Adaptive Robust Control and Admittance Control for Contact-Driven Robotic Surface Conditioning

J. Ernesto Solanes*[a], Luis Gracia[a], Pau Muñoz-Benavent[a], Alicia Esparza[a], Jaime Valls Miro[b], Josep Tornero[a]

[a] *Instituto de Diseño y Fabricación, Universitat Politècnica de València, Camino de Vera s/n, 46022 València, Spain (* Corresponding author e-mail: esolanes@idf.upv.es)*
[b] *Centre for Autonomous Systems, Faculty of Engineering and IT, University of Technology Sydney (UTS), NSW 2007 Sydney, Australia.*

Abstract

This work presents a hybrid position/force control of robots for surface contact conditioning tasks such as polishing, profiling, deburring, etc. The robot force control is designed using sliding mode ideas to benefit from robustness. On the one hand, a set of equality constraints are defined to attain the desired tool pressure on the surface, as well as to keep the tool orientation perpendicular to the surface. On the other hand, inequality constraints are defined to adapt the tool position to unmodeled features present in the surface, e.g., a protruding window frame. Conventional and non-conventional sliding mode controls are used to fulfill the equality and inequality constraints, respectively. Furthermore, in order to deal with sudden changes of the material stiffness, which are forwarded to the robot tool and can produce instability and bad performance, adaptive switching gain laws are considered not only for the conventional sliding mode control but also for the non-conventional sliding mode control. A lower priority tracking controller is also defined to follow the desired reference trajectory on the target surface. Moreover, the classical admittance control typically used in force control tasks is adapted for the proposed surface contact application in order to experimentally compare the performance of both control approaches. The effectiveness of the proposed method is substantiated by experimental results using a redundant 7R manipulator, whereas its advantages over the classical admittance control approach are experimentally shown.

Keywords: Robot force control, adaptive switching gain, sliding mode control

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]. 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 contact conditioning 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.

This work is focused on the automation of surface contact conditioning using robot systems and force feedback, which may be useful in many industrial processes such as polishing, profiling, deburring, etc. Since the robot tool has to be in contact with the target surface both the tool pose and the exerted forces have to be controlled. Many approaches can be found in the literature tackling the problem of robot force control. In particular, a classical technique to handle this issue consist in using an impedance/admittance controller, see [4, 5, 6, 7, 8] among others. Other robot force control approaches are based on Sliding Mode Control (SMC) theory [9], e.g., [10, 11, 12]. One typical problem of SMC is related to the controller switching gain. High values of the switching gain increase the control effort and the *chattering* band [13]. On the contrary, adjusting the switching gain to minimize the control effort and chattering band at a certain operating point may cause the control to become unstable for another operating point. To overcome this problem, Adaptive SMC (ASMC) solutions have been proposed in the literature, i.e., SMC approaches with an adaptive switching gain (ASG), e.g., see [14, 15, 16] among others.

Similarly to some of the works listed above, this paper addresses the problem of robotic surface contact conditioning using a hybrid position/force approach. In particular, force control is performed using ASMC in order to benefit from robustness, whereas position control is performed using a tracking controller. The proposed method is based on satisfying a set of constraints to properly perform the desired surface contact conditioning. In this sense, the authors previously presented in [17] a conventional SMC to fulfill equality constraints, whereas in [18] developed a non-conventional SMC to fulfill inequality constraints. In both cases, however, the control application was focused on robot visual servoing. On the other hand, the proposal in this work presents the following distinctive innovative features:

1. The development of the required constraints and control scheme for a *surface contact conditioning* task using force feedback.
2. The *simultaneous use* of both equality and inequality constraints in the same SMC application.
3. The ASG algorithm is considered not only as part for the conventional SMC, but is also incorporated within the context of a non-conventional SMC. In this sense, the *novel ASG method* proposed for the non-conventional SMC represents a relevant theoretical contribution of this work. The successful performance of the novel ASG algorithm is shown with real experimentation.
4. Classical admittance control, which is typically used in force control applications [5, 6, 7], has been revisited and *adapted* to use it in the proposed surface contact application for comparative purposes.
5. A thorough *experimental comparison* between the proposed ASMC method and the classical admittance control approach has been carried out to illustrate its advantages.

The paper is organized as follows. Next section introduces some preliminaries, while Section 3 presents the SMC theory used in this work. The proposed approach for robotic surface contact conditioning is developed in Section 4. For comparison purposes, the classical admittance control method is adapted in Section 5 for the proposed surface contact application. The effectiveness of the proposed approach is substantiated by experimental results in Section 6 using a redundant 7R manipulator. Furthermore, the advantages of the proposed method over the classical admittance control approach are experimentally shown in Section 7. Finally, some conclusions are given.

2. Preliminaries

Kinematics. The robot *pose* $\mathbf{p} = \begin{bmatrix} x & y & z & \alpha & \beta & \gamma \end{bmatrix}^{\mathrm{T}}$ with orientation expressed in roll-pitch-yaw angles and the robot *configuration* $\mathbf{q} =$

$\begin{bmatrix} q_1 & \cdots & q_n \end{bmatrix}^{\mathrm{T}}$, and their derivatives, are related by the following equations:

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

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

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

where vector $\mathbf{l}$ and matrix $\mathbf{J}$ are the kinematic function and analytical Jacobian [19] of the robot, respectively.

In this work, the so-called geometric Jacobian [19] relative to the tool coordinate system $\mathbf{J}_n$ is also required. This matrix transforms the joint velocities $\dot{\mathbf{q}}$ to the linear and angular velocities of the end-effector relative to the tool coordinate system. The relationship between the analytical Jacobian $\mathbf{J}$ and the geometric Jacobian $\mathbf{J}_n$ is given by [20]:

$$\mathbf{J} = \begin{bmatrix} \mathbf{I}_3 & \mathbf{O}_3 \\ \mathbf{O}_3 & \begin{bmatrix} 1 & 0 & \sin(\beta) \\ 0 & \cos(\alpha) & -\sin(\alpha)\cos(\beta) \\ 0 & \sin(\alpha) & \cos(\alpha)\cos(\beta) \end{bmatrix}^{-1} \end{bmatrix} \begin{bmatrix} {}^{\mathbf{0}}\mathbf{R}_{\mathbf{n}} & \mathbf{O}_3 \\ \mathbf{O}_3 & {}^{\mathbf{0}}\mathbf{R}_{\mathbf{n}} \end{bmatrix} \mathbf{J}_n = \mathbf{L}\,\mathbf{R}\,\mathbf{J}_n, \tag{4}$$

where α and β are the roll and pitch angles, respectively, ${}^{\mathbf{0}}\mathbf{R}_{\mathbf{n}}$ is the rotation matrix between the base and tool coordinate systems, $\mathbf{O}_3$ is the 3×3 zero matrix, $\mathbf{I}_3$ is the 3×3 identity matrix, matrix $\mathbf{L}$ is required to transform the end-effector angular velocities to the derivatives of roll-pitch-yaw angles and $\mathbf{R}$ represents a projection matrix.

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

$$\ddot{\mathbf{q}} = \ddot{\mathbf{q}}_c + \mathbf{d}_c \tag{5}$$

holds approximately true, where $\mathbf{d}_c$ represents inaccuracies due to disturbances.

Task-priority based redundancy resolution. The task-priority strategy [21] allows to tackle several objectives simultaneously assigning an order of priority to each one. Let us consider M tasks which consist on calculating the commanded joint acceleration vector $\ddot{\mathbf{q}}_c$ to fulfill the following equality constraints:

$$\mathbf{A}_i\ddot{\mathbf{q}}_c = \mathbf{b}_i, \quad i = 1, \ldots, M, \tag{6}$$

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 (6) is given by [22]:

$$\begin{aligned} \ddot{\mathbf{q}}_{c,i} &= \ddot{\mathbf{q}}_{c,i-1} + (\mathbf{A}_i\mathbf{N}_{i-1})^{\dagger}(\mathbf{b}_i - \mathbf{A}_i\ddot{\mathbf{q}}_{c,i-1}) & (7)\\ \mathbf{N}_i &= \mathbf{N}_{i-1}(\mathbf{I} - (\mathbf{A}_i\mathbf{N}_{i-1})^{\dagger}(\mathbf{A}_i\mathbf{N}_{i-1})), & (8)\\ &\text{with i} = 1, \ldots, \text{M}, \ \ddot{\mathbf{q}}_{\text{c},0} = \mathbf{0} \text{ and } \mathbf{N}_0 = \mathbf{I}, \end{aligned}$$

where $\mathbf{I}$ and $\mathbf{0}$ denote the identity matrix and 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

3.1. Conventional SMC to satisfy equality constraints

Theorem 1. *Consider the following dynamical system with n_x states and n_u inputs given by:*

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{d}) + \mathbf{g}(\mathbf{x})\,\mathbf{u}, \tag{9}$$

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 $\phi_{eq,i}(\mathbf{x}) = 0$, $i = 1, \ldots, N_{eq}$, where $\phi_{eq,i}(\mathbf{x})$ 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:*

$$\Phi_{eq} = \{\mathbf{x} \mid \phi_{eq,i}(\mathbf{x}) = 0\}, \tag{10}$$

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

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

$$\mathbf{L_g}\boldsymbol{\phi}_{eq}\mathbf{u} = -\mathbf{W}_{eq}\mathrm{sign}(\boldsymbol{\phi}_{eq})\, u_{eq}^{+} \tag{11}$$

$$u_{eq}^{+} > \|L_f\boldsymbol{\phi}_{eq}\|_1/\mathrm{diag}_{\min}(\mathbf{W}_{eq}), \tag{12}$$

where $\boldsymbol{\phi}_{eq}$ is a column vector with all the constraint functions $\phi_{eq,i}$, the scalar $L_f\phi_{eq,i} = \frac{\partial\phi_{eq,i}^T}{\partial\mathbf{x}}\mathbf{f}$ and the row vector $\mathbf{L_g}\phi_{eq,i} = \frac{\partial\phi_{eq,i}^T}{\partial\mathbf{x}}\mathbf{g}$ denote the Lie derivatives of $\phi_{eq,i}(\mathbf{x})$ in the direction of vector field $\mathbf{f}$ *and in the direction of the set of vector fields* $\mathbf{g}$*, respectively, column vector $L_f\boldsymbol{\phi}_{eq}$ contains the elements $L_f\phi_{eq,i}$ of all equality constraints, matrix $\mathbf{L_g}\boldsymbol{\phi}_{eq}$ contains the row vectors $\mathbf{L_g}\phi_{eq,i}$ of all equality constraints,* $\mathrm{sign}(\cdot)$*represents the* sign function *(typically used in SMC), positive scalar u_{eq}^{+} is the so-called switching gain, which can be either constant or varying in time, $\mathbf{W}_{eq}$ is a diagonal matrix representing the switching gain weights for the constraints,* $\|\cdot\|_1$ *represents the 1-norm (also known as the Taxicab norm) and function* $\mathrm{diag}_{\min}(\cdot)$ *computes the minimum value of the diagonal elements of a matrix.*

Proof. The proof can be obtained straightforward from that in [17]. Details omitted for brevity. □

3.2. Non-conventional SMC to satisfy inequality constraints

Theorem 2. *Consider the dynamical system given by* (9) *and consider also that the system state vector* $\mathbf{x}$ *is subject to* inequality *constraints $\phi_{in,i}(\mathbf{x}) \leq 0$, $i = 1, \ldots, N_{in}$, where $\phi_{in,i}(\mathbf{x})$ 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:*

$$\Phi_{in} = \{\mathbf{x}\,|\,\phi_{in,i}(\mathbf{x}) \leq 0\}\,, \tag{13}$$

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

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

$$\mathrm{v2dm}\left(\mathrm{pos}\left(\boldsymbol{\phi}_{in}\right)\right)\mathbf{L_g}\boldsymbol{\phi}_{in}\mathbf{u} = -\mathbf{W}_{in}\,\mathrm{pos}\left(\boldsymbol{\phi}_{in}\right)\; u_{in}^{+} \tag{14}$$

$$u_{in}^{+} > \sum_{i=1}^{n_a}\left(\max(L_f\phi_{in,i}, 0)\right)/\mathrm{diag}_{\min}(\mathbf{W}_{in}), \tag{15}$$

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

Proof. The proof can be obtained straightforward from that in [18]. Details omitted for brevity. □

3.3. SMC to satisfy both equality and inequality constraints

Combining the results of the previous two subsections, the SMC to satisfy simultaneously both equality and inequality constraints is given by:

$$\begin{aligned}
\begin{bmatrix} \mathbf{L_g}\boldsymbol{\phi}_{eq} \\ \text{v2dm}\left(\text{pos}\left(\boldsymbol{\phi}_{in}\right)\right)\mathbf{L_g}\boldsymbol{\phi}_{in} \end{bmatrix} \mathbf{u} &= -\begin{bmatrix} \mathbf{W}_{eq} & \mathbf{0} \\ \mathbf{0} & \mathbf{W}_{in} \end{bmatrix} \begin{bmatrix} \text{sign}(\boldsymbol{\phi}_{eq}) \\ \text{pos}\left(\boldsymbol{\phi}_{in}\right) \end{bmatrix} u^+ \\
\mathbf{L_g}\boldsymbol{\phi}\,\mathbf{u} &= -\mathbf{W} \begin{bmatrix} \text{sign}(\boldsymbol{\phi}_{eq}) \\ \text{pos}\left(\boldsymbol{\phi}_{in}\right) \end{bmatrix} u^+, \qquad (16)
\end{aligned}$$

where the second line represents the same equation using the new variables $\mathbf{L_g}\boldsymbol{\phi}$ and $\mathbf{W}$, and u^+ is the switching gain for the global SMC, whose lower bound is given by:

$$u^+ > \|L_f\boldsymbol{\phi}_{eq}\|_1/\text{diag}_{\min}(\mathbf{W}_{eq}) + \sum_{i=1}^{n_a}(\max(L_f\phi_{in,i},0))/\text{diag}_{\min}(\mathbf{W}_{in}). \quad (17)$$

3.4. Modified constraints

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

$$\phi_{eq,i} = \sigma_{eq,i} + K_{eq,i}\dot{\sigma}_{eq,i} = 0 \quad (18)$$

$$\phi_{in,i} = \sigma_{in,i} + K_{in,i}\dot{\sigma}_{in,i} \leq 0, \quad (19)$$

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

3.5. Chattering

The chattering band [9] of the proposal can be obtained using the Euler-integration of the discontinuous control action given by Eq. (11), that is:

$$\triangle\boldsymbol{\phi} = T_s \, |\mathbf{L_g}\boldsymbol{\phi}\, \mathbf{u}| = T_s \, u^+ \, \mathrm{diag}_v(\mathbf{W}), \tag{20}$$

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

3.6. 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.

