Cooperative Transport Tasks with Robots Using Adaptive Non-conventional Sliding Mode Control

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

[a]*Instituto de Diseño y Fabricación (IDF), Universitat Politècnica de València (UPV), Camino de Vera s/n, 46022 València, Spain ([*]Corresponding author e-mail: luigraca@isa.upv.es, fax: +34-963879579).*
[b]*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 aimed at handling applications using multi-task and sliding mode ideas. The proposed robot control is based on a novel adaptive non-conventional sliding mode control used to robustly satisfy a set of inequality constraints defined to accomplish the cooperative transport task. In particular, these constraints are used to guarantee the reference parameters imposed by the task (e.g., keeping the load at a desired orientation) and to guide the robot using the human operator's forces detected by a force sensor located at the robot tool. Another feature of the proposal is the multi-layered nature of the strategy, where a set of four tasks are defined with different priorities. The effectiveness of the proposed adaptive non-conventional sliding mode control is illustrated by simulation results. Furthermore, the applicability and feasibility of the proposed robot control for transport tasks are substantiated by experimental results using a redundant 7R manipulator.

Keywords: cooperative task, robot system, force control, sliding mode control

1. Introduction

Recent advances in technology and robotics are revolutionizing modern society. Robots are becoming more and more present in the form of unmanned aircraft systems, commonly known as drones, driver-less cars, robot-

assisted surgery and rehabilitation systems, robotic prosthetics and exoskeletons, service robots for personal and domestic use, artificial assistants and smart machines, among others.

Possibly, the manufacturing industry in general has been the most benefited by advances in the fields of robotics, control and sensing, bringing in improvements to production processes as well as worker's ergonomics and job quality. Contrary to the old tendency of developing autonomous systems to replace humans by robotic devices, currently the research is more focused on developing robots to work alongside humans and assist them. The reason is that the combination of human cognitive and sensorimotor skills with the technical capabilities of a robot have proven able to solve, facilitate, improve and/or speed up a large variety of complex tasks that neither humans nor robots could successfully afford to do in solitary [23, 22, 2, 38, 25].

A case in point in the manufacturing sector is the manipulation and safe transportation of precarious loads, most notably heavy objects such as car engines, or fragile items such as glass, liquid containers, hazardous materials etc. Moreover, in many cases loads have to be transported and awkwardly deposited in difficult-to-access areas that make it difficult or ergonomically challenging for an operator to keep them in a predetermined position and/or orientation , whilst simultaneously pursuing a higher level assembling or handling assignment. Yet the automatic realization of these type of assignments by a robot is usually discarded due to the limited flexibility afforded by a robot in adapting to changes in the production workspace. Hence, the combination of skillful guidance by the human operator on the one hand, and the sensorimotor stability and strength of the robot on the other can lead to industrially feasible human-robot collaborative solutions for handling applications.

Generally, in this kind of applications guidance for the motion of the manipulator is obtained via a wrist-mounted force sensor which evaluates the forces exerted by the human operators. The most commonly used method to convert these measurements into kinematic instructions to the robot is through compliance control, which establishes a direct relationship between the measured forces and the changes in the robot position [28, 15]. Yet other variants and methods can be found in the literature. For instance, authors in [11] presented a method to determine the compliance controller parameters of the physical model using a particle swarm optimization algorithm for a spinal surgery application. In [19] a force tracking method under the impedance control framework was extended to also account for uncertain hu-

