therefore, the actual constraint $\sigma_i(\mathbf{q}, t) = 0$ will be modified to include the speed of movement as follows:

$$\phi_i(\mathbf{q}, \dot{\mathbf{q}}, t) = \sigma_i + K_{c,i}\dot{\sigma}_i = 0 \quad (22)$$

where $K_{c,i}$ is a free design parameter that determines the rate of approach to the original constraint $\sigma_i = 0$.

4.1.3. Constraints for the surface treatment

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

$$\sigma_z(\mathbf{q}, t) = [0 \ 0 \ 1 \ 0 \ 0 \ 0]^T \mathbf{F} - F_{z,ref} = 0 \quad (23)$$

$$\sigma_\alpha(\mathbf{q}, t) = [0 \ 0 \ 0 \ 1 \ 0 \ 0]^T \mathbf{F} = 0 \quad (24)$$

$$\sigma_\beta(\mathbf{q}, t) = [0 \ 0 \ 0 \ 0 \ 1 \ 0]^T \mathbf{F} = 0, \quad (25)$$

where the first equality constraint is used to attain the desired force $F_{z,ref}$ between the tool and the surface 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 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 (19) and (21)–(25), the Lie derivative $\mathbf{L}_g\phi$ required

for the SMC in (10) is given by:

$$\begin{aligned} \mathbf{L}_g \boldsymbol{\phi} &= (\partial \boldsymbol{\phi} / \partial \dot{\mathbf{q}})^T = \mathbf{K}_c (\partial \boldsymbol{\sigma} / \partial \mathbf{q})^T \\ &= \mathbf{K}_c \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \mathbf{K}_s \mathbf{J}_n = \mathbf{K}_c \begin{bmatrix} K_{s,z} \\ K_{s,\alpha} \\ K_{s,\beta} \end{bmatrix} \mathbf{J}_n, \end{aligned} \quad (26)$$

where \mathbf{J}_n is the geometric Jacobian relative to the tool coordinate system, $\boldsymbol{\sigma}$ is a column vector composed of all equality constraints σ_i , \mathbf{K}_c is a diagonal matrix composed of all parameters $K_{c,i}$ and $\{K_{s,z}, K_{s,\alpha}, K_{s,\beta}\}$ are the stiffness coefficients for the linear Z -axis and the rotational X - and Y -axis, respectively.

4.1.4. Acceleration equality for Level 1

Since the stiffness coefficients \mathbf{K}_s in $\mathbf{L}_g \boldsymbol{\phi}$ (26) may not be known, they can be included without loss of generality in the switching gain weight matrix $\overline{\mathbf{W}}$, so that the SMC given by (10) is modified as follows:

$$\mathbf{K}_c \mathbf{J}_n \ddot{\mathbf{q}}_c = -\overline{\mathbf{W}} \text{sign}(\boldsymbol{\phi}) u^+ \rightarrow \mathbf{A}_1 \ddot{\mathbf{q}}_c = \mathbf{b}_1, \quad (27)$$

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

$$\overline{\mathbf{W}} = \begin{bmatrix} \overline{W}_z & 0 & 0 \\ 0 & \overline{W}_\alpha & 0 \\ 0 & 0 & \overline{W}_\beta \end{bmatrix} = \begin{bmatrix} W_z/K_{s,z} & 0 & 0 \\ 0 & W_\alpha/K_{s,\alpha} & 0 \\ 0 & 0 & W_\beta/K_{s,\beta} \end{bmatrix}. \quad (28)$$

Note that the SMC given by (27) only requires: the control parameters $\{u^+, \overline{W}_i, K_{c,i}\}$; the geometric Jacobian \mathbf{J}_n ; and the constraint functions $\{\phi_z, \phi_\alpha, \phi_\beta\}$, which are computed from the force sensor measurements $\{F_z, F_\alpha, F_\beta, F_x, F_y\}$ and their first-order time derivatives.

4.2. Level 2: Reference tracking

This work considers the classical operational space robot control [51] that, taking into account Eqs. (3) and (4), results in:

$$\mathbf{J} \ddot{\mathbf{q}}_c = \ddot{\mathbf{p}}_c - (\mathbf{J} \mathbf{d}_c + \dot{\mathbf{J}} \dot{\mathbf{q}}), \quad (29)$$

where $\ddot{\mathbf{p}}_c$ is the commanded acceleration for the robot tool pose and \mathbf{d}_c is typically set to zero since it represents an unknown disturbance.

Moreover, considering the classical acceleration-based kinematic controller used for trajectory tracking [52], i.e., a correction based on the position and velocity errors plus a feedforward of the second-order derivative of the reference, the commanded acceleration $\ddot{\mathbf{p}}_c$ results in:

$$\ddot{\mathbf{p}}_c = \ddot{\mathbf{p}}_{ref} + K_{T,v}\dot{\mathbf{e}} + K_{T,p}\mathbf{e}, \quad (30)$$

where \mathbf{e} is the error of the robot tool pose, i.e., $\mathbf{e} = \mathbf{p}_{ref} - \mathbf{p}$, and $K_{T,p}$ and $K_{T,v}$ are the correction gains for the position and velocity errors, respectively.

By comparing the acceleration equality (29) for Level 2 with equation (5) it is obtained that $\mathbf{A}_2 = \mathbf{J}$ and $\mathbf{b}_2 = \ddot{\mathbf{p}}_c - \dot{\mathbf{J}}\dot{\mathbf{q}}$.

4.3. Level 3: Home configuration

This level is considered only for the case of *redundant* robots (such as the one used to experimentally validate the proposed work, presented in Section 8) since otherwise there are no remaining degrees of freedom at this level. Therefore, the inclusion of this level allows to avoid an uncontrolled self-motion of the redundant robot. While there are various options described in the literature [53, 54], 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:

$$\ddot{\mathbf{q}}_c = -K_{3,v}\dot{\mathbf{q}} + K_{3,p}(\mathbf{q}_0 - \mathbf{q}), \quad (31)$$

where $K_{3,v}$ and $K_{3,p}$ are the gains used for the velocity and position corrections, respectively.

By comparing the acceleration equality (31) for Level 3 with equation (5) it is obtained that $\mathbf{A}_3 = \mathbf{I}$ and $\mathbf{b}_3 = -K_{3,v}\dot{\mathbf{q}} + K_{3,p}(\mathbf{q}_0 - \mathbf{q})$.

4.4. 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 in the high-priority Level 1. If the actual control action are the joint velocities (or positions), a single (or double) integrator can be applied to the discontinuous control signal to compute the actual continuous control action, see Fig. 1.

It is worth mentioning that if the robot can be commanded using joint torques, which is not the case of most industrial robots, the proposed approach can be adapted for robot dynamic control computing the joint torques from the commanded joint accelerations using the inverse dynamic model of the robot. However, from a practical point of view, it could be advisable to use the kinematic control to take advantage of the low-level joint control provided by the robot manufacturer, which is typically designed (and fully optimized) taking into account a more complete robot dynamic model, including frictions, dead-zones, etc.

Stability. The stability of the SMC in Level 1 is guaranteed if u^+ fullfills (11) and matrix $\mathbf{L}_g\phi$ is full row rank. That is, taking into account (27), the row rank of the robot Jacobian has to be equal to the number N_{eq} of 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 Level 2 and Level 3, which represent classical kinematic and inverse kinematic control algorithms, the reader is referred to [55], where the stability of this kind of algorithms is analyzed in a task prioritization framework. It is also worth mentioning that the limitation of degrees of freedom for the tracking controller is the common situation in hybrid position/force control, e.g., see the classical approach in [40].

Time derivatives. The method requires the first-order time derivatives of \mathbf{F} and \mathbf{J} for the SMC in Level 1 and the tracking controller in Level 2, respectively. 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, although the filter bandwidth must not limit the bandwidth of the control law, see Section 6.1.

4.5. Advantages of the proposed method

An advantage of the proposed approach is *complementarity*: the SMC is in charge of controlling the tool force and orientation, whereas the lower priority classical controller is in charge of tracking the reference trajectory. Other advantages of the proposed SMC to satisfy the constraints are:

- *Smoothness*: firstly, the joint velocities are continuous since the SM control action are the joint accelerations; and, secondly, the constraints

manifold Φ is reached progressively depending on $K_{c,i}$, i.e., the velocity perpendicular to the manifold Φ is progressively reduced to zero.

- *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 pose \mathbf{p}_s (position and orientation) of the surface of the environment and its derivative; the time derivative of the Jacobian matrix $\dot{\mathbf{J}}$; the joint velocities $\dot{\mathbf{q}}$; etc.
- *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 (27) only requires the robot Jacobian and the constraint functions ϕ_i , which are computed from the force sensor measurement \mathbf{F} and its time derivative. Hence, the method only requires a few program lines and has reduced computation time, see Section 7.

5. Simulation

In this section, the SMC described in Section 3 together with the ASG laws described in Section 3.2 are simulated for comparative purposes using the simple DC motor model given by:

$$\dot{\omega} = (K_{m,t} i_m - b_m \omega) / J_m \quad (32)$$

$$\dot{i}_m = (-R_m i_m + V_m - K_{m,e} \omega) / L_m, \quad (33)$$

where ω 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 ω and the input or control action the voltage V_m . In order to track the reference velocity ω_{ref} the original and modified constraints for the SMC are defined as follows:

$$\sigma_m(\omega) = \text{Error} = \omega_{ref} - \omega = 0 \quad (34)$$

$$\phi_m(\omega, \dot{\omega}) = \sigma_m + K_{c,m} \dot{\sigma}_m = 0, \quad (35)$$

where $K_{c,m}$ is the approaching parameter to the original constraint. Note that the relative degree between the constraint function $\phi_m(\omega, \dot{\omega})$ and the control action V_m is equal to one, as required by SM control theory.