3.6.1. Adaptive switching gain for the conventional sliding mode control

Many ASG laws can be found in the literature for conventional SMC, e.g., [14, 15, 16] among others. In particular, this works considers the ASG discrete algorithm proposed in [23, 24], which is given by the following law for i-th equality constraint:

$$u^+(k) = |u^+(k-1) + T_s \, \mu_{eq} \, \mathrm{sign}(\phi_{eq}(k)) \, \mathrm{sign}(\phi_{eq}(k-1))|, \tag{21}$$

where $u^+(k)$ and $u^+(k-1)$ are the values of the switching gain for the current and the previous time steps, respectively, $\phi_{eq}(k)$ and $\phi_{eq}(k-1)$ are values of the equality constraint function for the current and the previous time steps, respectively, and μ_{eq} is a positive configuration parameter that determines the speed of the adaptation.

3.6.2. Adaptive switching gain for the non-conventional sliding mode control

In the same spirit of the ASG discrete algorithm (21), this work proposes the following ASG law for the non-conventional SMC:

$$u^+(k) = |u^+(k-1) + T_s \, \mu_{in} \, f_{ASG}(\phi_{in}(k), \phi_{in}(k-1))|, \tag{22}$$

where μ_{in} is the speed adaptation parameter and function $f_{ASG}(A, B)$ is equal to: 1 if $B > 0$ and $A > 0$; -1 if $B > 0$ and $A < 0$; and 0 for the remaining cases. This ASG law is explained as follows: when the inequality constraint is unfulfilled at a given time step (i.e., $\phi_{in}(k-1) > 0$) the switching gain is decreased or increased at the next time step depending on whether the constraint boundary is successfully crossed or not at the next time step, i.e., depending on whether the inequality constraint is fulfilled (i.e., $\phi_{in}(k) < 0$) or not (i.e., $\phi_{in}(k) > 0$). Hence, the method aims to lead the system to cross the boundary of the inequality constraint whenever it is unfulfilled while minimizing the switching gain.

Therefore, starting from any initial point in the non-allowed region the system will cross the boundary of the inequality constraint in finite time and, for constant conditions, the switching gain given by (22) converges to a bounded region. The proof can be obtained considering the extension described in [23] and using $\rho^+(k) = u^+$ and $\rho^-(k) = 0$. Details omitted for brevity.

It is interesting to remark that, due to unmodeled dynamics (nonlinearities, delays, noisy signals, etc.), in some cases it could be difficult for real systems to cross the SM surface or the constraint boundary in just one time step. In these cases, the commutation condition (22) could be "relaxed", i.e., a few time steps (e.g., two or three) may be considered instead of just one time step to evaluate if the SM surface or the constraint boundary has been successfully crossed or not in order to decrease or increase the switching gain.

4. Proposed approach

4.1. System tasks

Three tasks with different priority levels are considered in this work:

- The first level (high-priority task) includes the equality and inequality *constraints* that must be satisfied at all times to properly accomplish the surface contact conditioning 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, inequality constraints are defined to adapt the tool position to any surface obstacles (e.g., protrusions, screws, nuts, etc.) when applying the surface contact conditioning.

- The second level (medium-priority task) is designed for *reference tracking* in order to apply the contact conditioning 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 this work) 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 Section 3. In order to use this theory, a dynamical system in the form of Eq. (9) is considered with the state vector $\mathbf{x} = \begin{bmatrix}\mathbf{q}^{\mathrm{T}} & \dot{\mathbf{q}}^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}}$, 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 (5) the state equation results in:

$$\dot{\mathbf{x}} = \begin{bmatrix}\mathbf{O} & \mathbf{I}\\ \mathbf{O} & \mathbf{O}\end{bmatrix}\mathbf{x} + \begin{bmatrix}\mathbf{0}\\ \mathbf{d}_c\end{bmatrix} + \begin{bmatrix}\mathbf{O}\\ \mathbf{I}\end{bmatrix}\mathbf{u}, \tag{23}$$

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

$$\mathbf{L}_{\mathbf{g}}\phi_i = \nabla\phi_i^{\mathrm{T}}\,\mathbf{g} = (\partial\phi_i/\partial\dot{\mathbf{q}})^{\mathrm{T}} \tag{24}$$

$$L_f\phi_i = \nabla\phi_i^{\mathrm{T}}\,\mathbf{f} = (\partial\phi_i/\partial\mathbf{q})^{\mathrm{T}}\ \dot{\mathbf{q}} + (\partial\phi_i/\partial\dot{\mathbf{q}})^{\mathrm{T}}\ \mathbf{d}_c. \tag{25}$$

4.3. Level 1: Constraints

4.3.1. Force model

The first level includes the equality and inequality constraints required to properly accomplish the surface contact conditioning. These constraints are defined below depending on the vector of forces and torques between the tool and the object being conditioned, which are measured by a force sensor located at the robot end-effector. In many applications, the interaction forces between the tool and the objects in its environment can be approximated by the ideal elastic model below [19]:

$$\mathbf{F}(\mathbf{q}, t) = \mathbf{K}_s\,\Delta\mathbf{s}(\mathbf{q}, \mathbf{p}_s) = \begin{bmatrix}F_x & F_y & F_z & F_\alpha & F_\beta & F_\gamma\end{bmatrix}^{\mathrm{T}}, \tag{26}$$

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

4.3.2. Equality constraints

Three equality constraints are defined as follows:

$$\sigma_{eq,z}(\mathbf{q},t) = F_z - F_{z,ref} = 0 \tag{27}$$

$$\sigma_{eq,\alpha}(\mathbf{q},t) = F_\alpha = 0 \tag{28}$$

$$\sigma_{eq,\beta}(\mathbf{q},t) = F_\beta = 0, \tag{29}$$

where the first equality constraint is used to attain the desired force $F_{z,ref}$ between the tool and the surface in the Z_n-axis, where subscript n is used to denote the tool coordinate system, whereas the last two equality constraints are used to keep the tool orientation perpendicular to the surface, since the torques in X_n- and Y_n-axes (i.e., F_α and F_β) are zero if the tool is perfectly perpendicular to the surface. It is worth noting that if the proposed application for surface contact conditioning is extended to more aggressive processes from a force-standpoint (e.g., material removal), equality constraints (28) and (29) could be ineffective since the planar forces exerted could become significant and may generate misalignment of the tool on the surface.

Taking into account (18), (24) and (26)–(29), the Lie derivative $\mathbf{L_g}\boldsymbol{\phi}_{eq}$ required for the SMC in (16) is given by:

$$\begin{aligned}\mathbf{L_g}\boldsymbol{\phi}_{eq} &= (\partial\boldsymbol{\phi}_{eq}/\partial\dot{\mathbf{q}})^{\mathrm{T}} = \mathbf{K}_{eq}\,(\partial\boldsymbol{\sigma}_{eq}/\partial\mathbf{q})^{\mathrm{T}} \\ &= \mathbf{K}_{eq}\begin{bmatrix}0&0&1&0&0&0\\0&0&0&1&0&0\\0&0&0&0&1&0\end{bmatrix}\mathbf{K}_s\mathbf{J}_n = \mathbf{K}_{eq}\,\mathbf{H}_{eq}\,\mathbf{K}_s\,\mathbf{J}_n,\end{aligned} \tag{30}$$

where $\boldsymbol{\sigma}_{eq}$ is a column vector composed of all equality constraints $\sigma_{eq,i}$ and $\mathbf{K}_{eq}$ is a diagonal matrix composed of all parameters $K_{eq,i}$.

4.3.3. Inequality constraints

In order to adapt the tool position to the surface obstacles (e.g., protrusions, screws, nuts, etc.), two inequality constraints are defined as follows:

$$\sigma_{in,x}(\mathbf{q},t) = |F_x| - F_{x,\max} \leq 0 \tag{31}$$

$$\sigma_{in,y}(\mathbf{q},t) = |F_y| - F_{y,\max} \leq 0, \tag{32}$$

where $F_{x,\max}$ and $F_{y,\max}$ are the maximum allowed values for the forces in the X_n- and Y_n-axes, respectively.

Taking into account (19), (24) and (31)–(32), the Lie derivative $\mathbf{L_g}\boldsymbol{\phi}_{in}$ required for the SMC in (16) is given by:

$$\begin{aligned} \mathbf{L_g}\boldsymbol{\phi}_{in} &= (\partial\boldsymbol{\phi}_{in}/\partial\dot{\mathbf{q}})^{\mathrm{T}} = \mathbf{K}_{in}\,(\partial\boldsymbol{\sigma}_{in}/\partial\mathbf{q})^{\mathrm{T}} \\ &= \mathbf{K}_{in}\begin{bmatrix} \mathrm{sign}(F_x) & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathrm{sign}(F_y) & 0 & 0 & 0 & 0 \end{bmatrix}\mathbf{K}_s\mathbf{J}_n = \mathbf{K}_{in}\mathbf{H}_{in}\mathbf{K}_s\mathbf{J}_n, \end{aligned} \tag{33}$$

where $\boldsymbol{\sigma}_{in}$ is a column vector composed of all inequality constraints $\sigma_{in,i}$ and $\mathbf{K}_{in}$ is a diagonal matrix composed of the parameters $K_{in,i}$.

4.3.4. Acceleration equality for Level 1

Since the stiffness coefficients $\mathbf{K}_s$ in $\mathbf{L_g}\boldsymbol{\phi}_{eq}$ (30) and $\mathbf{L_g}\boldsymbol{\phi}_{in}$ (33) may not be known, they can be included without loss of generality in the switching gain weight matrix $\mathbf{W}$, so that the SMC given by (16) is modified as follows:

$$\begin{aligned} \begin{bmatrix} \mathbf{K}_{eq}\mathbf{H}_{eq} \\ \mathrm{v2dm}\left(\mathrm{pos}\left(\boldsymbol{\phi}_{in}\right)\right)\mathbf{K}_{in}\mathbf{H}_{in} \end{bmatrix}\mathbf{J}_n\ddot{\mathbf{q}}_c &= -\begin{bmatrix} \overline{\mathbf{W}}_{eq} & \mathbf{0} \\ \mathbf{0} & \overline{\mathbf{W}}_{in} \end{bmatrix}\begin{bmatrix} \mathrm{sign}(\boldsymbol{\phi}_{eq}) \\ \mathrm{pos}(\boldsymbol{\phi}_{in}) \end{bmatrix} u^+ \\ &= -\overline{\mathbf{W}}\begin{bmatrix} \mathrm{sign}(\boldsymbol{\phi}_{eq}) \\ \mathrm{pos}(\boldsymbol{\phi}_{in}) \end{bmatrix} u^+ \;\rightarrow\; \mathbf{A}_1\ddot{\mathbf{q}}_c = \mathbf{b}_1, \end{aligned} \tag{34}$$

where $\mathbf{A}_1$ and $\mathbf{b}_1$ are the matrix and vector for the first task in (6) and:

$$\overline{\mathbf{W}}_{eq} = \begin{bmatrix} W_{eq,z}/K_{s,z} & 0 & 0 \\ 0 & W_{eq,\alpha}/K_{s,\alpha} & 0 \\ 0 & 0 & W_{eq,\beta}/K_{s,\beta} \end{bmatrix} = \begin{bmatrix} \overline{W}_{eq,z} & 0 & 0 \\ 0 & \overline{W}_{eq,\alpha} & 0 \\ 0 & 0 & \overline{W}_{eq,\beta} \end{bmatrix} \tag{35}$$

$$\overline{\mathbf{W}}_{in} = \begin{bmatrix} W_{in,x}/K_{s,x} & 0 \\ 0 & W_{in,y}/K_{s,y} \end{bmatrix} = \begin{bmatrix} \overline{W}_{in,x} & 0 \\ 0 & \overline{W}_{in,y} \end{bmatrix}, \tag{36}$$

where $\{K_{s,x}, K_{s,y}, K_{s,z}\}$ are the stiffness coefficients for the linear X_n-, Y_n- and Z_n-axes, respectively, and $\{K_{s,\alpha}, K_{s,\beta}\}$ are the stiffness coefficients for the rotational X_n- and Y_n-axes, respectively.