man limb dynamics. An adaptive controller was developed to deal with point-to-point movements, whereas learning and neural network controls were included to generate periodic and arbitrary continuous trajectories respectively. A hierarchical control system was presented in [16] for the co-manipulation task of folding sheets like fabrics/cloths. The system was based on force and RGB-D feedback at two distinctive control levels. At a higher level, the perception of the human's intention was used to decide on the robot's action, whereas at a lower level the robot reacted to force and RGB-D feedback in following the guidance from the human. In [41] a decision-and-control architecture was proposed for hand-arm systems with "soft robotics" capabilities via dedicated human-machine interfaces. The robot was controlled through a multi-priority Cartesian impedance controller, and the behavior extended with collision detection and reflex reactions. The problem of human-directed position/force control of a robot end-effector interacting with an environment given unknown geometry and stiffness was addressed in [20]. In free space (non-contact) motion, the input was interpreted as a linear velocity command. When contact occurred, a generalized damper-type of impedance control was used for the regulation of force in the constrained direction, while the input from the user would adjust the contact force set point. In [13] a mathematical relation between the velocity of the human-robot interaction point and the force applied by the human operator was established using impedance control for handling tasks, where an adjustable force threshold was used to enable the operator to keep authority over the robot motion. An optimal impedance adaptation was investigated in [42] for interaction control in constrained motions, which lead to an optimal realization of trajectory tracking and force regulation.

A relevant aspect to be taken into account for human-robot collaboration that directly affects the maneuverability and the human's ergonomics is the compensation of the effort demanded by the human throughout the guidance task. In other words, the manipulator should adapt itself according to the force exerted by the human, causing a greater sensation of freedom of movement and avoiding slow motions. None of the above works deals with this aspect and, to the best of the authors' knowledge, this is the first work that proposes a feasible solution based on an adaptive controller [50, 8, 29, 6, 21, 26].

Some approaches for human-robot collaborative applications driven by force-control strategies are based on sliding mode control (SMC) theory given its inherent robustness and low computational cost characteristics [9]. For

instance, in [49] a non-singular terminal SMC was developed to ensure trajectory tracking precision for the case of a lower limb rehabilitation parallel robot. The device would adjust the gait trajectory online according to the indications from a human-machine interaction force set-up. In [12] a human-robot interaction controller was introduced for a lower extremity exoskeleton whose aim was to improve tracking performance with the development of a fuzzy SMC that considered system uncertainties. In this way, the controller was able to drive the exoskeleton to shadow the wearer in the presence of weaker interactive driving forces. In [14] a proxy-based SMC was proposed to obtain effective tracking during normal operations for flexible joint manipulators working close to humans, whilst retaining the ability to recover from positional errors in a smooth and damped manner. In [48] a robust SMC was proposed that relied on basic information from the human subject (weight, height, age and gender) to handle model uncertainties due to biomechanical variation of patients using an upper limb rehabilitation robot. An SMC consisting of a proportional-integral-derivative sliding surface and a fuzzy hitting control law was developed in [44] to guarantee robust tracking performance and reduce the chattering effect for a class of robot-assisted therapeutic exoskeleton. A fuzzy SMC presented in [24] considered a nonlinear model for trajectory tracking of micro robots in the human vasculature system. It is worth mentioning that SMC has been recently used in the field of robot force control not only to improve controller robustness but also to improve force estimation by means of a sliding perturbation observer (SPO) in order to avoid the use of expensive force sensors. For examples of this approach, see [31, 30].

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, which is a well known issue to be solved in SMC techniques [32, 17, 39]. 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. In the specific problem treated in this paper, this issue is present due to changes on the forces exerted by the human operator in order to guide the robot to perform the collaborative transport task.

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 [50] an ASMC was developed using an integral/exponential adaptation law with boundary-layer in order to re-

duce the switching gain overestimation while simultaneously speeding up the system response to the uncertainties. In [33] 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 [37] a high-order ASMC was proposed based on the concepts of integral sliding mode and real high-order sliding mode detector. For more SMCs with ASG solutions, the reader is referred to [8, 29, 6, 1, 21, 4, 26, 5, 18, 43, 45], among others.

PARCEL

Fig. 1. Human-robot cooperation to transport a parcel.

The approach proposed in this work also exploits SMC so that human operator and robot can cooperatively undertake the transportation of objects, such as the one depicted in Fig. 1, with the aid of force feedback sensing. However, the SMC in this work offers several distinctive features that sets it apart from other works in the literature. In particular, the main contributions and features of the proposed approach are as follows:

- The introduction of inequality constraints within the SMC framework is a key novelty of the proposed method.