5.1. Simulation conditions and parameter values

The simulation was run under the following conditions:

- Parameters used for the DC motor model: $J_m(t_0) = 0.1 \text{ kg.m}^2$, $b_m = 0.1 \text{ N.m.s}$, $K_{m,e} = 0.01 \text{ V.s}$, $K_{m,t} = 0.01 \text{ N.m/A}$, $R_m = 1 \text{ } \Omega$ and $L_m = 0.5 \text{ H}$.
- The approaching parameter to the original constraint is $K_{c,m} = 0.1$.
- The parameters used for each switching gain law are:
 - FSG (fixed switching gain): $u^+ = 100$
 - M1: $u^+(t_0) = 100$, $K_{int} = 500$
 - M2: $u^+(t_0) = 100$, $K_{int} = 500$, $\psi = 0.01$
 - M3: $\eta = 160$, $K_{int} = 1000$, $\psi = 0.01$
 - M4: $\eta = 160$, $K_{int} = 1000$, $\psi = 0.01$
 - M5: $u^+(k_0) = 100$, $\mu = 500$.
- The simulation period T_s was set to one millisecond

5.2. Simulation results

The simulation results presented below were obtained using MATLAB[®].

A first simulation has been conducted with a constant reference velocity ω_{ref} of 0.5 rad/s and a variable moment of inertia J_m for the motor (see the bottom plot in Fig. 2) in order to emulate a system with changing conditions: for the interval 0–2s the value of J_m is constant and equal to 0.1 kg.m²; for the interval 2s–3s the value of J_m progressively increases to 0.3 kg.m²; subsequently, for the interval 3s–5s the value of J_m is constant and equal to 0.3 kg.m²; furthermore, for the interval 5s–6s the value of J_m progressively decreases to 0.033 kg.m²; and, finally, for the interval 6s–8s the value of J_m is constant and equal to 0.033 kg.m². Fig. 2 shows that the FSG case does not work properly when the value of the moment of inertia J_m is large, see the error signal in the top plot. Besides, once the initial error has been

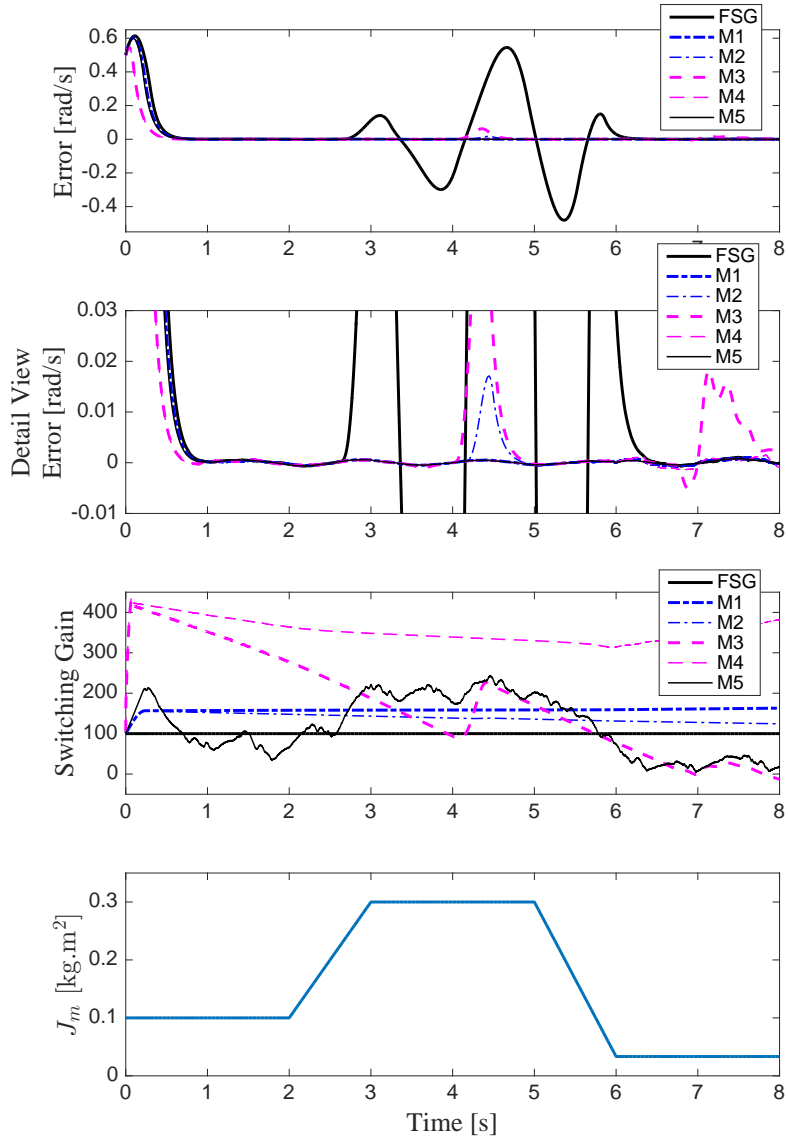


Figure 2. Simulation with changes in the moment of motor's inertia J_m : comparison of FSG and ASG methods in terms of error and control switching gain.

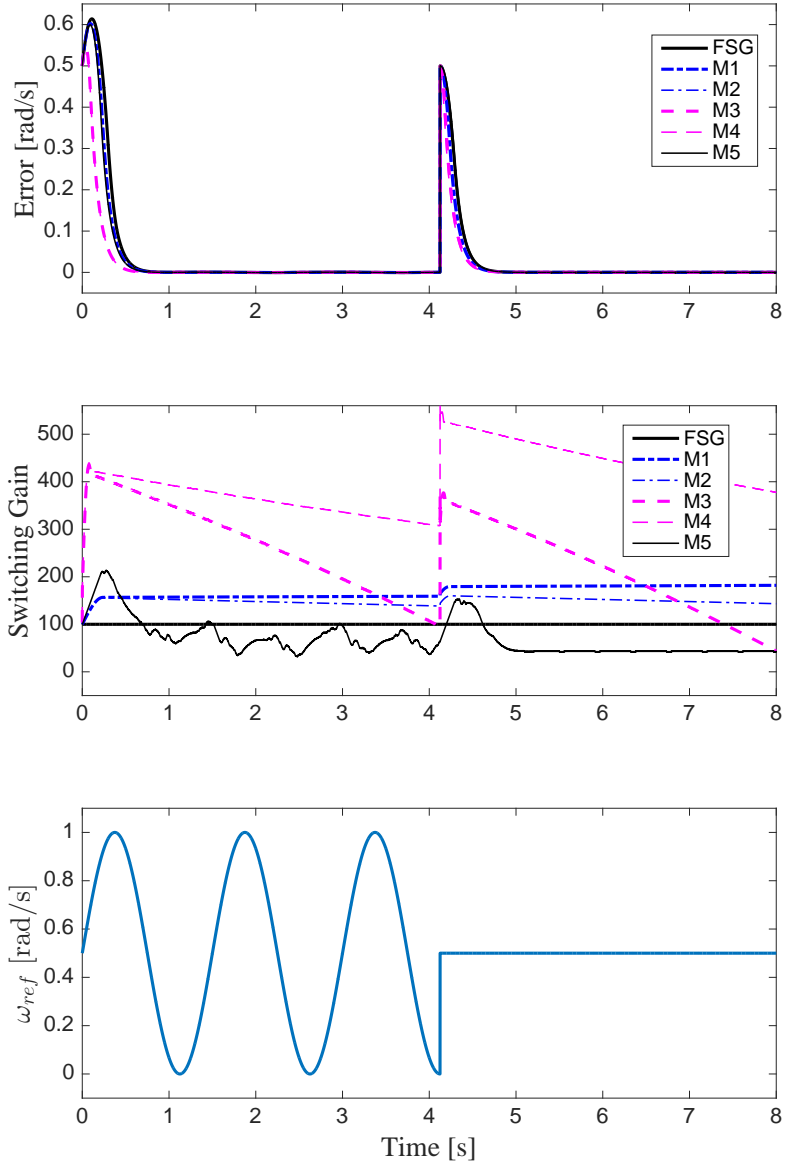


Figure 3. Simulation with constant motor parameters and sinusoidal and constant reference velocity: comparison of the FSG and ASG methods.

corrected, among the ASG methods only methods M1, M4 and M5 maintain the zero error, see the detail view in the second plot, which means that they operate in SM during all the simulation (except for a short time at the beginning of the simulation, while the initial error is being corrected), as required. However, only the discrete method M5 properly modifies the switching gain according to each part of the simulation (see the third and fourth plots): a medium value for $J_m = 0.1$, a large value for $J_m = 0.3$ and a small value for $J_m = 0.033$. Furthermore, the chattering band for the discrete method M5 is significantly smaller than that for methods M1 and M4 since the switching gain obtained by method M5 is smaller, see the third plot. Therefore, although methods M1 and M4 properly operate in SM, their ASG laws clearly overestimate the switching gain, which is not advisable.