Note that the SMC given by (34) only requires: the control parameters $\{u^+, \overline{W}_{eq,i}, \overline{W}_{in,i}, K_{eq,i}, K_{in,i}\}$; the robot Jacobian $\mathbf{J}$; and the constraint functions $\{\phi_{eq,z}, \phi_{eq,\alpha}, \phi_{eq,\beta}, \phi_{in,x}, \phi_{in,y}\}$, which are computed from the force sensor measurements $\{F_z, F_\alpha, F_\beta, F_x, F_y\}$ and their derivatives.

4.4. Level 2: Reference tracking

The following equality is considered for this level:

$$\begin{aligned} &\mathbf{J}\ddot{\mathbf{q}}_c = \ddot{\mathbf{p}}_{ref} + K_{T,v}\dot{\mathbf{e}} + K_{T,p}\mathbf{e} + \mathrm{sign}(\dot{\mathbf{e}} + K_{T,p}K_{T,v}^{-1}\mathbf{e})u_T^+ \\ &\rightarrow \ \mathbf{A}_2\ddot{\mathbf{q}}_c = \mathbf{b}_2, \end{aligned} \tag{37}$$

where $\mathbf{p}_{ref}$ is the reference for the tool pose, $\mathbf{e} = \mathbf{p}_{ref} - \mathbf{p}$ is the tool pose error, $K_{T,p}$ and $K_{T,v}$ are the correction gains for the position and velocity errors, respectively, the tool speed $\dot{\mathbf{p}}$ is obtained from the first-order kinematics (2), u_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 (6). In this hybrid control law the switching term represents a conventional SMC used to compensate the term $\dot{\mathbf{J}}\dot{\mathbf{q}}$ of the robot second-order kinematics (3), which yields two advantages: the Jacobian derivative is not required; and, due to the other continuous terms in the control action, the switching gain u_3^+ can be relatively small, reducing the chattering effects.

4.5. 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) since otherwise there are no remaining degrees of freedom at this level. While there are various options described in the literature [25, 26] to use the robot's redundant degrees of freedom, this work considers "pushing" the robot to a home configuration $\mathbf{q}_0$ so as to avoid, to a certain extent, possible critical areas due to e.g. joint limits, singular configurations or possible obstacles in the robot workspace. While doing so adds a degree of safety, it is clear that it can not completely guarantee avoidance of critical areas, especially in complex robot cells, and it is not aimed as a substitute for high-level planning.

Therefore, the following equality is considered:

$$\begin{aligned} & \ddot{\mathbf{q}}_c = -K_{3,v}\dot{\mathbf{q}} + K_{3,p}(\mathbf{q}_0 - \mathbf{q}) \\ \rightarrow \; & \mathbf{A}_3\ddot{\mathbf{q}}_c = \mathbf{b}_3, \end{aligned} \tag{38}$$

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 second task in (6).

4.6. Additional remarks

4.6.1. 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. 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. Stability

The stability of the SMC in the first task is guaranteed if u^+ fullfills (17) and matrix $\mathbf{L_g}\boldsymbol{\phi}$ (excluding the rows of the non-active inequality constraints) is full row rank. That is, taking into account (34), 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 level, which represent kinematic and inverse kinematic control algorithms, the reader is referred to [27], where the stability of this kind of algorithms is analyzed in a task prioritization framework.

4.6.3. 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 first-order low-pass filter, which is implemented in the sensor electronics.

It is important to remark that the attenuation of the low-pass filter at the frequency of the SM control action, which is equal to $(2T_s)^{-1}$ Hz (i.e., theoretically the SM control action switches its value from positive to negative, or vice versa, every sampling time), should be relatively small. Otherwise, changes in the SM control action would not be properly "followed" by the filtered measurement signal, causing instability in the system. Moreover, the bandwidth of the kinematic control performed in Level 1 (given by $K_{eq,i}$ and $K_{in,i}$), Level 2 (given by $K_{T,p}$ and $K_{T,v}$) and Level 3 (given by $K_{3,p}$ and $K_{3,v}$) should be significantly lower than the SM frequency for stability reasons. Note that the bandwidth of the controlled system corresponds to the bandwidth of the aforementioned kinematic controller and not to the frequency of the SM control action.

4.7. Controller implementation

The pseudo-code of the proposed method is shown below, which used the following auxiliary functions:

- Kinematic function $\mathbf{l}(\mathbf{q})$ and Jacobian matrices $\mathbf{J}$ and $\mathbf{J}_n$.
- Moore-Penrose pseudoinverse $(\cdot)^\dagger$, using a tolerance to set to zero the very small singular values [28].
- Symbol $\circ$ denotes the element-wise or Hadamard product.
- Robot sensors: *GetRobotStateAndForce*, which returns the current robot state $\{\mathbf{q}, \dot{\mathbf{q}}\}$ and the current force $\mathbf{F}$ measured by the sensor located at the robot tool, which has already been filtered by the sensor electronics.
- Actuators: *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 using compiled C code in a computer with Intel Core i5-3470 processor at 3.2 GHz clock frequency was around 15 microseconds for the experiments in next sections.

Algorithm executed at sampling time of T_s seconds

```
$[\mathbf{q}, \dot{\mathbf{q}}, \mathbf{F}]$ =GetRobotStateAndForce;
$\mathbf{p} = \mathbf{l}(\mathbf{q})$ ;                                   // Eq. (1)
$\dot{\mathbf{p}} = \mathbf{J}\dot{\mathbf{q}}$ ;                                   // Eq. (2)
$\dot{\mathbf{p}}_{ref} = (\mathbf{p}_{ref} - \mathbf{p}_{ref,prev})/T_s$ ;                // Derivative
$\ddot{\mathbf{p}}_{ref} = (\dot{\mathbf{p}}_{ref} - \dot{\mathbf{p}}_{ref,prev})/T_s$ ;                // Derivative
$\dot{\mathbf{F}} = (\mathbf{F} - \mathbf{F}_{prev})/T_s$ ;                        // Derivative
$\boldsymbol{\phi}_{eq} = \begin{bmatrix} F_z - F_{z,ref} + K_{eq,z}\dot{F}_z \\ F_\alpha + K_{eq,\alpha}\dot{F}_\alpha \\ F_\beta + K_{eq,\beta}\dot{F}_\beta \end{bmatrix}$ ;          // Eqs. (18),(27)-(29)
$\boldsymbol{\phi}_{in} = \begin{bmatrix} |F_x| - F_{x,\max} + K_{in,x}\text{sign}(F_x)\dot{F}_x \\ |F_y| - F_{y,\max} + K_{in,y}\text{sign}(F_y)\dot{F}_y \end{bmatrix}$ ;    // Eqs. (19),(31)-(32)
$\overline{\mathbf{W}}_{eq} = \left|\overline{\mathbf{W}}_{eq,prev} + T_s\,\mu_{eq}\,\text{sign}(\boldsymbol{\phi}_{eq}) \circ \text{sign}(\boldsymbol{\phi}_{eq,prev})\right|$ ;    // Eq. (21)
$\overline{\mathbf{W}}_{in} = \left|\overline{\mathbf{W}}_{in,prev} + T_s\,\mu_{in}\,f_{ASG}(\boldsymbol{\phi}_{in}, \boldsymbol{\phi}_{eq,prev})\right|$ ;         // Eq. (22)
$\mathbf{A}_1 = \begin{bmatrix} \mathbf{K}_{eq}\mathbf{H}_{eq} \\ \text{v2dm}\left(\text{pos}\left(\boldsymbol{\phi}_{in}\right)\right)\mathbf{K}_{in}\mathbf{H}_{in} \end{bmatrix}\mathbf{J}_n$ ;            // Eq. (34)
$\mathbf{b}_1 = -\overline{\mathbf{W}}\begin{bmatrix} \text{sign}(\boldsymbol{\phi}_{eq}) \\ \text{pos}(\boldsymbol{\phi}_{in}) \end{bmatrix} u^+$ ;                        // Eq. (34)
$\mathbf{A}_2 = \mathbf{J}$ ;                                    // Eq. (37)
$\mathbf{b}_2 = \ddot{\mathbf{p}}_{ref} + K_{T,v}\dot{\mathbf{e}} + K_{T,p}\mathbf{e} + \text{sign}(\dot{\mathbf{e}} + K_{T,p}K_{T,v}^{-1}\mathbf{e})u_T^+$ ;    // Eq. (37)
$\mathbf{A}_3 = \mathbf{I}$ ;                                    // Eq. (38)
$\mathbf{b}_3 = -K_{3,v}\dot{\mathbf{q}} + K_{3,p}(\mathbf{q}_0 - \mathbf{q})$ ;                   // Eq. (38)
$\ddot{\mathbf{q}}_{c,1} = \mathbf{A}_1^\dagger\mathbf{b}_1$ ;                             // Eq. (7), $i = 1$
$\mathbf{N}_1 = \mathbf{I} - \mathbf{A}_1^\dagger\mathbf{A}_1$ ;                          // Eq. (8), $i = 1$
$\ddot{\mathbf{q}}_{c,2} = \ddot{\mathbf{q}}_{c,1} + (\mathbf{A}_2\mathbf{N}_1)^\dagger(\mathbf{b}_2 - \mathbf{A}_2\ddot{\mathbf{q}}_{c,1})$ ;        // Eq. (7), $i = 2$
$\mathbf{N}_2 = \mathbf{N}_1(\mathbf{I} - (\mathbf{A}_2\mathbf{N}_1)^\dagger(\mathbf{A}_2\mathbf{N}_1))$ ;            // Eq. (8), $i = 2$
$\ddot{\mathbf{q}}_{c,3} = \ddot{\mathbf{q}}_{c,2} + (\mathbf{A}_3\mathbf{N}_2)^\dagger(\mathbf{b}_3 - \mathbf{A}_3\ddot{\mathbf{q}}_{c,2})$ ;        // Eq. (7), $i = 3$
SendToJointControllers($\ddot{\mathbf{q}}_{c,3}$);
$\mathbf{p}_{ref,prev} = \mathbf{p}_{ref}$ ;                        // For next iteration
$\dot{\mathbf{p}}_{ref,prev} = \dot{\mathbf{p}}_{ref}$ ;                        // For next iteration
$\mathbf{F}_{prev} = \mathbf{F}$ ;                               // For next iteration
$\boldsymbol{\phi}_{eq,prev} = \boldsymbol{\phi}_{eq}$ ;                        // For next iteration
$\boldsymbol{\phi}_{in,prev} = \boldsymbol{\phi}_{in}$ ;                        // For next iteration
$\overline{\mathbf{W}}_{eq,prev} = \overline{\mathbf{W}}_{eq}$ ;                      // For next iteration
$\overline{\mathbf{W}}_{in,prev} = \overline{\mathbf{W}}_{in}$ ;                      // For next iteration
```

5. Adaptation of the classical admittance control approach for the proposed application

For comparison purposes, the classical admittance control (CAC) typically used in force control tasks [5, 6, 7] is adapted below for the proposed surface contact conditioning. The general form of the admittance controller is given by:

$$\mathbf{M}_d\,\ddot{\mathbf{e}}_n + \mathbf{C}_d\,\dot{\mathbf{e}}_n + \mathbf{K}_d\,\mathbf{e}_n = -\Delta\mathbf{F}, \tag{39}$$

where $\mathbf{e}_n = \mathbf{p}_{n,ref} - \mathbf{p}_n = (\mathbf{L}\,\mathbf{R})^{-1}(\mathbf{p}_{ref} - \mathbf{p})$ is the error relative to the tool coordinate system, $\Delta\mathbf{F}$ is the external force vector relative to the tool coordinate system and the controller gains $\mathbf{M}_d$, $\mathbf{C}_d$ and $\mathbf{K}_d$ are 6×6 diagonal matrices representing the virtual inertia, virtual damping and virtual stiffness, respectively.