- The control strategy relies on a novel adaptive non-conventional SMC regulator to fulfill the inequality constraints.

- The strategy constrains a subset of the robot pose coordinates by prudent reference values imposed by the handling operation, e.g., keeping the load at a desired orientation to prevent spill-offs or to reduce undue stresses that may compromise fragile items as in glass transportation.

- Given the multi-layered nature of the proposed strategy, remaining degrees of freedom in the robot pose are thus left to be guided by the human operator in a lower priority loop using a force sensor located at the robot tool to detect the operator's forces to accomplish the task safely and concurrently. Note that the proposed ASG algorithm is a key advantage in order to adapt the robot control to the changes on the forces exerted by the human operator to guide the robot.

- A low-priority task for the specific case of redundant robots (as is the case with the experiments shown in this work) is also suggested to keep the robot close to the home configuration for increased safety.

Although the motivation of this work is the cooperative transportation of an object, the proposed hybrid position/force robot control can also be used to tackle other industrial applications where the human operator guides the robot under the constraint of some reference values. For instance, in a drilling application, the robot can be guided by the operator to an *a priori unknown* position in order to operate a drill, whilst the tool orientation is constrained to remain perpendicular to a (pre-set or sensed on-the-fly) drilling surface.

The paper is organized as follows. Next section introduces some preliminaries, while Section 3 develops the adaptive non-conventional SMC used in this work. The proposed hybrid control approach is presented in Section 4, while some important remarks about the method are given in Section 5. A simulation is presented in Section 6 to show the performance of the proposed adaptive non-conventional SMC. The implementation of the proposed robot controller is detailed in Section 7. The feasibility of the proposed approach is substantiated by experimental results in Section 8 using a redundant 7R manipulator: the Rethink Sawyer collaborative robot. Finally, some conclusions are given.

2. Preliminaries

2.1. Kinematics

Following the standard notation [7], the robot *pose* $\mathbf{p}$ depends on the robot *configuration* $\mathbf{q}$ as follows:

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

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

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

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

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

2.2. Robot control

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

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

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

2.3. Task-priority based redundancy resolution

It is useful to consider the task-priority strategy [7] 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 [27]. 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 $\ddot{\mathbf{q}}_c$

7

(i.e., the commanded joint acceleration vector) in order to fulfill the following acceleration equality constraints:

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

where matrix $\mathbf{A}_i$ and vector $\mathbf{b}_i$ of the ith task are assumed known and index i represents the priority order: $i = 1$ for highest priority 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 [36]:

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

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

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

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. Note that the mentioned pseudoinverse may be computed via the singular value decomposition (SVD) method [10] and using a tolerance to set to zero the very small singular values in order to avoid extremely large values for the commanded accelerations.

3. Adaptive One-side Sliding Mode Control

The proposed method to accomplish the cooperative transport task using a robotic system is based on satisfying a set of inequality constraints, defined in Section 4. The fulfillment of the mentioned inequality constraints will be achieved using SMC theory to benefit from the typical advantages of this type of controllers, such as robustness and low computational cost. However, conventional SMC can only be used to satisfy equality constraints. Hence, this section presents a non-conventional SMC algorithm, coined as one-side SMC, developed to satisfy inequality constraints.

3.1. One-side vs. conventional sliding mode control

The proposed one-side SMC is different to the conventional SMC, see Fig. 2 for a graphical two dimensional example to illustrate both approaches.