Another simulation has been considered with constant motor parameters a reference velocity ω_{ref} consisting of two segments (see the bottom plot in Fig. 3): for the first segment the reference is a sinusoidal wave $\omega_{ref}(t) = 0.5 + 0.5 \sin(1.33\pi t)$ rad/s, whereas for the second segment the reference velocity is a constant value of 0.5 rad/s. As before, once the initial error of both segments has been corrected, only the discrete method M5 properly modifies the switching gain according to the reference velocity (see the middle and bottom plots in Fig. 3): for the sinusoidal segment the switching gain is approximately a periodic signal with the same period of the reference signal; whereas for the constant segment the switching gain remains constant. Note that despite that the system conditions are constant for the second segment, among the ASG methods only the discrete method M5 properly reaches a low constant value for the switching gain.

Therefore, according to the above simulation analysis it can be concluded that the discrete ASG method M5 has the best performance, since it adapts promptly and effectively the switching gain in order to reduce the control effort and the chattering amplitude. Hence, this method is used for the real experimentation in Section 8.

6. Procedure to design the control parameters

6.1. Study of system frequencies

Next, the various signal rates and bandwidths involved in the design of the proposed control method are discussed. Fig. 4 shows a frequency diagram with the following parameters: $f_{SM} = (2T_s)^{-1}$ is the frequency of the SM control action; f_{FC} is the bandwidth or cutoff frequency of the sensor filter,

which is implemented in the sensor electronics; f_{RC} is the bandwidth of the robot controller, imposed by design by the robot manufacturer; and $f_{KC,i}$ is the bandwidth of the kinematic control performed at Level i , which is given by its poles. In particular, the pole of the kinematic control in Level 1, which defines the approaching speed to the original constraint $\sigma_i = 0$, see (22), is given by the inverse of the approaching parameter $K_{c,i}$. The poles of the kinematic control in Level 2, which are used for reference tracking, are given by the roots of the polynomial with coefficients $[1 K_{T,v} K_{T,p}]$. Finally, the poles of the kinematic control in Level 3, which are used to avoid an uncontrolled self-motion for the case of a redundant robot, are given by the roots of the polynomial with coefficients $[1 K_{3,v} K_{3,p}]$.

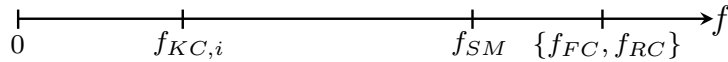


Figure 4. Frequency diagram.

As shown in Fig. 4, the frequency f_{SM} of the SM control action has to be lower than the filter cutoff frequency f_{FC} , i.e., the filter attenuation at this frequency should be relatively small. Similarly, the frequency f_{SM} must also be lower than the bandwidth f_{RC} of the robot controller, otherwise changes in the SM control action would not be properly “followed”. Furthermore, the bandwidth $f_{KC,i}$ of the kinematic control should be significantly lower than the SM frequency f_{SM} for stability reasons.

1 st Step	2 nd Step	3 rd Step			4 th Step	5 th Step
f_{FC}	$f_{SM} (T_s)$	$K_{c,i}$	$\{K_{T,v}, K_{T,p}\}$	$\{K_{3,v}, K_{3,p}\}$	$F_{z,ref}$	$\{u^+, W_i, \mu\}$

Table 1. Steps to design the control parameters.

6.2. Practical guidelines for selecting the parameter values

Table 1 summarizes the steps to design the control parameters, whereas the guidelines to select them are detailed as follows:

- 1st) The bandwidth f_{FC} of the sensor filter is selected as high as possible but guaranteeing that the measurement noise is effectively negated in practical terms.

- 2nd) The control sampling time T_s is selected as low as possible but guaranteeing that the SM frequency $f_{SM} = (2T_s)^{-1}$ is lower than the bandwidths of the sensor filter f_{FC} and robot controller f_{RC} .
- 3rda) The approaching parameter $K_{c,i}$ should be as low as possible to reduced the chattering of the original constraint function σ_i , see (22), but it also has to guarantee that the bandwidth $f_{KC,1} = K_{c,i}^{-1}$ is significantly lower than the SM frequency f_{SM} . The proposed relationship $f_{KC,1} \approx f_{SM}/4$ has been used in the work hereby presented with successful results.
- 3rdb) The parameter $K_{T,v}$ is chosen to obtain a reference tracking with fast overdamped response. In particular, since the critically damped response is given by $K_{T,v,crit} = 2\sqrt{K_{T,p}}$, it is proposed to use $K_{T,v} \approx 3\sqrt{K_{T,p}}$. Moreover, the parameter $K_{T,p}$ is chosen to guarantee that the bandwidth of the reference tracking, which is given by its fastest pole, fulfills the relationship $f_{KC,2} \approx f_{SM}/6$, since successful experimental results have been obtained using this relationship.
- 3rdc) It is proposed to use $K_{3,v} \approx 3\sqrt{K_{3,p}}$ to obtain a fast overdamped response. Since Level 3 is used to control robot self-motion, its dynamics can be relatively slow without degrading robot performance. Hence, the parameter $K_{T,p}$ is selected to fulfill the relationship $f_{KC,2} \approx f_{SM}/10$.
- 4th) The reference force $F_{z,ref}$ is established depending on the requirements of the actual surface treatment task. Some examples are given below in Section 8 for the practical scenarios presented.
- 5th) The switching gain u^+ and switching gain weights \bar{W}_i are empirically tuned as small as possible to reduce the chattering effect whilst guaranteeing that the SM behavior of the control action remains effective in pursuing a desired reference trajectory for the task at hand. Moreover, the speed adaptation parameter μ for the discrete ASG law is tuned as high as possible whilst ensuring low oscillation behavior for the switching gain.

7. Controller implementation

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

- Expression $\mathbf{A}[i, j]$ is used to access the element of matrix \mathbf{A} at row i and column j ; whereas expression $\mathbf{A}[i_1 : i_2, j_1 : j_2]$ is used to access the submatrix of \mathbf{A} from the i_1 -th row to the i_2 -th row and from the j_1 -th column to the j_2 -th column
- Symbol \circ denotes the element-wise or Hadamard product.
- Function $length(\mathbf{x})$ returns the number of elements of vector \mathbf{x} .
- Function $atan2(y, x)$ computes the four-quadrant arctangent of y/x .
- Moore-Penrose pseudoinverse $(\cdot)^\dagger$, see Section 2, which is computed via the singular value decomposition (SVD) method [56] and using a tolerance to set to zero the very small singular values in order to avoid extremely large values for the commanded accelerations. In particular, the singular values whose modulus is less than a hundredth of the magnitude of the maximum singular value are set to zero.
- Function *GetRobotStateAndPoseFromController()* returns the current robot state $\{\mathbf{q}, \dot{\mathbf{q}}\}$ and end-effector pose \mathbf{p} (orientation is represented by the robot controller in quaternions $\bar{\mathbf{q}}$ and, hence, function *ConvertQuaternionsToRPY* below is used to convert them to the roll-pitch-yaw angles of the pose \mathbf{p}).
- Function *GetForceVector()* returns the current force \mathbf{F} , which has already been filtered by the sensor electronics.
- Function *SendToJointControllers($\dot{\mathbf{q}}_c$)* sends the current commanded joint velocity vector to the joint controllers.

The computation time per iteration of the algorithm in a computer with Intel Core i7-4710HQ processor at 2.5 GHz clock frequency using compiled C code was around 15 microseconds for the experiment in Section 8.

8. Real experimentation

The setup used for the experiments consists of (Fig. 5): 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 43x43x10mm attached to the sensor; and an object with flat flexible surface as target.