The following parameters are considered to adapt (39) for the proposed surface contact conditioning:

$$\mathbf{K}_d = \mathrm{diag}\left(\begin{bmatrix} K_{d,x} & K_{d,y} & 0 & 0 & 0 & K_{d,\gamma}\end{bmatrix}\right) \tag{40}$$

$$\dot{\mathbf{p}}_{n,ref} = \begin{bmatrix} \dot{p}_{n,ref,x} & \dot{p}_{n,ref,y} & 0 & 0 & 0 & \dot{p}_{n,ref,\gamma}\end{bmatrix}^{\mathrm{T}} \tag{41}$$

$$\ddot{\mathbf{p}}_{n,ref} = \begin{bmatrix} \ddot{p}_{n,ref,x} & \ddot{p}_{n,ref,y} & 0 & 0 & 0 & \ddot{p}_{n,ref,\gamma}\end{bmatrix}^{\mathrm{T}} \tag{42}$$

$$\Delta\mathbf{F} = \mathbf{F} - \begin{bmatrix} 0 & 0 & F_{z,ref} & 0 & 0 & 0\end{bmatrix}^{\mathrm{T}}, \tag{43}$$

where function diag($\cdot$) gives a diagonal matrix from a vector and $\dot{\mathbf{p}}_{n,ref} = (\mathbf{L}\,\mathbf{R})^{-1}\,\dot{\mathbf{p}}_{ref}$ and $\ddot{\mathbf{p}}_{n,ref} = (\mathbf{L}\,\mathbf{R})^{-1}\,(\ddot{\mathbf{p}}_{ref} - \dot{\mathbf{L}}\,\mathbf{R}\,\mathbf{J}_n\,\dot{\mathbf{q}})$ are the reference velocities and accelerations, respectively, relative to the tool coordinated system. Note that (40) gives rise to a position control in the linear X_n-, Y_n- and rotational Z_n-axes together with a velocity control in the remaining axes. The former allows to track the reference signals in the mentioned axes, while the latter allows to adapt the tool speed in the linear Z_n- and rotational X_n- and Y_n-axes according to the detected forces in these axes. Note also that (41)–(43) yield that the equilibrium point for the mentioned velocity control is given by $F_z = F_{z,ref}$, $F_\alpha = 0$ and $F_\beta = 0$, respectively, as required by the surface contact application, see (27)–(28).

Taking into account the first- and second-order kinematics of the robot system (2)–(4), the admittance force control given by (39) yields the following

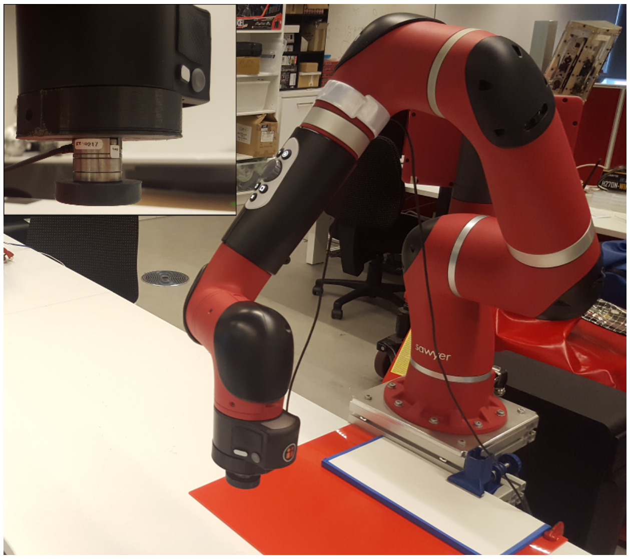


Fig. 1. Experimental setup: 7R serial manipulator with a force sensor rigidly attached to the robot end-effector, a tool consisting of a cylinder of 43x43x10mm attached to the sensor to emulate a tool and an object with flat flexible surface as target.

acceleration equality:

$$\begin{aligned}\mathbf{M}_d\,\mathbf{J}_n\,\ddot{\mathbf{q}}_c =& \Delta\mathbf{F} + \mathbf{K}_d\,\mathbf{e}_n - (\mathbf{C}_d\,\mathbf{J}_n + \mathbf{M}_d\,\dot{\mathbf{J}}_n)\dot{\mathbf{q}} \\ & + \mathbf{C}_d\,\dot{\mathbf{p}}_{n,ref} + \mathbf{M}_d\,\ddot{\mathbf{p}}_{n,ref} \\ \rightarrow \quad \mathbf{A}_1\ddot{\mathbf{q}}_c =& \mathbf{b}_1, \end{aligned} \tag{44}$$

which replaces both Level 1 and Level 2 of the proposed method, see Section 4.3.4 and Section 4.4.

6. Experiments for the proposed method

In this section, several experiments are shown to demonstrate the effectiveness of the proposed approach. In particular, four tests have been conducted: three comparison experiments to analyze the performance of the proposed adaptive non-conventional SMC; and a fourth dynamic experiment to show the capabilities of the proposed robot control for surface contact conditioning using the adaptive conventional and non-conventional SMC.

6.1. Experiment conditions

The setup used for the experiments consists of (see Fig. 1): a Sawyer collaborative robot; a Force/Torque Sensor Nano25 SI-25-25 attached to the robot end-effector; a simple surface profile tool consisting of a cylinder of 43x43x10mm attached to the sensor to demonstrate the ability of the robot controller to track the contour of the surface with the tool, keeping permanent contact with the surface; and a flat flexible surface target. The experiments shown below were run under the following conditions:

i) The control period T_s is set to 20 milliseconds.

ii) 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.

iii) The force sensor signal is filtered using a discrete first-order low-pass filter with a cutoff frequency of 73 Hz, which is implemented in the sensor electronics (Net F/T interface).

iv) Two kind of objects have been used: a polycarbonate flat object of 400x400x3mm used as the surface to be treated, see the red surface on the table in Fig. 1, which 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; and two polycarbonate flat objects of 230x300x10mm (white object in Fig. 1) and 160x300x10mm (black object in Fig. 9) used as obstacles, which represent e.g. a protrusion on a surface, to be circumvented while the robot force is constrained to maintain contact with the surface.

v) Parameters used for the Level 1 described in Section 4.3: parameters of the constraint functions $K_{eq,i} = K_{in,i} = 0.15$, $F_{z,ref} = -15$ N, $F_{x,\max} = F_{y,\max} = 20$ N; switching gain given by $u^+ = 0.06$, $\{\overline{W}_{eq,z} = 0.33, \overline{W}_{eq,\alpha} = \overline{W}_{eq,\beta} = 16\}$ (parameter $\overline{W}_{in,i}$ depends on the considered experiment and can be seen in the corresponding figure) and $\mu_{eq,i} = \mu_{in,i} = 0.02$. (The values for the switching gain have been empirically selected for the flat object resting on a table, as shown in Fig. 1.)

vi) Parameters used for the Level 2 described in Section 4.4: parameters of the tracking controller $K_{T,v} = 5.0$, $K_{T,p} = 3.0$ and $u_T^+ = 0.01$; and a reference trajectory given by $\gamma_{ref} = -\pi/2$ and a 2D circle of radius 80mm and period 10 seconds.

vii) Parameters used for the Level 3 described in Section 4.5: parameters of the PD controller $K_{3,v} = 0.5$ and $K_{3,p} = 1.0$; and home configuration $\mathbf{q}_0 = [0.003, -0.577, 0.002, 2.044, -0.067, 0.104, 3.382]^{\mathrm{T}}$ rad.

6.2. First comparison experiment: adaptive non-conventional SMC depending on the position of the obstacles along the system trajectory

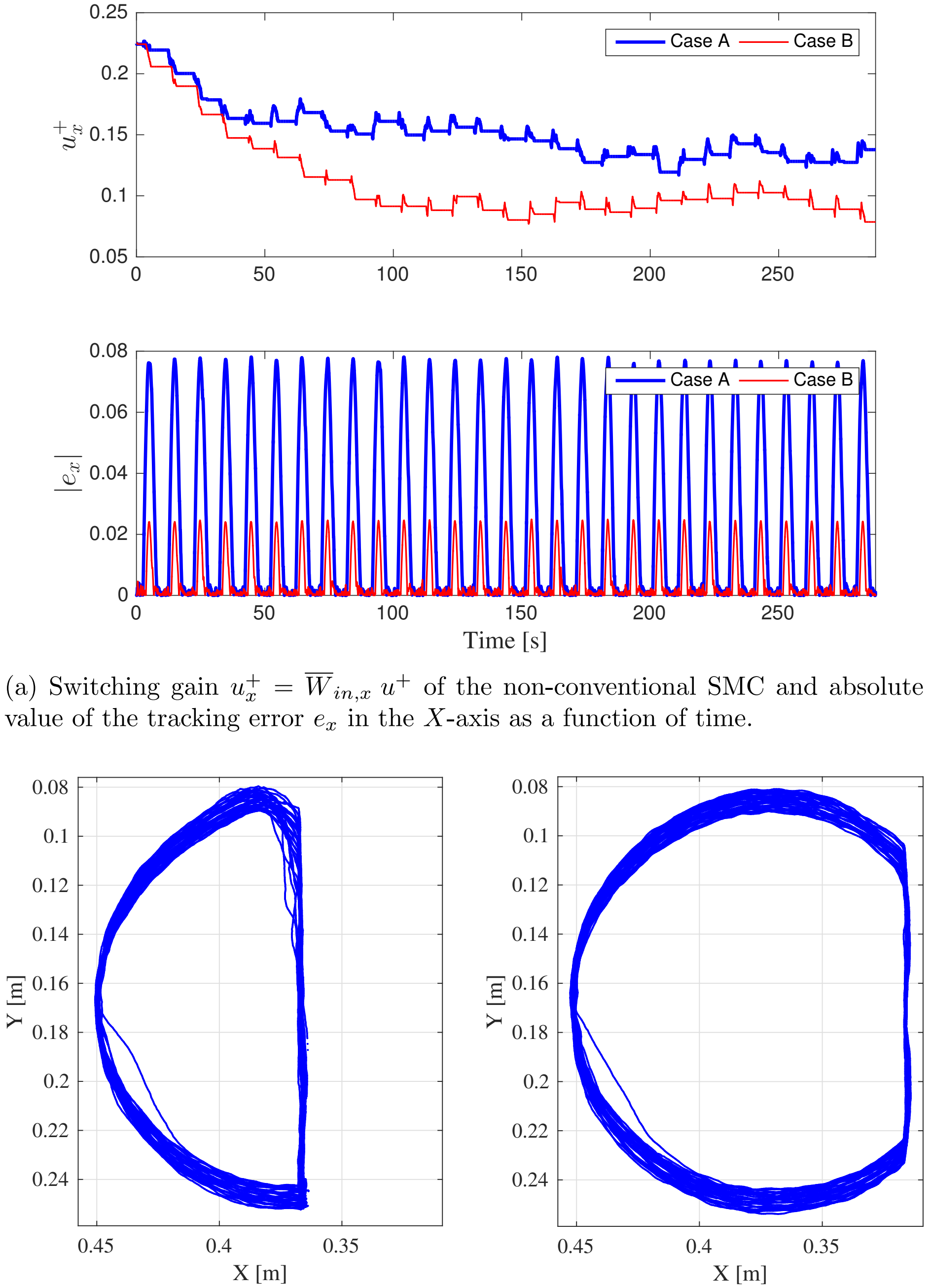
For the first comparison experiment, two different positions of the obstacle have been considered: for case A the obstacle is located at the mid-point of the circular reference trajectory, whereas for case B the obstacle is placed at 3/4 of the trajectory, see Fig. 2b. The comparison in Fig. 2a shows that the switching gain for case A converges to a larger value than that for case B. This is because case A has to compensate a larger control action of the tracking controller due to a larger tracking error.

6.3. Second comparison experiment: adaptive non-conventional SMC for different initial values.

For the second comparison experiment the obstacle is located at the mid-point of the circular reference, as in case A above, and three different values are considered for the initial switching gain $u_x^+(0)$: 0.165, 0.225 and 0.069. Fig. 3 shows that for all three cases the switching gain converges around 0.138 for this particular position of the obstacle, which demonstrates the repeatability of the proposed ASG method.

6.4. Third comparison experiment: FSG vs. ASG for the non-conventional SMC

For the third comparison experiment the obstacle is located again at the mid-point of the circular reference (see Fig. 4b) and two options are considered for the switching gain: a fixed switching gain (FSG) and the proposed ASG method, both with the same initial value 0.069, see Fig. 4a. A video of this comparison can be played at https://media.upv.es/player/?id=1c4d5930-37f0-11e8-a92e-a3483ec58dec, which shows a better performance for the proposed ASG method in terms of smoothness. This fact can be noted in the path followed by the robot tool in Fig. 4b, which is thinner for the proposed ASG in the part where the robot tool is in contact with



(a) Switching gain $u_x^+ = \overline{W}_{in,x}\, u^+$ of the non-conventional SMC and absolute value of the tracking error e_x in the X-axis as a function of time.

(b) Top view of the path followed by the robot tool. Left, case A (obstacle at the mid-point of the circular reference); Right, case B (obstacle at 3/4 of the circular reference).

Fig. 2. First comparison experiment: Adaptive non-conventional SMC for different obstacle positions.