For conventional SMC (see Fig. 2-left) the state space of the system is divided into two regions, $\mathbf{A}$ and $\mathbf{B}$, separated by the sliding surface. The
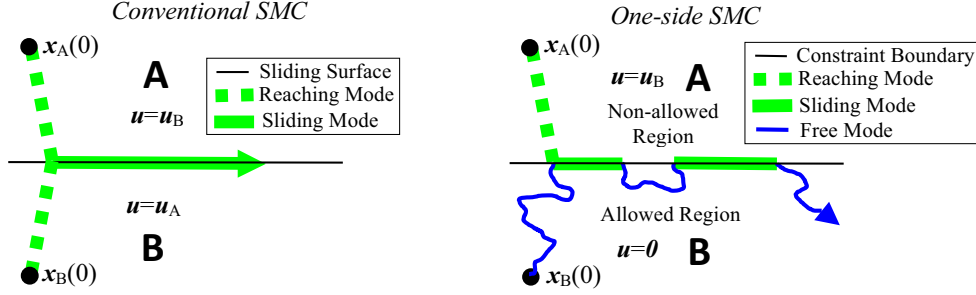
Fig. 2. Graphical comparison between conventional SMC (left) and one-side SMC (right).

value of the control action $\mathbf{u}$ when the system state is in region $\mathbf{A}$ is such that it "pushes" the system into region $\mathbf{B}$, namely $\mathbf{u}_B$. Analogously, when the system state is in region $\mathbf{B}$, the value of control action is such that it pushes the system into region $\mathbf{A}$, namely $\mathbf{u}_A$. Hence, regardless of whether the system starts in region $\mathbf{A}$ - $\mathbf{x}_A(0)$, or $\mathbf{B}$ - $\mathbf{x}_B(0)$, it evolves to the sliding surface in what is referred as *reaching mode* [9]. Once the system has reached the sliding surface, the system is kept on it by a control action $\mathbf{u}$ that switches between $\mathbf{u}_A$ and $\mathbf{u}_B$ at a theoretically infinite frequency, which is known as *sliding mode* (SM) [9].

The one-side SMC proposed in this work is graphically represented in Fig. 2-right and is used to satisfy inequality constraints. In this case, the state space of the system is divided into the non-allowed region $\mathbf{A}$ and the allowed region $\mathbf{B}$, which are separated by the inequality constraint boundary. Similarly to conventional SMC, when the system sate is in the non-allowed region $\mathbf{A}$ the control action $\mathbf{u} = \mathbf{u}_B$ pushes the system into the allowed region $\mathbf{B}$. But in contrast to conventional SMC, when the system state is in the allowed region $\mathbf{B}$, no control action is applied, i.e., $\mathbf{u} = \mathbf{0}$. Hence, if the system starts in region $\mathbf{A}$ - $\mathbf{x}_A(0)$, it evolves in reaching mode to the boundary of the constraint. Nevertheless, when the system starts in the allowed region $\mathbf{B}$ - $\mathbf{x}_B(0)$, the system state can "freely" evolve according to some other criterion, e.g., a control law for reference tracking. Therefore, only when the state trajectory tries to leave the allowed region, the one-side SMC will make $\mathbf{u}$ switch between $\mathbf{0}$ and $\mathbf{u}_B$ at a theoretically infinite frequency, which can be seen as an ideal SM behavior [9].

3.2. Remarks about the proposed one-side sliding mode control

The main distinctive features of the proposed one-side SMC over conventional SMC are as follows. Conventional SMC is used to fulfill *equality* constraints, i.e., to keep the system state always on the sliding surface (see Fig. 2-left), hence it is *always active*. In contrast, the proposed one-side SMC is used to fulfill *inequality* constraints and *only becomes active* when the state trajectory tries to leave the allowed region, applying the required SM control action to keep the system state on the boundary of the allowed region (see Fig. 2-right), i.e., giving rise to the SM behavior. Therefore, in contrast to conventional SMC, the proposed method does not seek for SM, but it arises when the system state is at the boundary of the allowed region and about to leave it, as illustrated in Section 6.

The following remarks can be made for the proposed one-side SMC:

- During the SM phase, the sliding surface corresponds to the boundary of the allowed region, as usual in SMC strategies. The manifold remains the main characteristic in the design of the SMC controller. For instance, this boundary can represent the maximum allowed error (see Section 4.3), the maximum allowed force (see Section 4.4), etc.

- During the SM phase, the one-side SMC has the typical advantages of SMC strategies, such as robustness and low computational cost, see Section 5.4.