Algorithm executed at sampling time of T_s seconds

```

1  $[\mathbf{q}, \dot{\mathbf{q}}, \mathbf{p}] = \text{GetRobotStateAndPoseFromController}();$ 
2  $\mathbf{F} = \text{GetForceVector}();$ 
3  $[\mathbf{J}, \mathbf{J}_n] = \text{GetRobotJacobian}(\mathbf{q}, \mathbf{p});$ 
4  $\dot{\mathbf{p}}_{ref} = (\mathbf{p}_{ref} - \mathbf{p}_{ref,prev})/T_s ;$  // Derivative
5  $\ddot{\mathbf{p}}_{ref} = (\dot{\mathbf{p}}_{ref} - \dot{\mathbf{p}}_{ref,prev})/T_s ;$  // Derivative
6  $\dot{\mathbf{J}} = (\mathbf{J} - \mathbf{J}_{prev})/T_s ;$  // Derivative
7  $\dot{\mathbf{F}} = (\mathbf{F} - \mathbf{F}_{prev})/T_s ;$  // Derivative
8  $\dot{\mathbf{p}} = \mathbf{J}\dot{\mathbf{q}} ;$  // Eq. (2)
9  $\ddot{\mathbf{p}}_c = \ddot{\mathbf{p}}_{ref} + K_{T,v}(\dot{\mathbf{p}} - \dot{\mathbf{p}}_{ref}) + K_{T,p}(\mathbf{p} - \mathbf{p}_{ref}) ;$  // Eq. (30)
10  $\phi = \begin{bmatrix} F_z - F_{z,ref} + K_{c,z}\dot{F}_z \\ F_\alpha + K_{c,\alpha}\dot{F}_\alpha \\ F_\beta + K_{c,\beta}\dot{F}_\beta \end{bmatrix} ;$  // Eqs. (22)-(25)
11  $\overline{\mathbf{W}} = |\overline{\mathbf{W}}_{prev} + T_s \mu \text{sign}(\phi) \circ \text{sign}(\phi_{prev})| ;$  // Eq. (17)
12  $\mathbf{A}_1 = \mathbf{K}_c \mathbf{J}_n ;$  // Eq. (27)
13  $\mathbf{b}_1 = -\overline{\mathbf{W}} \text{sign}(\phi) u^+ ;$  // Eq. (27)
14  $\mathbf{A}_2 = \mathbf{J} ;$  // Eq. (29)
15  $\mathbf{b}_2 = \ddot{\mathbf{p}}_c - \dot{\mathbf{J}}\dot{\mathbf{q}} ;$  // Eq. (29)
16  $\mathbf{A}_3 = \mathbf{I} ;$  // Eq. (31)
17  $\mathbf{b}_3 = -K_{3,v}\dot{\mathbf{q}} + K_{3,p}(\mathbf{q}_0 - \mathbf{q}) ;$  // Eq. (31)
18  $\ddot{\mathbf{q}}_{c,1} = \mathbf{A}_1^\dagger \mathbf{b}_1 ;$  // Eq. (6),  $i = 1$ 
19  $\mathbf{N}_1 = \mathbf{I} - \mathbf{A}_1^\dagger \mathbf{A}_1 ;$  // Eq. (7),  $i = 1$ 
20  $\ddot{\mathbf{q}}_{c,2} = \ddot{\mathbf{q}}_{c,1} + (\mathbf{A}_2 \mathbf{N}_1)^\dagger (\mathbf{b}_2 - \mathbf{A}_2 \ddot{\mathbf{q}}_{c,1}) ;$  // Eq. (6),  $i = 2$ 
21  $\mathbf{N}_2 = \mathbf{N}_1 (\mathbf{I} - (\mathbf{A}_2 \mathbf{N}_1)^\dagger (\mathbf{A}_2 \mathbf{N}_1)) ;$  // Eq. (7),  $i = 2$ 
22  $\ddot{\mathbf{q}}_{c,3} = \ddot{\mathbf{q}}_{c,2} + (\mathbf{A}_3 \mathbf{N}_2)^\dagger (\mathbf{b}_3 - \mathbf{A}_3 \ddot{\mathbf{q}}_{c,2}) ;$  // Eq. (6),  $i = 3$ 
23  $\dot{\mathbf{q}}_c = T_s \ddot{\mathbf{q}}_{c,3} + \dot{\mathbf{q}}_{c,prev} ;$  // Integration
24  $\text{SendToJointControllers}(\dot{\mathbf{q}}_c);$ 
25  $\dot{\mathbf{q}}_{c,prev} = \dot{\mathbf{q}}_c ;$  // For next iteration
26  $\mathbf{p}_{ref,prev} = \mathbf{p}_{ref} ;$  // For next iteration
27  $\dot{\mathbf{p}}_{ref,prev} = \dot{\mathbf{p}}_{ref} ;$  // For next iteration
28  $\mathbf{J}_{prev} = \mathbf{J} ;$  // For next iteration
29  $\mathbf{F}_{prev} = \mathbf{F} ;$  // For next iteration
30  $\phi_{prev} = \phi ;$  // For next iteration
31  $\overline{\mathbf{W}}_{prev} = \overline{\mathbf{W}} ;$  // For next iteration

```

Computation of the robot Jacobian

```

1 Function GetRobotJacobian(q, p):
2    $\mathbf{Z} = \mathbf{0} \in \mathcal{R}^{3 \times 3}$ ,  $\mathbf{I} = \text{diag}(\mathbf{1}) \in \mathcal{R}^{3 \times 3}$ ,  $\mathbf{U} = \text{diag}(\mathbf{1}) \in \mathcal{R}^{4 \times 4}$ ,
    $N = \text{length}(\mathbf{q})$ ,  $\mathbf{J}_n = \mathbf{0} \in \mathcal{R}^{6 \times N}$ ;
3    $[\bar{\boldsymbol{\theta}}, \bar{\mathbf{d}}, \bar{\mathbf{a}}, \bar{\boldsymbol{\alpha}}] = \text{GetDHPParameters}(\mathbf{q})$ ;
4   for  $i = N$ ;  $i \geq 0$ ;  $i = i - 1$  do
5      $\mathbf{U} = \begin{bmatrix} \cos \bar{\theta}_i & -\sin \bar{\theta}_i \cos \bar{\alpha}_i & \sin \bar{\theta}_i \sin \bar{\alpha}_i & \bar{a}_i \cos \bar{\theta}_i \\ \sin \bar{\theta}_i & \cos \bar{\theta}_i \cos \bar{\alpha}_i & -\cos \bar{\theta}_i \sin \bar{\alpha}_i & \bar{a}_i \sin \bar{\theta}_i \\ 0 & \sin \bar{\alpha}_i & \cos \bar{\alpha}_i & \bar{d}_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{U}$ ;
6      $\mathbf{J}_n[1 : 6, i] = \begin{bmatrix} -\mathbf{U}[1, 1]\mathbf{U}[2, 3] + \mathbf{U}[2, 1]\mathbf{U}[1, 4] \\ -\mathbf{U}[1, 2]\mathbf{U}[2, 4] + \mathbf{U}[2, 2]\mathbf{U}[1, 4] \\ -\mathbf{U}[1, 3]\mathbf{U}[2, 4] + \mathbf{U}[2, 3]\mathbf{U}[1, 4] \\ \mathbf{U}[3, 1] \\ \mathbf{U}[3, 2] \\ \mathbf{U}[3, 3] \end{bmatrix}$ ;
7   end
8    $\mathbf{M} = \begin{bmatrix} 1 & 0 & \sin p_5 \\ 0 & \cos p_4 & -\sin p_4 \cos p_5 \\ 0 & \sin p_4 & \cos p_4 \cos p_5 \end{bmatrix}$ ;
9    $\mathbf{T} = \begin{bmatrix} \mathbf{I} & \mathbf{Z} \\ \mathbf{Z} & \mathbf{M}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{U}[1 : 3, 1 : 3] & \mathbf{Z} \\ \mathbf{Z} & \mathbf{U}[1 : 3, 1 : 3] \end{bmatrix}$ ;
10   $\mathbf{J} = \mathbf{T} \mathbf{J}_n$ ;
11  return  $[\mathbf{J}, \mathbf{J}_n]$ 

```

Computation of the robot DH parameters

```

1 Function GetDHPParameters (q):
2    $[\bar{\boldsymbol{\theta}}, \bar{\mathbf{d}}, \bar{\mathbf{a}}, \bar{\boldsymbol{\alpha}}] = \begin{bmatrix} q_1 & 0.3159 & 0.081 & -\pi/2 \\ q_2 + \pi/2 & 0.1925 & 0 & \pi/2 \\ q_3 + \pi & 0.4 & 0 & \pi/2 \\ q_4 & -0.1685 & 0 & -\pi/2 \\ q_5 + \pi & 0.4 & 0 & -\pi/2 \\ q_6 & 0.1363 & 0 & \pi/2 \\ q_7 - 1.7449 & 0.1337 & 0 & 0 \end{bmatrix}$ ;
3   return  $[\bar{\boldsymbol{\theta}}, \bar{\mathbf{d}}, \bar{\mathbf{a}}, \bar{\boldsymbol{\alpha}}]$ 

```

Conversion from quaternions to roll-pitch-yaw angles

```

1 Function ConvertQuaternionsToRPY( $\bar{\mathbf{q}}$ ):
2    $\mathbf{M} = \begin{bmatrix} 1 - 2(\bar{q}_2^2 + \bar{q}_3^2) & 2(\bar{q}_1\bar{q}_2 - \bar{q}_4\bar{q}_3) & 2(\bar{q}_1\bar{q}_3 + \bar{q}_4\bar{q}_2) \\ 2(\bar{q}_1\bar{q}_2 + \bar{q}_4\bar{q}_3) & 1 - 2(\bar{q}_1^2 + \bar{q}_3^2) & 2(\bar{q}_2\bar{q}_3 - \bar{q}_4\bar{q}_1) \\ 2(\bar{q}_1\bar{q}_3 - \bar{q}_4\bar{q}_2) & 2(\bar{q}_2\bar{q}_3 + \bar{q}_4\bar{q}_1) & 1 - 2(\bar{q}_1^2 + \bar{q}_2^2) \end{bmatrix};$ 
3   if  $|\mathbf{M}[3, 3]| < 10^{-6}$  and  $|\mathbf{M}[2, 3]| < 10^{-6}$  then
4      $\alpha = 0;$ 
5      $\beta = \text{atan2}(\mathbf{M}[1, 3], \mathbf{M}[3, 3]);$ 
6      $\gamma = \text{atan2}(\mathbf{M}[2, 1], \mathbf{M}[2, 2]);$ 
7   end
8   else
9      $\alpha = \text{atan2}(-\mathbf{M}[2, 3], \mathbf{M}[3, 3]);$ 
10     $\beta = \text{atan2}(\mathbf{M}[1, 3], \cos \alpha \mathbf{M}[3, 3] - \sin \alpha \mathbf{M}[2, 3]);$ 
11     $\gamma = \text{atan2}(-\mathbf{M}[1, 2], \mathbf{M}[1, 1]);$ 
12  end
13  return  $[\alpha, \beta, \gamma]$ 

```

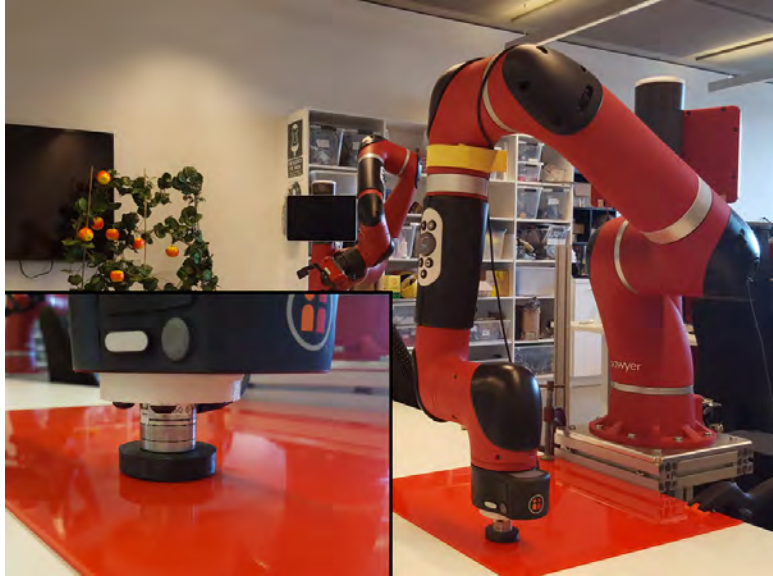


Figure 5. Experimental setup: 7R serial manipulator with a force sensor rigidly attached to the robot end-effector, a tool consisting of a cylinder of 43x43x10mm attached to the sensor to emulate a surface preparation operation and a flat surface as target.