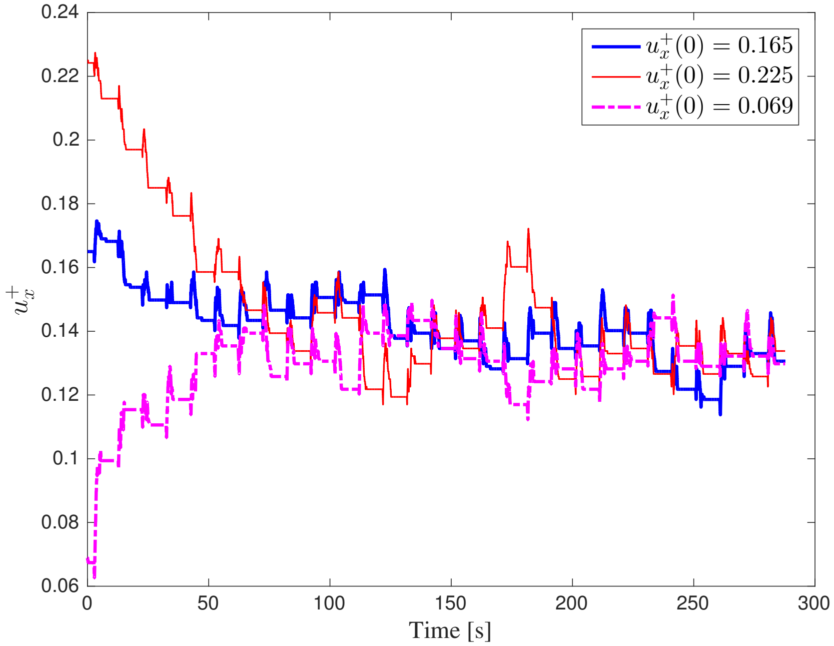
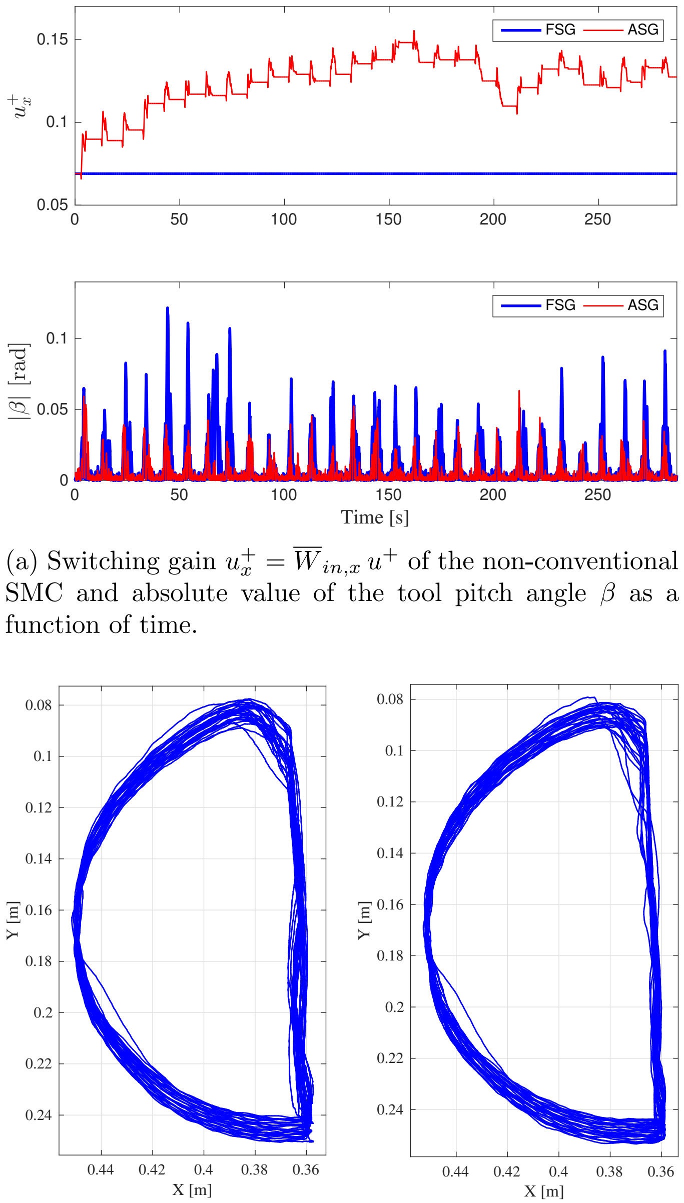


Fig. 3. Second comparison experiment: switching gain $u_x^+ = \overline{W}_{in,x}\, u^+$ of the non-conventional SMC as a function of time for three different initial values to demonstrate the repeatability of the of the proposed ASG method.

the obstacle; and it can also be noted in the pitch angle β of the robot tool, see Fig. 4a, which has smaller deviations for the proposed ASG than for the conventional FSG.

6.5. Dynamic experiment: adaptive conventional and non-conventional SMC

Finally, in order to verify the robustness and performance of the proposed approach, a more dynamic experiment has been conducted introducing different perturbations during the task: changes in the position, orientation and stiffness of the flat object together with dynamic obstacles along the trajectory. The video of this dynamic experiment can be played at https://media.upv.es/player/?id=8c451c80-37e8-11e8-a92e-a3483ec58dec and the results are depicted at different figures. Fig. 5 shows the performance of the system in terms of the constraints fulfillment. It can be seen that, the inequality constraints are fulfilled, i.e., $\max(\sigma_{in,i}) \leq 0$, and that the inequality for the X-axis becomes active during seven phases, whereas that for the Y-axis becomes active during nine phases. Note that when the tool comes upon an obstacle (e.g., a



(a) Switching gain $u_x^+ = \overline{W}_{in,x}\, u^+$ of the non-conventional SMC and absolute value of the tool pitch angle β as a function of time.

(b) Top view of the path followed by the robot tool: Left, FSG; Right, ASG.

Fig. 4. Third comparison experiment: FSG vs. ASG.

protrusion on the surface) an inequality constraint becomes active and the tool does not continue in the same direction to prevent the tool breakage. However, the execution of the task is not stopped and the robot tool continues in a direction similar to that given by the reference path and satisfying the active inequality constraint. For instance, around time 36s in the graph (46s in the video) an obstacle has come up in the Y direction, but $\sigma_{in,y}$ does not overcome the zero value, meaning that the inequality is fulfilled and, hence, the force exerted in this direction never exceeds the maximum value $F_{y,\max}$ chosen by the user, protecting thus the integrity of the tool.

It can also be seen in Fig. 5 that the equality constraints $\{\sigma_{eq,z}, \sigma_{eq,\alpha}, \sigma_{eq,\beta}\}$ are switching around zero as expected. This means that the surface contact task is being done properly: the tool orientation is perpendicular to the object surface and the pressure with the tool on the surface is being kept regardless changes in the object position, orientation and stiffness and the obstacles introduced. The figure also shows the variation of the computed switching gain u_z^+ in order to keep the pressure on the object surface regardless the changes in its stiffness. In particular, for the interval 80s–170s in the graph (1m30s–3m00s in the video) the flat object is held above the table and, hence, its stiffness coefficient in the Z-axis is significantly reduced because it has no support. Thus, for this interval the proposed ASG method automatically increases the switching gain u_z^+ in order to properly keep the desired pressure, see the bottom plot in Fig. 5. It can also be seen in this figure that the switching gains u_x^+ and u_y^+ are also modified, although to a lesser extent than u_z^+, when the corresponding inequality constraints become active.

Fig. 6 and Fig. 7 show the trajectory followed by the tool and reference as a function of time and the corresponding 3D representation, respectively. Note that the tracking error in the X- and Y-axes is not zero when such deviations are required to satisfy the inequality constraints, i.e., tracking error arises when an obstacle comes upon the trajectory (see the video), although the robot tool retrieves the reference when the obstacle is overcome.

Fig. 8 shows the commanded accelerations and velocities computed during the experiment, whereas Fig. 9 shows several instants of the experiment: Fig. 9a, 9b and 9c show the robot behavior when the obstacle comes upon; Fig. 9d, 9e, 9f 9g, 9h and 9i, show the system behavior when the flat object is held above the table; and Fig. 9j, 9k and 9l show the robot behavior when the obstacle comes upon and the flat object is held above the table at a different position and orientation.

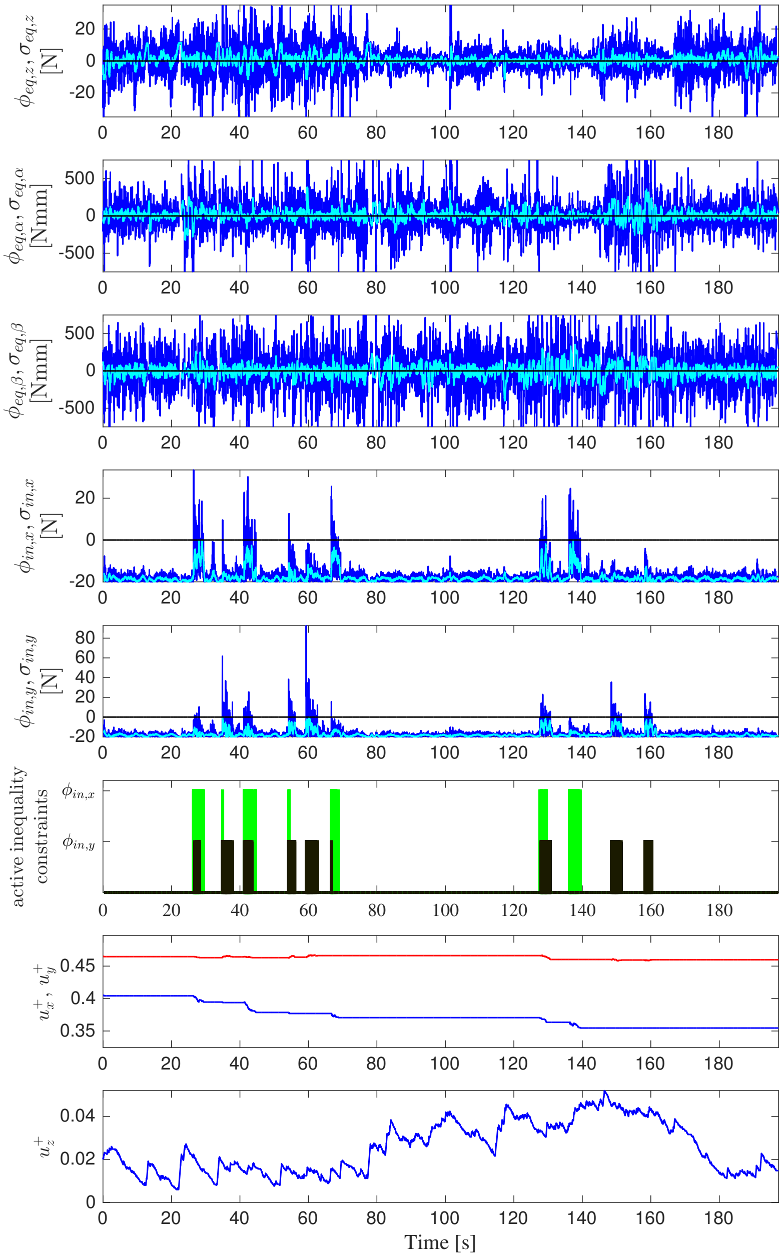


Fig. 5. Dynamic experiment: constraint functions, active inequality constraints and switching gains $\{u_x^+ = \overline{W}_{in,x}u^+, u_y^+ = \overline{W}_{in,y}u^+, u_z^+ = \overline{W}_{eq,z}u^+\}$. The modified constraint function ϕ_i is in dark-blue, whereas the original constraint function σ_i is in light-cyan.

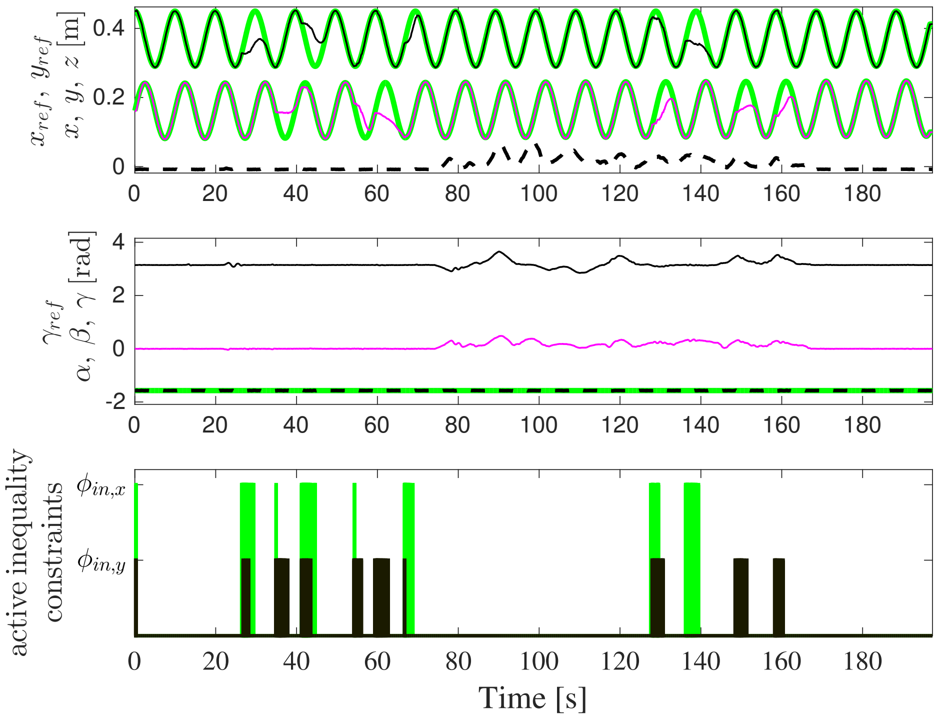


Fig. 6. Dynamic experiment: top, tool position $\{x, y, z\}$ and reference signals $\{x_{ref}, y_{ref}\}$; bottom, tool orientation $\{\alpha, \beta, \gamma\}$ in roll-pitch-yaw angles and reference signal γ_{ref}. The reference signals are in solid-thick line, $\{x, \alpha\}$ in thin-dark line, $\{y, \beta\}$ in thin-light line and $\{z, \gamma\}$ in dashed line.