- When the one-side SMC is not active, it uses no degrees of freedom of the system and its state can evolve according to other lower priority tasks, see Section 2.3.

Note that the proposed method has been coined one-side SMC because it behaves as conventional SMC only from one side of the sliding surface (constraint boundary), i.e., only when the system state is in the non-allowed region. In fact, conventional SMC (equality constraint) can be obtained combining two one-side SMCs (inequality constraints), where the first one considers **A** as the non-allowed region, while the second one considers **B** as the non-allowed region, see Fig. 2-right. Thus, in this case only the boundary of both inequality constraints is allowed, i.e., the sliding surface.

It is also worth noting that in conventional SMC the sliding surface (i.e., the equality constraint) is typically designed to assign the desired dynamics, to achieve a reference state, etc. In contrast, the proposed one-side SMC

is intended to "confine" the system state within the allowed region given by the inequality constraints and, in doing so, the system state can thus "freely" evolve within the boundaries of the allowed region, e.g., according to other lower priority tasks as discussed in Section 2.3.

3.3. Control law for the one-side sliding mode control

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

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

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

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 inequality *constraints $\phi_i(\mathbf{x}) \leq 0$, $i = 1, \ldots, N$, where $\phi_i(\mathbf{x})$ is the ith inequality constraint function. Thus, the region Φ of the state space compatible with the constraints on state $\mathbf{x}$ is given by:*

$$\Phi = \left\{\mathbf{x} \,\middle|\, \phi_i(\mathbf{x}) \leq 0\right\}, \tag{9}$$

with $i = 1, \ldots, N$.

Then, assuming that the constraint functions ϕ_i are differentiable around the boundary of the allowed region, which is given by $\phi_i = 0$, 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:

$$\text{v2m}\left(\text{pos}\left(\boldsymbol{\phi}\right)\right) \mathbf{L_g}\boldsymbol{\phi}\, \mathbf{u} = -\mathbf{W}\,\text{pos}\left(\boldsymbol{\phi}\right)\, u^{+} \tag{10}$$

$$u^{+} > \sum_{i=1}^{n_a} \left(\max(L_f\phi_i, 0)\right)/\text{diag}_{\min}(\mathbf{W}), \tag{11}$$

where function $\text{v2m}(\cdot)$ converts a vector into a diagonal matrix, function

pos($\cdot$) *represents the* positive function[1] *(i.e.,* pos(x) *is equal to* 0 *if* $x < 0$ *and equal to* 1 *if* $x > 0$*), matrix* $\mathbf{L_g}\boldsymbol{\phi}$ *contains the row vectors* $\mathbf{L_g}\phi_i$ *of all inequality constraints, the scalar* $L_f\phi_i$ *and the row vector* $\mathbf{L_g}\phi_i$ *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,* $\boldsymbol{\phi}$ *is a column vector with all the inequality constraint functions* ϕ_i, *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 inequality constraints,* n_a *is the number of active inequality constraints, i.e., those with* $\phi_i \geq 0$, *and function* $\text{diag}_{\min}(\cdot)$ *computes the minimum value of the diagonal elements of a matrix.*

Therefore, the required control action $\mathbf{u}$ can be easily computed from the control equality (10) by applying the Moore-Penrose pseudoinverse [10], see Section 2.3.

Note that the expression v2m(pos($\boldsymbol{\phi}$)) is used in (10) to obtain the zero value on the left-side of the control equality for the non-active inequality constraints, i.e., those with $\phi_i < 0$. Thus, no degrees of freedom of the system are used by these constraints when computing the control action $\mathbf{u}$ via the Moore-Penrose pseudoinverse, as mentioned above.

The proof of Theorem 1 is analogous to Proof 2.1 in [40] for conventional SMC. The main difference is that the commutation function sign($\cdot$) typically used in conventional SMC has to be replaced with the commutation function pos($\cdot$) used in the proposed one-side SMC. Obviously, the condition obtained for the switching gain is different and, hence, the proof is given below for completeness.