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.

8.1. Parameter values

The control parameters shown below were selected using the steps and practical guidelines detailed in Section 6.2.

- i) The force sensor signal is filtered using a first-order low-pass filter with a cutoff frequency f_{FC} of 73 Hz (see 1st step in Section 6.2), which is implemented in the sensor electronics.
- ii) The estimated bandwidth f_{RC} for robot controller is about 30 Hz and, hence, the control period T_s is set to 20 milliseconds, i.e., $f_{SM} = 25$ Hz (see 2nd step in Section 6.2).
- iii) Parameters used for the Level 1 SMC described in Section 4.1: parameters of the constraint functions $K_{c,z} = K_{c,\alpha} = K_{c,\beta} = 0.15$ (see 3rda step in Section 6.2) and $F_{z,ref} = -15$ N (see 4th step in Section 6.2); FSG method given by $u^+ = 0.84$ and $\{\bar{W}_z = 0.01, \bar{W}_\alpha = 1, \bar{W}_\beta = 1\}$ (see 5th step in Section 6.2). Furthermore a second option has been tested for the switching gain using the discrete ASG method given by (17) with $\mu = 0.02$ (see 5th step in Section 6.2), where the commutation condition has been relaxed to two sample times, see Section 3.2.
- iv) Parameters used for the Level 2 described in Section 4.2: parameters of the tracking controller $K_{T,v} = 5$ and $K_{T,p} = 3$ (see 3rdb step in Section 6.2); and a 2D reference trajectory consisting of a circle of radius 80mm and period 10 seconds.
- v) Parameters used for the Level 3 described in Section 4.3: parameters of the PD controller $K_{3,v} = 3$ and $K_{3,p} = 1$ (see 3rdc step in Section 6.2); and home configuration $\mathbf{q}_0 = [0.003 \quad -0.577 \quad 0.002 \quad 2.044 \quad -0.067 \quad 0.104 \quad 3.382]^T$ rad.

8.2. Comparison experiment

Firstly, an experimental comparison has been conducted between the SMC with FSG and discrete ASG methods when the tool is exposed to different stiffness along its trajectory on the object surface. For this purpose, a polycarbonate flat object of 400x400x3mm is used, see the red surface on the table in Fig. 5. This flat object allows some flexibility and can be folded along the trajectory of the tool, modifying thus the stiffness of the flat object in its perpendicular direction. The switching gain u^+ or initial switching gain $u^+(0)$ for the FSG and ASG methods, respectively, was experimentally tuned to obtain the lowest chattering band and better system performance when the flat object is completely rested on a table, as shown in Fig. 5. The video of both experiments can be played at <https://media.upv.es/player/?id=eeb6fad0-21b0-11e8-b43a-51b816915a74>. The comparison below between both methods is focused on the constraint used to attain the desired force between the tool and the surface in the tool Z-axis, since the differences for the other two constraints used to keep the tool orientation perpendicular to the surface are not significant.

As shown in Fig. 6, the SMC with FSG u^+ is not able to keep the pressure on the surface when changes in the flat object stiffness arise. In particular, the system behavior for the first 26 seconds is the one expected, keeping the contact on the object surface and compensating the pressure on it switching properly the constraint function ϕ_z according to the sensor measurements. From then on, the flat object has different stiffness along the tool trajectory because one corner of the object is held above the table, see snapshots of the video recording depicted in Fig. 7. For this situation, the SMC with FSG is not able to compensate for such disturbances and thus the pressure exerted on the object surface is different at several locations. In particular, around time instants 30s, 40s, 50s and 60s the SM behavior is clearly lost (see the commutation function), which means that the pressure exerted is not enough and the tool is but lightly brushing the flat surface. This poor performance can lead to loss of contact with the object, as indeed occurs around time 70s, which results in task failure.

It is interesting to note that the amplitude of the chattering band for the constraint function ϕ_z is significantly reduced during the SM phases of the second part of the experiment, i.e., when one corner of the object is held above the table. This is because the stiffness coefficient $K_{s,z}$ is significantly reduced when the surface has no support and, since the used value for \bar{W}_z remains constant, it effectively means that the actual value for W_z is also

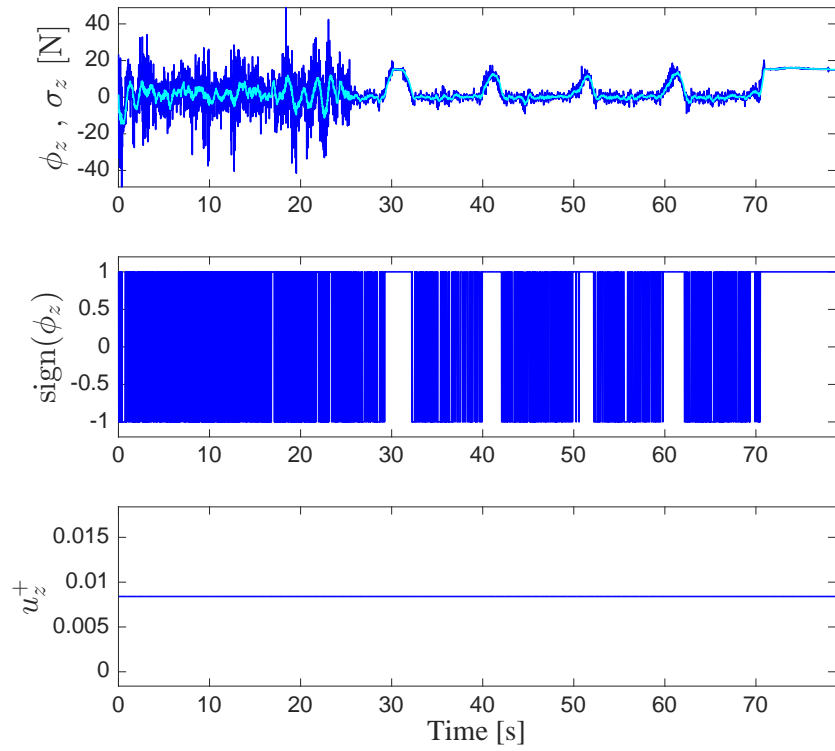


Figure 6. Comparison experiment: SMC with FSG method. From top to bottom: modified and original constraints functions (ϕ_i in dark-blue and σ_i in light-cyan), commutation function and switching gain $u_z^+ = \bar{W}_z u^+$.

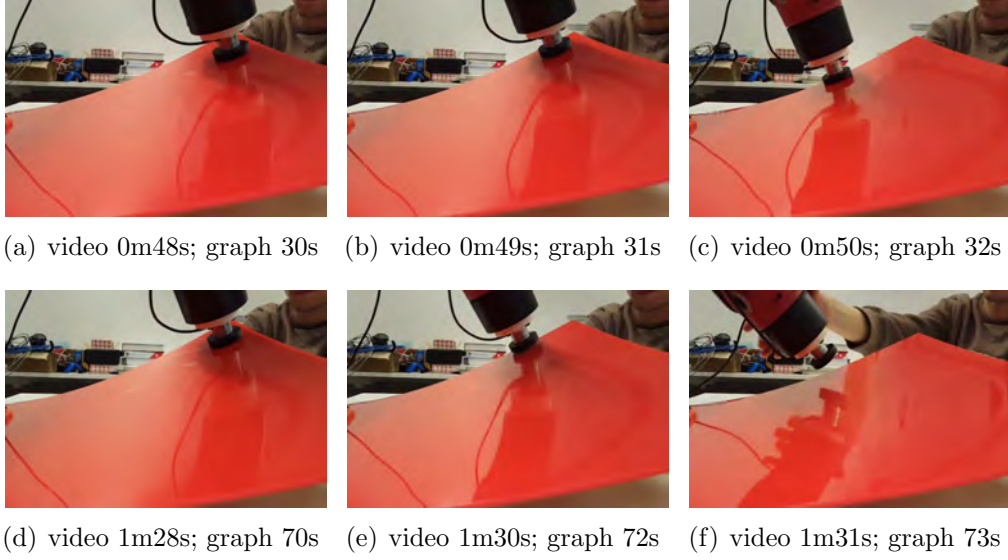


Figure 7. Frames of the video of the comparison experiment. The time instant is indicated for each frame.

reduced, see (28). The end result is a lower chattering band, see (12), as can be clearly observed in the top plot of Fig. 6 from time instant 26s.