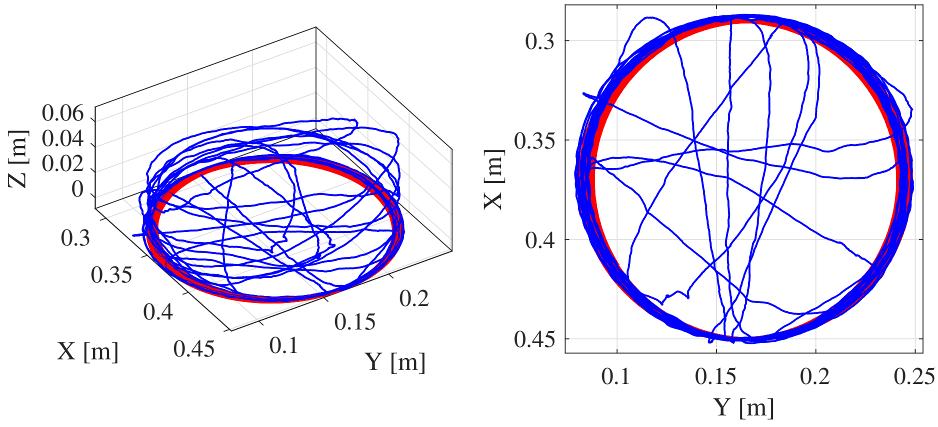


Fig. 7. Dynamic experiment: 3D representation of the trajectory followed by the tool (thin-blue line) and the circular reference trajectory (thick-red line).

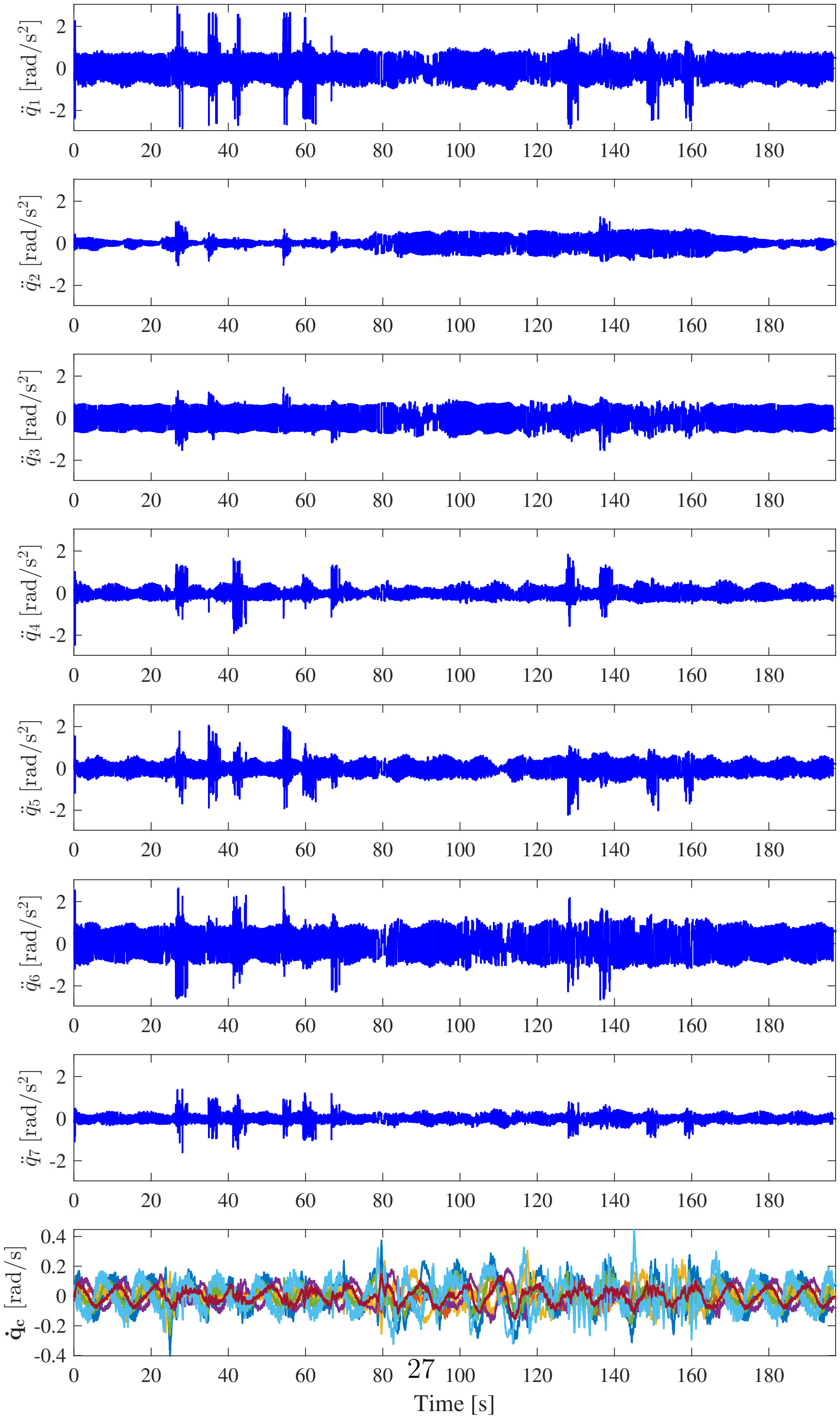
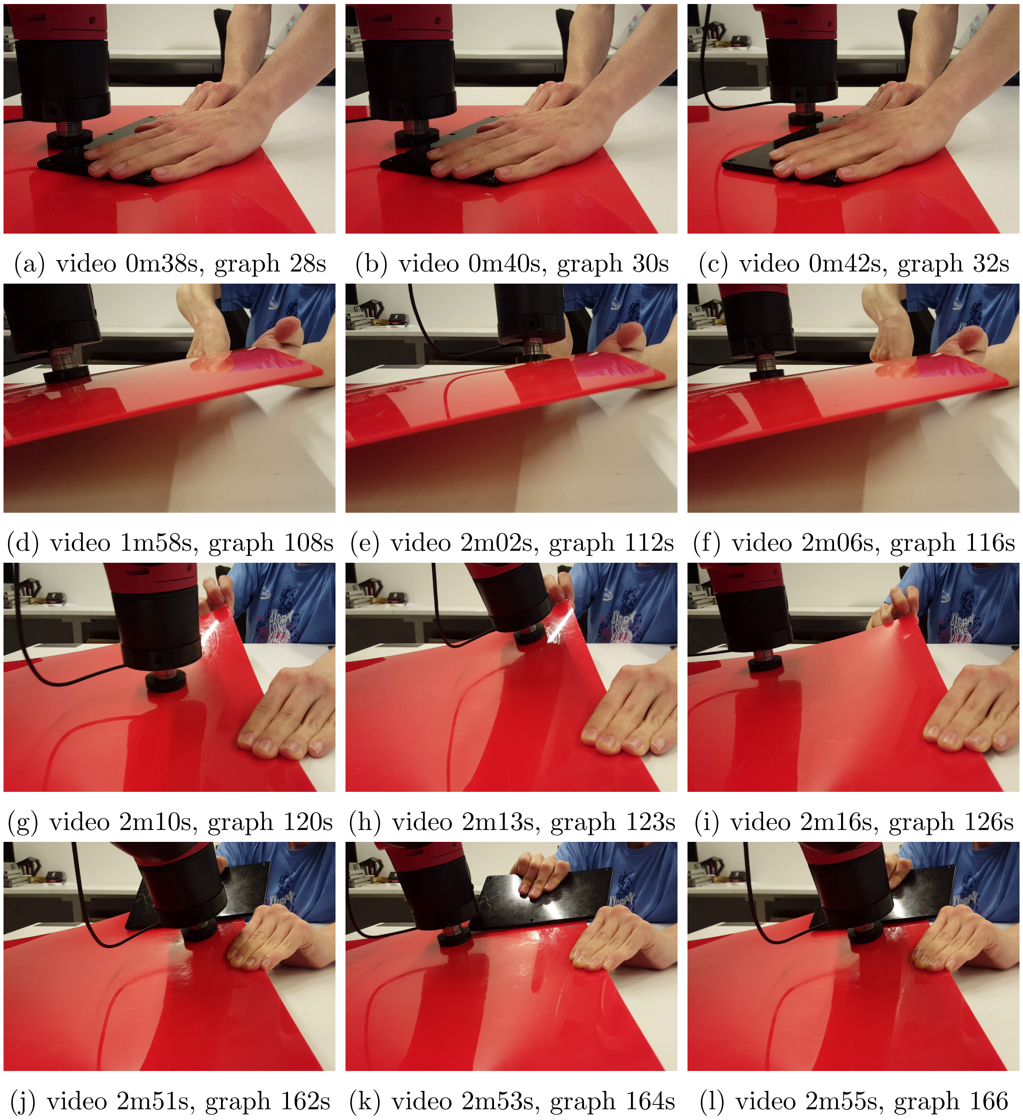


Fig. 8. Dynamic experiment: commanded acceleration for each joint in the top seven plots and commanded joint velocities combined in the bottom plot.



(a) video 0m38s, graph 28s (b) video 0m40s, graph 30s (c) video 0m42s, graph 32s

(d) video 1m58s, graph 108s (e) video 2m02s, graph 112s (f) video 2m06s, graph 116s

(g) video 2m10s, graph 120s (h) video 2m13s, graph 123s (i) video 2m16s, graph 126s

(j) video 2m51s, graph 162s (k) video 2m53s, graph 164s (l) video 2m55s, graph 166

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

6.6. Conclusions from the experiments

From the above experiments the following conclusions can be highlighted:

- The proposed adaptive non-conventional SMC properly tunes the switching gain depending on the position of the obstacles along the system trajectory.
- For constant conditions, the switching gain of the proposed adaptive non-conventional SMC converges to the same value regardless its initial condition, ensuring the repeatability.
- The behavior of the non-conventional SMC with the proposed ASG algorithm is smoother than that obtained with the classical FSG.
- The proposed robot control for surface contact conditioning using the adaptive conventional and non-conventional SMC has a good performance even for the case of dynamic environments.

7. Comparison experiments with the classical admittance control approach

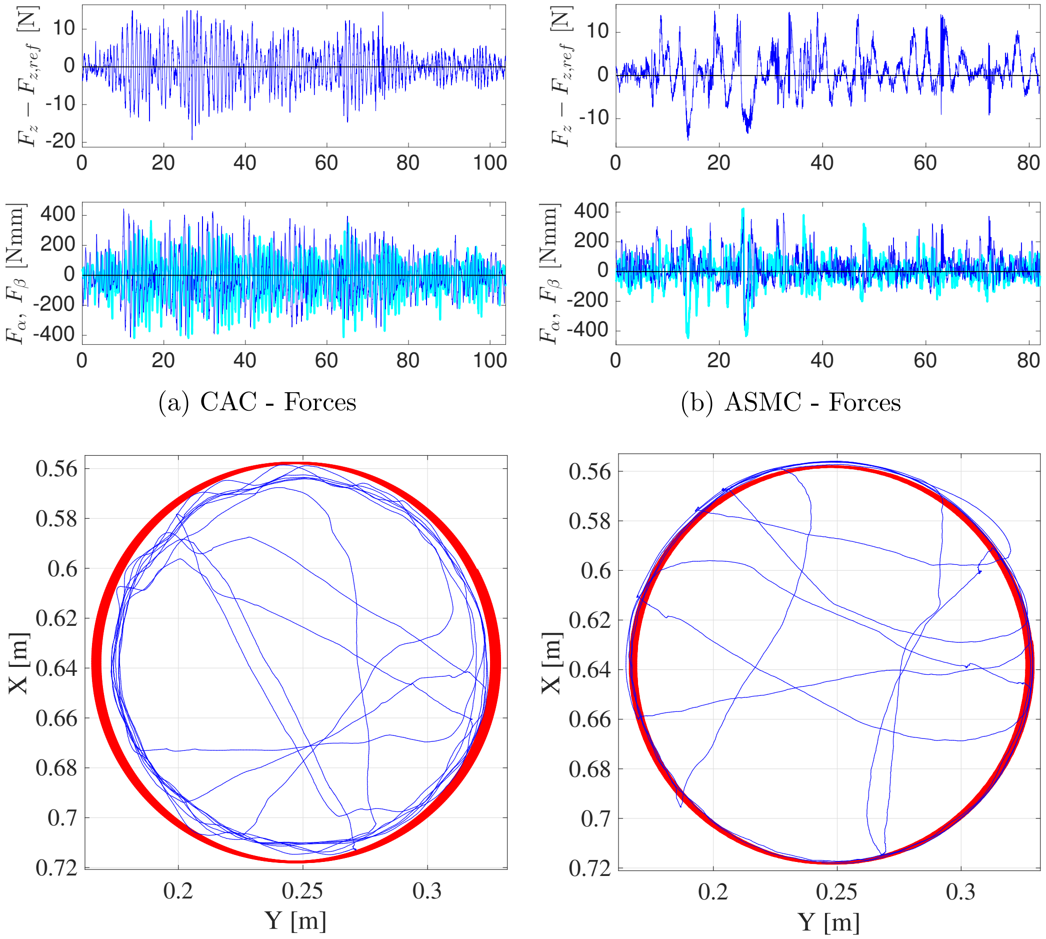
In this section, several experiments are shown to illustrate the advantages of the proposed ASMC method over the Classical Admittance Control (CAC) approach.

7.1. Experiment conditions

On the one hand, the parameter values used for the proposed ASMC are those detailed in Section 6.1. On the other hand, the parameter values used for the CAC described in Section 5 have been empirically tuned to obtain the fastest response, i.e., lowest settling time and highest gain, without compromising the system stability. In particular, the following values have been used: $\mathbf{M}_d = \mathrm{diag}(\begin{bmatrix} 37 & 37 & 48 & 1 & 1 & 1 \end{bmatrix})$; $\mathbf{C}_d = \mathrm{diag}(\begin{bmatrix} 187 & 187 & 240 & 5 & 5 & 5 \end{bmatrix})$ and $\mathbf{K}_d = \mathrm{diag}(\begin{bmatrix} 250 & 250 & 0 & 0 & 0 & 0.5 \end{bmatrix})$.

7.2. Comparison experiment with obstacles

The experiment consists in tracking the circular trajectory with the flat target resting on the table and random dynamic obstacles along the reference trajectory. The video of this experiment can be played https://media.upv.es/player/?id=5ed32580-37e3-11e8-a92e-a3483ec58dec, where it



(a) CAC - Forces

(b) ASMC - Forces

(c) CAC - Trajectories

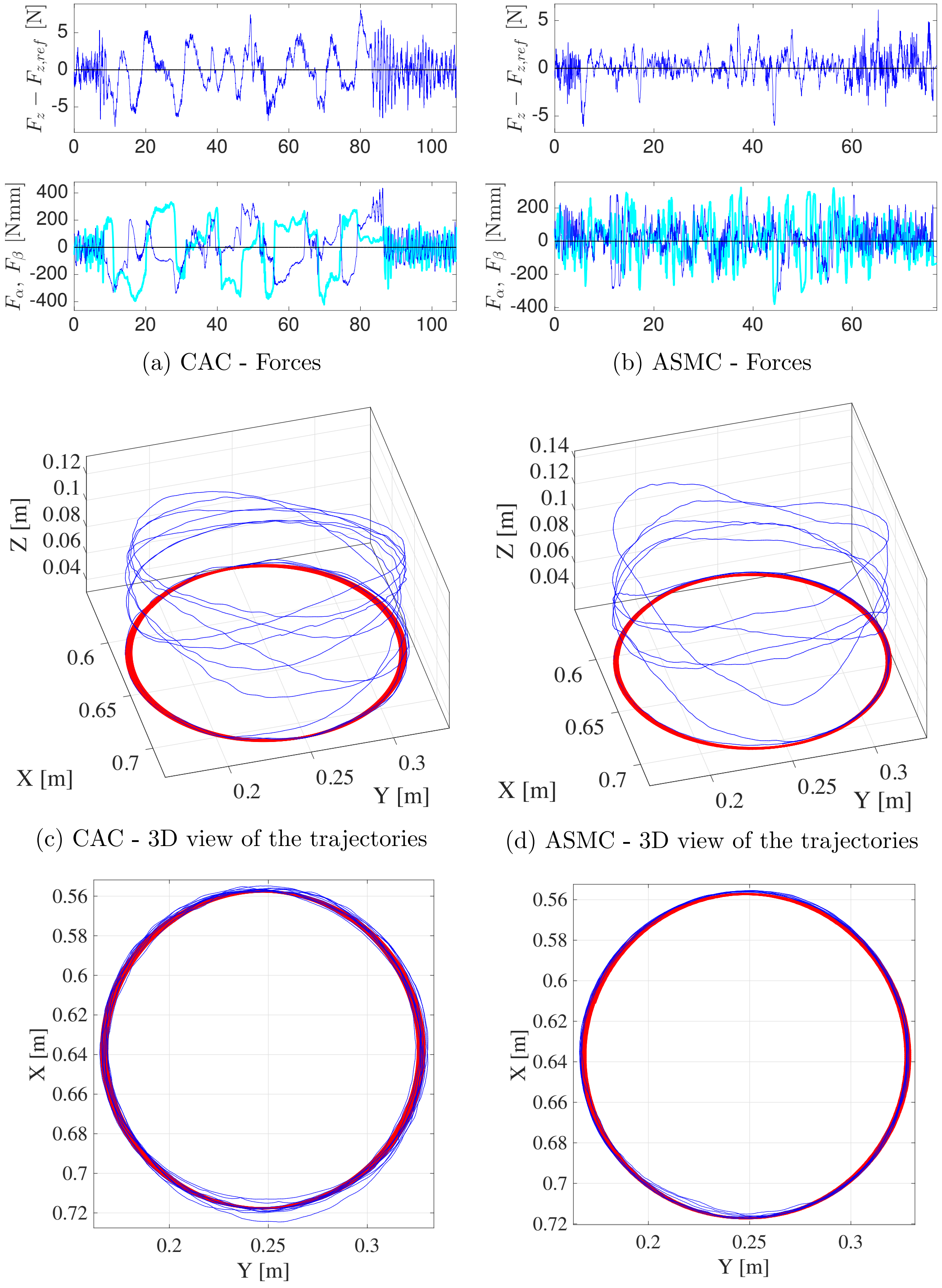
(d) ASMC - Trajectories

Fig. 10. CAC vs. ASMC comparison experiment with obstacles: top, forces (F_α thin-blue line, F_β thick-cyan line); bottom, top view of the trajectory followed by the tool (thin-blue line) and reference (thick-red line).

can be appreciated the behavior of both control approaches. Fig. 10 shows the forces and trajectory followed by the tool for both control approaches. In particular, in both cases the forces are switching with a similar magnitude around the reference values, see Fig. 10a and Fig. 10b, which means that the surface contact task is being done properly: the tool orientation is perpendicular to the object surface and the pressure with the tool on the surface is being kept. However, it can be seen that the behavior of the reference tracking is significantly worse for the CAC approach, see Fig. 10c and Fig. 10d. The reason is as follows: the pressure between the tool and the object together with the tool motion along the surface gives rise to small forces in the X- and Y-axes of the tool; these forces are interpreted by the CAC algorithm as "soft" obstacles, yielding small tracking errors, as can be clearly seen in Fig. 10c. On the contrary, the detected forces in the X- and Y-axes do not perturb the reference tracking of the ASMC method as long as they are below the thresholds $F_{x,\max}$ and $F_{x,\max}$ used to detect the obstacles. Obviously, the perturbation of the reference tracking for the CAC approach can be avoided if the force measurements in the X- and Y-axes are not considered, although this would deactivate the obstacle avoidance feature.

7.3. Comparison experiment with no obstacles and changes in the target position and orientation

For this experiment the force measurements in the tool X- and Y-axes are not considered for the CAC approach in order to avoid degrading its reference tracking performance and, consequently, no obstacles are introduced. The experiment consists in tracking the circular trajectory changing the position and orientation of the flat target. The video of this experiment can be played https://media.upv.es/player/?id=84cfd960-37e6-11e8-a92e-a3483ec58dec, where it can be appreciated the behavior of both control approaches. It can be seen in Fig. 11e and Fig. 11f that the reference tracking is successfully performed by both methods. It is also apparent in the video that the surface contact task is being performed properly by both approaches: the tool orientation is perpendicular to the object surface and the pressure with the tool on the surface is being kept. However, it can be seen in Fig. 11a that the forces of the CAC approach are not properly switching around the reference values in the time interval 9s–85s (35s–1m51s in the video), which corresponds to the phase when the flat object is held above the table and its orientation is being modified.



(a) CAC - Forces

(b) ASMC - Forces

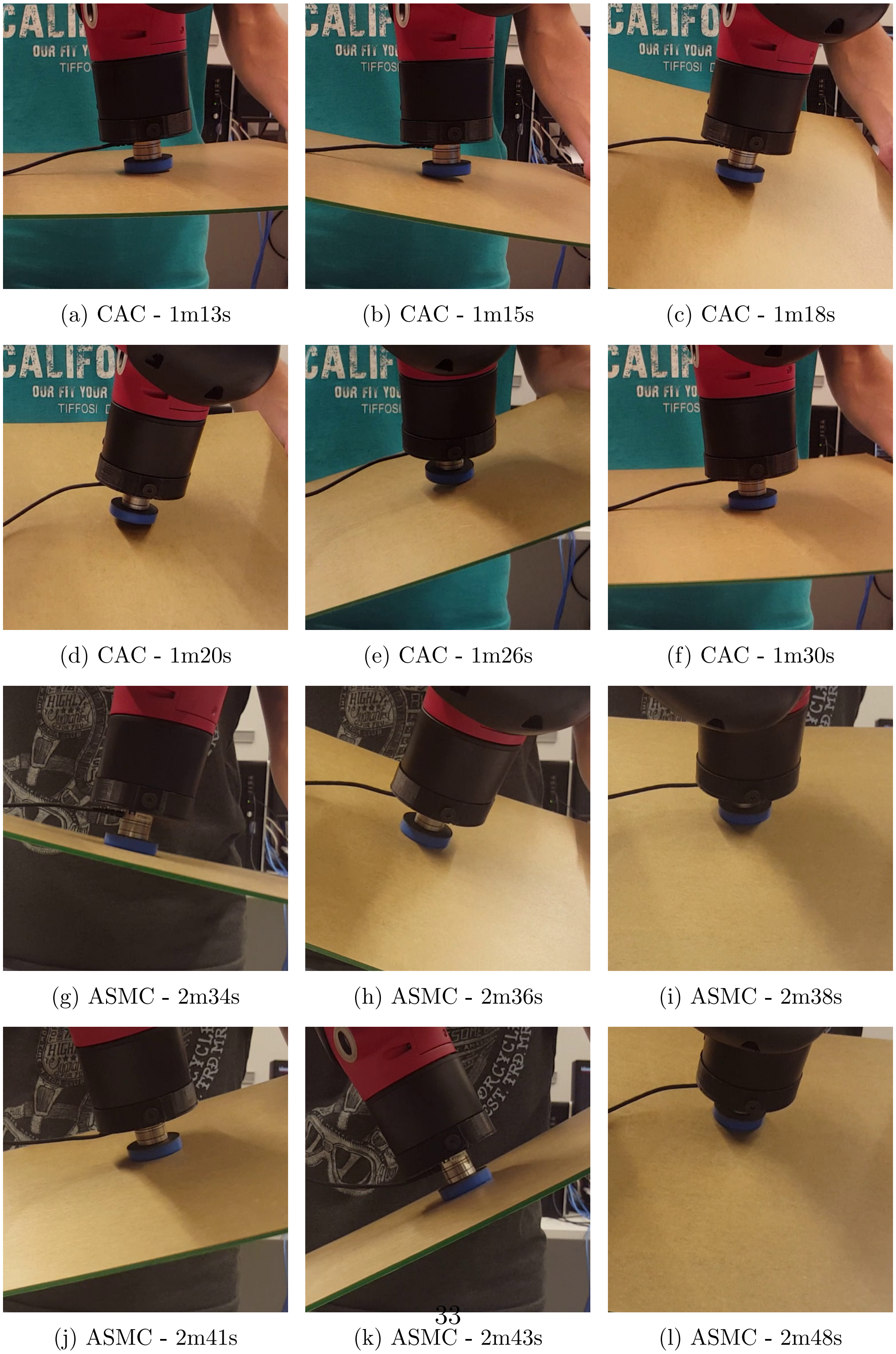
(c) CAC - 3D view of the trajectories

(d) ASMC - 3D view of the trajectories

(e) CAC - Top view of the trajectories

(f) ASMC - Top view of the trajectories

Fig. 11. CAC vs. ASMC comparison experiment with no obstacles and changes in the target position and orientation: top, forces (F_α thin-blue line, F_β thick-cyan line); middle and bottom, 3D and top views, respectively, of the trajectory followed by the tool (thin-blue line) and reference (thick-red line).



(a) CAC - 1m13s

(b) CAC - 1m15s

(c) CAC - 1m18s

(d) CAC - 1m20s

(e) CAC - 1m26s

(f) CAC - 1m30s

(g) ASMC - 2m34s

(h) ASMC - 2m36s

(i) ASMC - 2m38s

(j) ASMC - 2m41s

(k) ASMC - 2m43s

(l) ASMC - 2m48s

Fig. 12. Frames of the video of the CAC vs. ASMC comparison experiment with fast changes in the target orientation: (a) to (f) CAC and (g) to (l) proposed ASMC. The time instant is indicated for each frame.