In contrast to the FSG approach, when the SMC with the discrete ASG algorithm is used, the system is able to compensate for the changes in object stiffness during the entire experiment and the SM behavior is not lost, as it is shown in Fig. 8. Note that from the interval 25s–70s the controller needs to increase the switching gain in order to deal with changes in the object stiffness, whereas the switching gain from 0s to 24s and from 80s to the end remains around 0.01, which is the optimal value previously designed when the flat object is completely rested on the table, as it is the case. Note as well that the constraint function ϕ_z is permanently switching for the entire experiment, thus allowing to keep the pressure exerted on the object surface regardless the stiffness changes.

Note also that, as before, the amplitude of the chattering band for the constraint function ϕ_z is reduced during the second part of the experiment, i.e., when one corner of the object is held above the table. However, this reduction is lower than that obtained for the FSG case because the value of \bar{W}_z does not remain the constant, it is increased by the discrete ASG law, as seen at the bottom plot in Fig. 8.

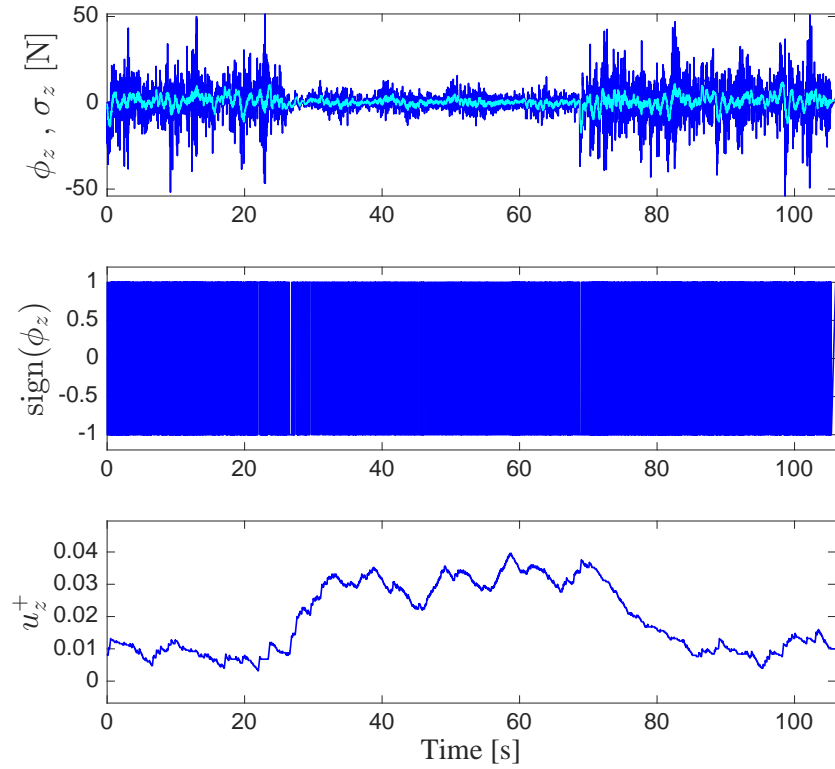


Figure 8. Comparison experiment: SMC with the discrete ASG method. From top to bottom: modified and original constraints functions (ϕ_i in dark-blue and σ_i in light-cyan), commutation function and switching gain $u_z^+ = \overline{W}_z u^+$.

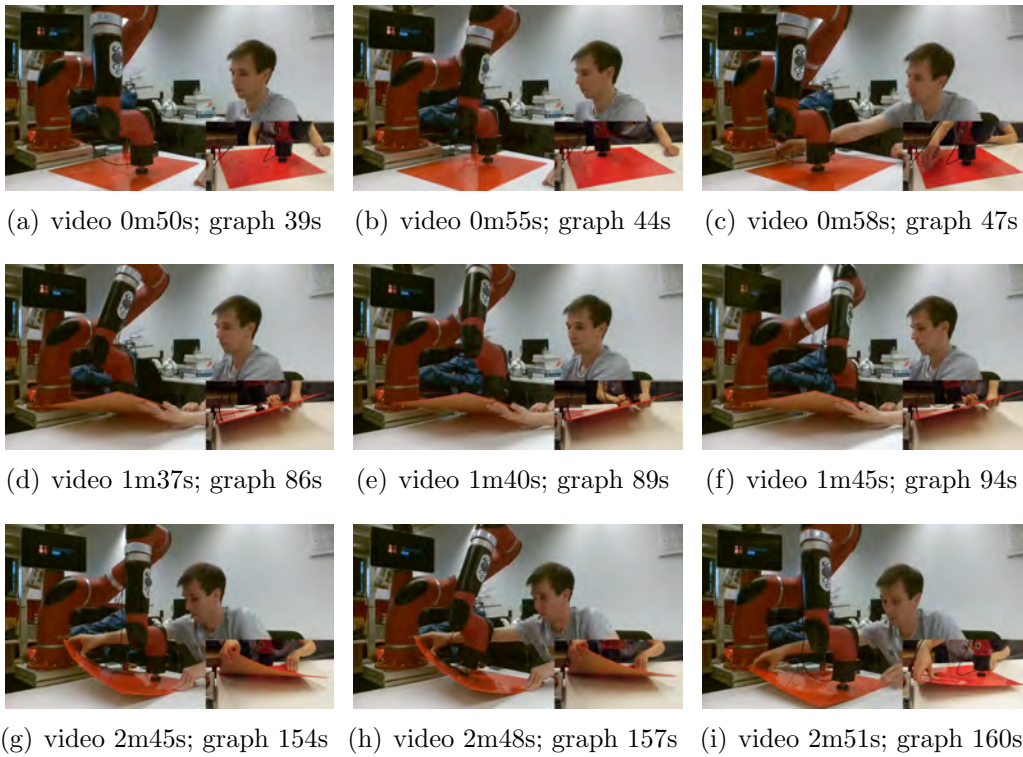


Figure 9. Frames of the video of the dynamic experiment. The time instant is indicated for each frame.

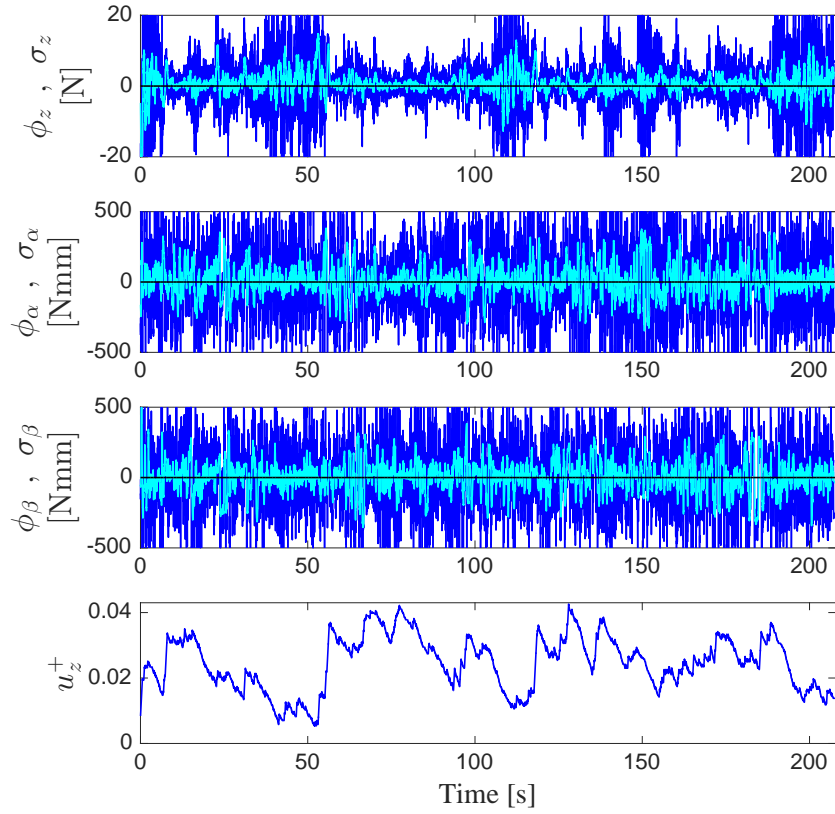


Figure 10. Dynamic experiment: modified and original constraints functions (ϕ_i in dark-blue and σ_i in light-cyan) and switching gain $u_z^+ = \overline{W}_z u^+$.

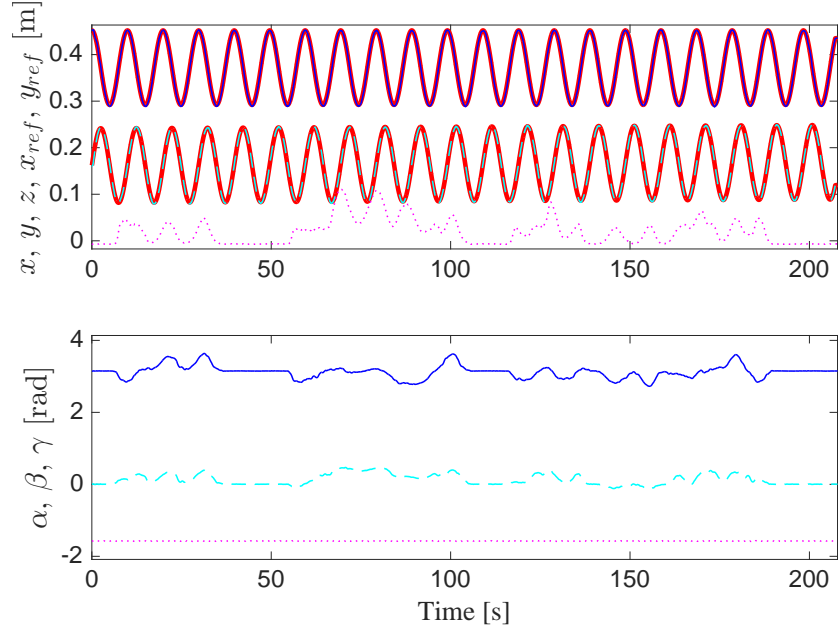


Figure 11. Dynamic experiment. Top: tool position $\{x, y, z\}$ and reference signals $\{x_{ref}, y_{ref}\}$ as a function of time. Bottom: tool orientation $\{\alpha, \beta, \gamma\}$ in roll-pitch-yaw angles as a function of time. The reference signals are in solid-thick line, $\{x, \alpha\}$ signals in solid-thin line, $\{y, \beta\}$ signals in dashed-thin line and $\{z, \gamma\}$ signals in dotted-thin line.

8.3. Dynamic and reorientation experiments

In order to verify the robustness and performance of the proposed approach, a more dynamic experiment has been conducted introducing changes in the position, orientation and stiffness of the flat target object. The video of this experiment can be played at <https://media.upv.es/player/?id=fe067e10-21ac-11e8-b43a-51b816915a74> and Fig. 9 shows several instants of the video: Fig. 9(a), 9(b) and 9(c) show the robot behavior when the flat target object rests on the table; the other frames in Fig. 9 show the system behavior when the flat object is held above the table at different positions and orientations. Fig. 10 shows the performance of the system in terms of constraints, where it can be seen that the constraint functions $\{\sigma_z, \sigma_\alpha, \sigma_\beta\}$ are switching around zero as expected. This means that the surface treatment task is being carried out as expected: the tool orientation is kept perpendicular to the object surface and the pressure with the tool on the surface is maintained regardless of the changes on the object position, orientation and stiffness. The figure also shows the variation of the computed switching gain

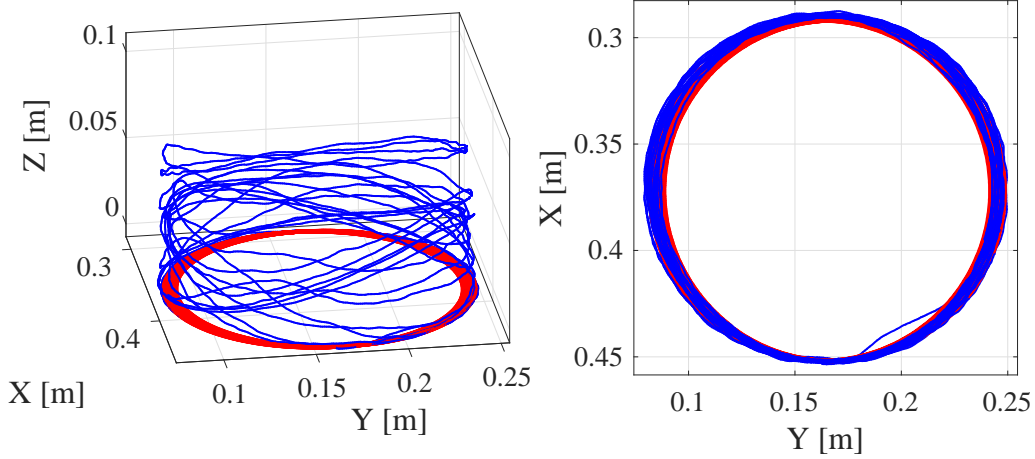


Figure 12. Dynamic experiment: Trajectory followed by the robot tool (thin-blue line) and circular reference trajectory (thick-red line). Left: 3D representation. Right: top view.

in order to maintain a stable pressure on the object surface regardless of the changes in stiffness. Fig. 11 and 12 show the Cartesian trajectory followed by the tool pose as a function of time and the corresponding 3D representation, respectively. Note that the tool coordinates in X - and Y -axes follow the reference signals so closely that they can be hardly distinguished.

Another experiment has been conducted to highlight the reorientation capability of the SMC in Level 1. For this experiment, the reference trajectory for the tracking controller is just a 2D point. The video of this experiment can be played at <https://media.upv.es/player/?id=79c83140-21aa-11e8-b43a-51b816915a74>, where it can be seen that the controller is able to dynamically adapt the tool orientation for a wide range of values: three snapshots of this experiment are shown in Fig. 13 to illustrate that the robot tool is kept perpendicular to the surface in very challenging cases.

8.4. Discussion

Regarding the first advantage listed in Section 4.5 for the proposed SMC, it can be seen in the above experiments that due to the approaching parameter $K_{c,i}$, the original constraint functions σ_i are significantly smoothed when compared to ϕ_i , see the first plot in Fig. 6 and Fig. 8 and the first three plots in Fig. 10.

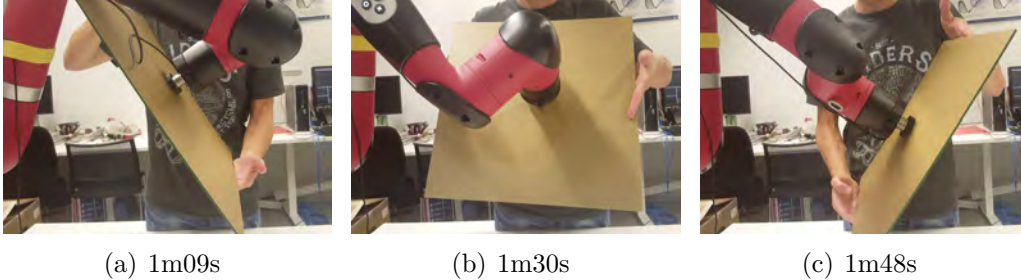


Figure 13. Frames of the video of the reorientation experiment. The time instant is indicated for each frame.

Regarding the robustness of the method, the dynamic experiments have revealed how the robot is able to automatically adapt the tool position and orientation to properly perform the surface treatment regardless of changes in the object position, orientation and stiffness. In fact, the proposed SMC algorithm is robust [19] against all the terms included in the Lie derivatives $L_f\phi_i$: the inaccuracies \mathbf{d}_c of the low-level control loop; the pose \mathbf{p}_s (position and orientation) of the surface of the environment and its derivative; and, in general, all the reasonable perturbations and unmodeled dynamics (non-linearities, friction forces, etc.) that can be met in practical scenarios for the task of surface treatment, as the challenging examples shown have demonstrated (please refer also to all the video links included).

Furthermore, the SMC algorithm only requires the Lie derivatives $\mathbf{L}_g\phi_i$, which are given by the robot Jacobian. Thus, the proposed method can be programmed in a few lines of code, see Section 7, which translates in efficient computational times for the control loop: 15 microseconds for the above experiments using a modern computer.

As is the case with other SMC-based controllers, the main disadvantage of the method is the chattering drawback, see Section 3.1, although this problem becomes negligible for reasonable fast sampling rates.

9. Conclusions

A hybrid position-force control approach has been developed in this work for robotic surface treatment using task priority and sliding mode control. In particular, sliding mode control has been used to satisfy equality constraints in order to perform surface treatment tasks, i.e., to attain a desired

tool pressure on the surface and to keep the tool orientation perpendicular to the surface. In order to deal with sudden changes of the material stiffness suffered by the polishing tool along the trajectory, which can produce instability and poor polishing performance, several adaptive switching gain laws have been considered and compared. 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.

The applicability and effectiveness of the proposed approach is substantiated by experimental results 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 Grants VALi+d APOSTD/2016/044 and BEST/2017/029.

References

- [1] S. S. Martínez, J. G. Ortega, J. G. García, A. S. García, E. E. Estévez, An industrial vision system for surface quality inspection of transparent parts, *The International Journal of Advanced Manufacturing Technology* 68 (2013) 1123–1136.
- [2] Y. Shi, D. Zheng, L. Hu, Y. Wang, 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 (2012) 1061–1073.
- [3] G. Orta, A. S. Bilgi, K. Tasdemir, H. Kalkan, A hyperspectral imaging based control system for quality assessment of dried figs, *Computers and Electronics in Agriculture* 130 (2016) 38 – 47.
- [4] J. Molina, J. E. Solanes, L. Arnal, J. Tornero, On the detection of defects on specular car body surfaces, *Robotics and Computer-Integrated Manufacturing* 48 (2017) 263 – 278.