To further analyze this issue, another experiment has been conducted using just a point as reference and changing the target orientation faster than in the previous case. The video of this experiment can be played in https://media.upv.es/player/?id=30c22800-37e1-11e8-a92e-a3483ec58dec, where it can be clearly seen that the ability of the CAC approach to quickly adapt the tool orientation to changes in the target orientation is significantly worse than that of the proposed ASMC method, see video frames in Fig. 12 that show the different behavior of both approaches. Hence, it can be concluded that, although the design parameters of the CAC approach have been empirically tuned to obtain the fastest stable response, its adaptation is significantly slower than that of the proposed ASMC.

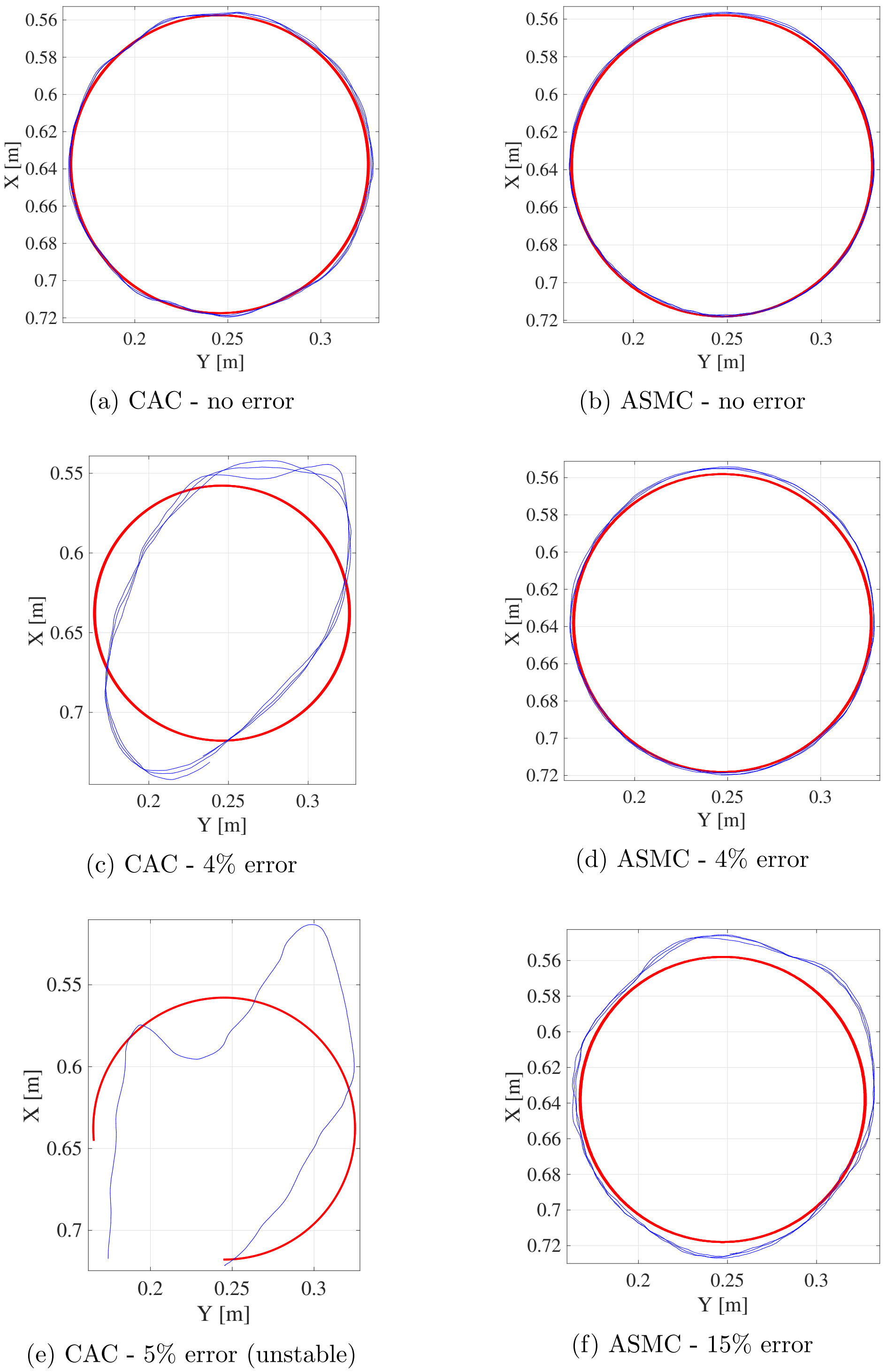
7.4. Comparison experiment with model errors

The robustness of the proposed ASMC method over the CAC approach has been experimentally analyzed by adding a signed error in the Jacobian matrix as follows:

$$\mathbf{J}_{n,e} = \mathbf{J}_n + j_e \begin{bmatrix} -1 & -1 & -1 & 1 & 1 & -1 & -1 \\ -1 & 1 & 1 & -1 & 1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & -1 & 1 \\ -1 & 1 & 1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & 1 & 1 & -1 & 1 \\ 1 & 1 & -1 & 1 & 1 & 1 & -1 \end{bmatrix} \circ |\mathbf{J}_n|, \tag{45}$$

where $\mathbf{J}_{n,e}$ is the Jacobian matrix with error, j_e is the considered percentage error and symbol $\circ$ denotes the element-wise or Hadamard product. As before, the force measurements in the tool X- and Y-axes are not considered for the CAC approach to avoid degrading its reference tracking performance.

The video of this experiment can be played in https://media.upv.es/player/?id=de453480-37f6-11e8-a92e-a3483ec58dec, where the following values have been considered for the percentage error: $\{0\%, 4\%, 5\%\}$ for the CAC approach and $\{0\%, 4\%, 15\%\}$ for the proposed ASMC. In particular, it is obtained that: the CAC approach becomes unstable for 5% of error, see Fig. 13a, Fig. 13c and Fig. 13e; while the ASMC method allows up to 15% error without significantly degrading the control task, see Fig. 13b, Fig. 13d and Fig. 13f.



(a) CAC - no error

(b) ASMC - no error

(c) CAC - 4% error

(d) ASMC - 4% error

(e) CAC - 5% error (unstable)

(f) ASMC - 15% error

Fig. 13. CAC vs. ASMC comparison experiment with model errors: top, no error; middle, 4% error; bottom left, 5% errror; bottom right, 15% error. Top views of the trajectory followed by the tool (thin-blue line) and reference (thick-red line) for each case.

7.5. Advantages and limitations of the proposed method

The main advantages of the proposed ASMC method over the CAC approach are listed as follows:

- The proposed method allows to *avoid obstacles* without degrading the performance of the tracking of the reference signals, see Fig. 10c and Fig. 10d.
- The proposed method allows *fast adaptation* to changes in the target position and orientation, see Fig. 12. This is mainly due to the high reactive nature of the SMC together with the ASG algorithm designed in this work, whereas classical continuous control methods typically have significant inertia.
- A main advantage of SMC techniques over classical continuous control approaches is *robustness*, which has been illustrated in this work by adding an error in the Jacobian matrix, see Fig. 13.
- The proposed ASMC uses *partial information* of the system model, i.e., the Lie derivatives $L_f\phi_i$ are not needed, only the Lie derivatives $\mathbf{L_g}\phi_i$ are required, which are given by the robot Jacobian and the constraint functions ϕ_i computed from the sensor measurements, see (34). In contrast, the CAC approach requires the computation of the derivative of the robot Jacobian, see (44).

The main limitation of the proposed ASMC method is the *chattering* drawback, although it is partially mitigated in this work introducing a first-order low-pass filter between the original constraint function σ_i and the modified constraint function ϕ_i, see (18) and (19). Therefore, although ϕ_i theoretically switches its value every sampling period, the original constraint function σ_i is significantly smoothed, see Fig. 5. In particular, the *maximum force peaks* obtained in the experiments for the constraint functions $\sigma_{eq,z} = F_z - F_{z,ref}$, $\sigma_{eq,\alpha} = F_\alpha$ and $\sigma_{eq,\beta} = F_\beta$ are around 10N, 400Nmm and 400Nmm, respectively, see Fig. 5, Fig. 10b and Fig. 11b.

It is also worth mentioning that classical continuous control approaches present a similar switching behavior when large correction gains are used in the system controller and non-negligible noise is introduced in the feedback loop. This behavior can be clearly seen for the CAC in Fig. 10a and Fig. 11a, where similar force peaks to those indicated above for the ASMC method can be seen, which is mainly due to the large correction gains used to obtain the fastest stable response of the CAC approach, see Section 7.1.

8. Conclusions

A hybrid position/force control approach has been developed in this work for surface contact conditioning using sliding mode ideas. In particular, conventional and non-conventional sliding mode controls have been used to satisfy equality and inequality constraints, respectively, in order to perform the surface contact conditioning. In addition, adaptive switching gain laws were considered for both conventional and non-conventional sliding mode controls. A lower priority tracking controller has also been defined to follow the desired reference trajectory on the target surface.

The classical admittance control typically used in force control tasks has been adapted for the proposed surface contact application in order to experimentally compare the performance of both control approaches. In this sense, the effectiveness and advantages of the proposed robot controller in achieving contact-driven tasks have been demonstrated by exhaustive experimentation using a redundant 7R manipulator, and a simple profiling tool used to undertake various surface contour-following exercises.

As further work, the proposed controller for surface contact tasks will be tested in a surface polishing task under realistic conditions. For this purpose, a bespoke polishing tool suitable to be mounted on the Sawyer robot end-effector is currently being developed.

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] 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.

[3] 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.

[4] F. Dimeas, V. Moulianitis, N. Aspragathos, Manipulator performance constraints in human-robot cooperation, Robotics and Computer-Integrated Manufacturing 50 (2018) 222–233.

[5] F. Dimeas, N. Aspragathos, Online stability in human-robot cooperation with admittance control, IEEE Transactions on Haptics 9 (2016) 267–278.

[6] B. Yao, Z. Zhou, L. Wang, W. Xu, Q. Liu, A. Liu, Sensorless and adaptive admittance control of industrial robot in physical human-robot interaction, Robotics and Computer-Integrated Manufacturing 51 (2018) 158–168.

[7] 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.

[8] M. Beschia, E. Villagrossi, L. Tosatti, D. Surdilovic, Sensorless model-based object-detection applied on an underactuated adaptive hand enabling an impedance behavior, Robotics and Computer-Integrated Manufacturing 46 (2017) 38–47.

[9] C. Edwards, S. Spurgeon, Sliding Mode Control: Theory and Applications, Taylor & Francis, UK, 1st edition, 1998.

[10] 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.

[11] 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.

[12] 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 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference, pp. 2966–2971.

[13] C. Kunusch, P. Puleston, M. Mayosky, Fundamentals of Sliding-Mode Control Design, Springer London, London, pp. 35–71.

[14] 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.

[15] 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.

[16] F. Plestan, Y. Shtessel, V. Bregeault, A. Poznyak, New methodologies for adaptive sliding mode control, International Journal of Control 83 (2010) 1907–1919.

[17] P. Muñoz Benavent, L. Gracia, J. Solanes, A. Esparza, J. Tornero, Sliding mode control for robust and smooth reference tracking in robot visual servoing, International Journal of Robust and Nonlinear Control 28 (2018) 1728–1756.

[18] P. Muñoz Benavent, L. Gracia, J. Solanes, A. Esparza, J. Tornero, Robust fulfillment of constraints in robot visual servoing, Control Engineering Practice 71 (2018) 79–95.

[19] B. Siciliano, L. Sciavicco, L. Villani, G. Oriolo, Robotics: Modelling, Planning and Control, Springer-Verlag, London, UK, 2009.

[20] P. Corke, Robotics, Vision and Control: Fundamental Algorithms in MATLAB, Springer-Verlag, Berlin, Germany, 2nd edition, 2017, ch. 8.1.2 and ch. 8.1.3, pp. 232-233.

[21] Y. Nakamura, H. Hanafusa, T. Yoshikawa, Task-priority based redundancy control of robot manipulators, The Int. Journal of Robotics Research 6 (1987) 3–15.

[22] B. Siciliano, J. Slotine, A general framework for managing multiple tasks in highly redundant robotic systems, in: Proceedings of the Fifth Int. Conference on Advanced Robotics (ICAR'91), Pisa, Italy, pp. 1211–1216, 1991.

[23] G. Monsees, J. Scherpen, Adaptive switching gain for a discrete-time sliding mode controller, International Journal of Control 75 (2002) 242–251.

[24] Q. Xu, Adaptive discrete-time sliding mode impedance control of a piezoelectric microgripper, IEEE Transactions on Robotics 29 (2013) 663–673.

[25] A. Liégeois, Automatic supervisory control of the configuration and behavior of multibody mechanisms, IEEE Trans. on Systems, Man and Cybernetics 7 (1977) 868–871.

[26] L. Huo, L. Baron, The joint-limits and singularity avoidance in robotic welding, Industrial Robot: An Int. Journal 35 (2008) 456–464.

[27] G. Antonelli, Stability analysis for prioritized closed-loop inverse kinematic algorithms for redundant robotic systems, IEEE Transactions on Robotics 25 (2009) 985–994.

[28] G. Golub, C. Van Loan, Matrix Computations, The Johns Hopkins University Press, Baltimore, MD, 3rd edition, 1996.