- [5] Micro-Epsilon, Fully automatic surface inspection of painted car bodies, 2015.
- [6] ISRA, Paintscan: Paint inspection with in-line deflectometry, 2015.
- [7] L. Arnal, J. E. Solanes, J. Molina, J. Tornero, Detecting dings and dents on specular car body surfaces based on optical flow, *Journal of Manufacturing Systems* (2017).
- [8] H. Elbehery, A. Hefnawy, M. Elewa, Surface defects detection for ceramic tiles using image processing and morphological techniques, in: *WEC*.
- [9] N. Sameer Ahamad, J. Bhaskara Rao, Analysis and Detection of Surface Defects in Ceramic Tile Using Image Processing Techniques, Springer India, New Delhi, pp. 575–582.
- [10] F. Tian, Z. Li, C. Lv, G. Liu, Polishing pressure investigations of robot automatic polishing on curved surfaces, *The International Journal of Advanced Manufacturing Technology* 87 (2016) 639–646.
- [11] Q. Wu, X. Wang, B. Chen, H. Wu, S. Z., Development and hybrid force/position control of a compliant rescue manipulator, *Mechatronics* 46 (2017) 143–153.
- [12] X. Xie, L. Sun, Force control based robotic grinding system and application, in: 2016 12th World Congress on Intelligent Control and Automation (WCICA), pp. 2552–2555.
- [13] L. Roveda, F. Vicentini, N. Pedrocchi, L. M. Tosatti, Impedance control based force-tracking algorithm for interaction robotics tasks: An analytically force overshoots-free approach, in: 2015 12th International Conference on Informatics in Control, Automation and Robotics (ICINCO), volume 02, pp. 386–391.
- [14] E. Akdoğan, M. Aktan, A. Koru, M. Arslan, M. Atlıhan, B. Kuran, Hybrid impedance control of a robot manipulator for wrist and forearm rehabilitation: Performance analysis and clinical results, *Mechatronics* 49 (2018) 77–91.

- [15] Y. Kakinuma, K. Igarashi, S. Katsura, T. Aoyama, Development of 5-axis polishing machine capable of simultaneous trajectory, posture, and force control, *CIRP Annals - Manufacturing Technology* 62 (2013) 379 – 382.
- [16] Y. Oba, Y. Yamada, K. Igarashi, S. Katsura, Y. Kakinuma, Replication of skilled polishing technique with serial-parallel mechanism polishing machine, *Precision Engineering* 45 (2016) 292 – 300.
- [17] Y. Oba, Y. Kakinuma, Simultaneous tool posture and polishing force control of unknown curved surface using serial-parallel mechanism polishing machine, *Precision Engineering* 49 (2017) 24 – 32.
- [18] L. Pilny, G. Bissacco, Development of on the machine process monitoring and control strategy in robot assisted polishing, *CIRP Annals - Manufacturing Technology* 64 (2015) 313 – 316.
- [19] C. Edwards, S. Spurgeon, *Sliding Mode Control: Theory and Applications*, Taylor & Francis, UK, 1st edition, 1998.
- [20] A. T. Massoud, H. A. ElMaraghy, T. Lahdhiri, On the robust nonlinear motion position and force control of flexible joints robot manipulators, *Journal of Intelligent and Robotic Systems* 25 (1999) 227–254.
- [21] S. Sakaino, K. Ohnishi, Sliding mode control based on position control for contact motion applied to hopping robot, in: *2006 IEEE International Conference on Industrial Technology*, pp. 170–175.
- [22] E. Bassi, F. Benzi, L. M. Capisani, D. Cuppone, 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*, pp. 2966–2971.
- [23] S.-J. Huang, Y.-C. Liu, S.-H. Hsiang, Robotic end-effector impedance control without expensive torque/force sensor, *International Journal of Mechanical, Aerospace, Industrial, Mechatronic and Manufacturing Engineering* 7 (2013) 1446 – 1453.

- [24] A. Fedele, A. Fioretti, G. Ulivi, Implementation of a hybrid force-position controller using sliding mode techniques, in: IEEE International Conference on Robotics and Automation, pp. 2126–2133.
- [25] R. Rascón, D. Rosas, L. Moreno, Tracking and force control of a scara robot under a constraint using sliding mode control, in: Congreso Nacional de Control Automático.
- [26] E. Engeberg, S. Meek, M. Minor, Hybrid force-velocity sliding mode control of a prosthetic hand, IEEE Transactions on Biomedical Engineering 55 (2008) 1572–1581.
- [27] M. Farooq, D. Wang, N. Dar, Adaptive sliding-mode hybrid force/position controller for flexible joint robot, in: IEEE International Conference on Mechatronics and Automation (ICMA 2008), volume 25, pp. 724–731.
- [28] L. Massimiliano Capisani, A. Ferrara, Trajectory planning and second-order sliding mode motion/interaction control for robot manipulators in unknown environments, IEEE Transactions on Industrial Electronics 59 (2012) 3189–3198.
- [29] E. Bassi, F. Benzi, L. M. Capisani, D. Cuppone, 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, pp. 2966–2971.
- [30] A. Rashid Husain, M. Noh Ahmad, A. Halim Mohd Yatim, Chattering-free sliding mode control for an active magnetic bearing system, volume 59, pp. 385–391.
- [31] C. Korusch, P. Puleston, M. Mayosky, Fundamentals of Sliding-Mode Control Design, Springer London, London, pp. 35–71.
- [32] V. Utkin, Discussion aspects of high-order sliding mode control, IEEE Transactions on Automatic Control 61 (2016) 829–833.
- [33] J. Zhu, K. Khayati, On a new adaptive sliding mode control for mimo nonlinear systems with uncertainties of unknown bounds, International Journal of Robust and Nonlinear Control 27 (2017) 942–962. Rnc.3608.

- [34] Q. Shen, D. Wang, S. Zhu, E. K. Poh, Integral-type sliding mode fault-tolerant control for attitude stabilization of spacecraft, *IEEE Transactions on Control Systems Technology* 23 (2015) 1131–1138.
- [35] M. Taleb, F. Plestan, B. Bououlid, An adaptive solution for robust control based on integral high-order sliding mode concept, *International Journal of Robust and Nonlinear Control* 25 (2015) 1201–1213.
- [36] B. C. Z. Chen, X. Liu, Switching gain reduction in adaptive sliding mode control for rigid spacecraft attitude maneuvers, *Journal of Mathematical Problems in Engineering* 2013 (2013).
- [37] B. Cong, Z. Chen, X. Liu, On adaptive sliding mode control without switching gain overestimation, *International Journal of Robust and Nonlinear Control* 24 (2014) 515–531.
- [38] F. Plestan, Y. Shtessel, V. Bregeault, A. Poznyak, New methodologies for adaptive sliding mode control, *International Journal of Control* 83 (2010) 1907–1919.
- [39] Y. S. Lu, Sliding-mode disturbance observer with switching-gain adaptation and its application to optical disk drives, *IEEE Transactions on Industrial Electronics* 56 (2009) 3743–3750.
- [40] M. Raibert, J. Craig, Hybrid position/force control of manipulators, *Journal of Dynamic Systems, Measurement, and Control* 102 (1981) 126–133.
- [41] S. Chiaverini, G. Oriolo, I. Walker, Kinematically redundant manipulators, *Springer Handbook of Robotics* (2008) 245–268.
- [42] Y. Nakamura, H. Hanafusa, T. Yoshikawa, Task-priority based redundancy control of robot manipulators, *The Int. Journal of Robotics Research* 6 (1987) 3–15.
- [43] B. Siciliano, J. Slotine, A general framework for managing multiple tasks in highly redundant robotic systems, in: *Proc. of the Fifth Int. Conf. on Advanced Robotics (ICAR'91)*, Pisa, Italy, pp. 1211–1216, 1991.
- [44] V. Utkin, J. Guldner, J. Shi, *Sliding Mode Control in Electro-Mechanical Systems*, Taylor & Francis, London, 2nd edition, 2009.

- [45] T.-P. Leung, Q.-J. Zhou, C.-Y. Su, An adaptive variable structure model following control design for robot manipulators, *IEEE Transactions on Automatic Control* 36 (1991) 347–353.
- [46] Y. A. Jiang, D. J. Clements, T. Hesketh, J. S. Park, Adaptive learning control of robot manipulators in task space, in: *American Control Conference, 1994*, volume 1, pp. 207–211 vol.1.
- [47] H. Lenz, R. Berstecher, M. K. Lang, Adaptive sliding-mode control of the absolute gain, *IFAC Proceedings Volumes* 31 (1998) 639 – 644. 4th IFAC Symposium on Nonlinear Control Systems Design 1998 (NOLCOS'98), Enschede, The Netherlands, 1-3 July.
- [48] G. Wheeler, C. Su, Y. Stepanenko, A sliding mode controller with improved adaption laws for the upper bounds on the norm of uncertainties, *Automatica* 34 (1998) 1657–1661.
- [49] G. Monsees, J. Scherpen, Adaptive switching gain for a discrete-time sliding mode controller, *Int. Journal of Control* 75 (2002) 242–251.
- [50] Q. Xu, Adaptive discrete-time sliding mode impedance control of a piezoelectric microgripper, *IEEE Trans. on Robotics* 29 (2013) 663–673.
- [51] B. Siciliano, L. Sciavicco, L. Villani, G. Oriolo, *Robotics: Modelling, Planning and Control*, Springer-Verlag, London, UK, 2009.
- [52] W. Khalil, E. Dombre, *Modeling, Identification and Control of Robots*, Taylor & Francis Inc., Bristol, PA, 2002.
- [53] A. Liégeois, Automatic supervisory control of the configuration and behavior of multibody mechanisms, *IEEE Trans. on Systems, Man and Cybernetics* 7 (1977) 868–871.
- [54] L. Huo, L. Baron, The joint-limits and singularity avoidance in robotic welding, *Industrial Robot: An Int. Journal* 35 (2008) 456–464.
- [55] G. Antonelli, Stability analysis for prioritized closed-loop inverse kinematic algorithms for redundant robotic systems, *IEEE Transactions on Robotics* 25 (2009) 985–994.
- [56] G. Golub, C. Van Loan, *Matrix Computations*, The Johns Hopkins University InPress, Baltimore, MD, 3rd edition, 1996.