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

---

[1]Note that pos($\cdot$) is a *commutation function* that plays the same role as the sign function sign($\cdot$) typically used in conventional SMC [34]. Hence, while by definition the values of pos(x) and sign(x) fall within the intervals $[0, 1]$ and $[-1, 1]$ respectively, at the commutation point $x = 0$ they are actually indeterminate. Although it is unlikely that x equals exactly zero, to be conservative pos(0) = 1 will be used in order to guarantee the fulfillment of the inequality constraints.

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

The column vector $\dot{\phi}$ composed of the constraint function derivatives $\dot{\phi}_i$ is given by

$$\dot{\phi} = \frac{\partial \phi^{\mathrm{T}}}{\partial \mathbf{x}} \mathbf{f}(\mathbf{x}, \mathbf{d}) + \frac{\partial \phi^{\mathrm{T}}}{\partial \mathbf{x}} \mathbf{g}(\mathbf{x}) \mathbf{u} = L_f \phi + \mathbf{L_g} \phi \mathbf{u}. \tag{12}$$

Premultiplying (12) by $\mathrm{v2m}\,(\mathrm{pos}\,(\phi))$ and substituting (10) yields:

$$\mathrm{v2m}(\mathbf{z})\dot{\phi} = \mathrm{v2m}(\mathbf{z})L_f\phi - \mathbf{W}\,\mathbf{z}\,u^+, \tag{13}$$

where $\mathbf{z}$ is a column vector with the ith-component $z_i = 1$ if $\phi_i > 0$ and $z_i = 0$ if $\phi_i < 0$.

Let $V = \mathbf{z}^{\mathrm{T}}\,\mathrm{v2m}(\mathbf{z})\,\phi$ be a Lyapunov function candidate. Vector ϕ^{n_a} can be generically partitioned into two subvectors $\phi^{n_a} = [\phi^{b\,\mathrm{T}} \quad \phi^{n_a-b\,\mathrm{T}}]^{\mathrm{T}}$, where SM occurs in the manifold given by $\phi^b = \mathbf{0}$, whereas the components of vector ϕ^{n_a-b} are greater than zero. Obviously, one of these two subvectors may be empty at a certain time: typically ϕ^b is empty at the beginning, while ϕ^{n_a-b} is empty at the end, i.e., there are no constraints in SM at the beginning, while all of them are in SM at the end. Since vectors $\mathbf{z}^{n_a-b} = \mathbf{1}$

and $\mathbf{z}^{N-n_a} = \mathbf{0}$ are constant, the time derivative of V results in:

$$
\begin{aligned}
\dot{V} &= \frac{d\left(\mathbf{z}^{\mathrm{T}}\, \text{v2m}(\mathbf{z})\boldsymbol{\phi}\right)}{dt} \\[2mm]
&= \frac{d\left(\mathbf{z}^{\mathrm{T}}\, \text{v2m}(\mathbf{z})\right)}{dt}\boldsymbol{\phi} + \mathbf{z}^{\mathrm{T}}\, \text{v2m}(\mathbf{z})\, \dot{\boldsymbol{\phi}} \\[2mm]
&= \frac{d\left(\begin{bmatrix}\mathbf{z}^b\\1\\0\end{bmatrix}^{\mathrm{T}} \text{v2m}\left(\begin{bmatrix}\mathbf{z}^b\\1\\0\end{bmatrix}\right)\right)}{dt}\begin{bmatrix}\mathbf{0}\\\boldsymbol{\phi}^{n_a-b}\\\boldsymbol{\phi}^{N-n_a}\end{bmatrix} + \mathbf{z}^{\mathrm{T}}\, \text{v2m}(\mathbf{z})\, \dot{\boldsymbol{\phi}} \\[2mm]
&= \frac{d\left(\begin{bmatrix}\mathbf{z}^b\\1\\0\end{bmatrix}^{\mathrm{T}} \begin{bmatrix}\text{v2m}(\mathbf{z}^b) & \mathbf{O} & \mathbf{O}\\ \mathbf{O} & \mathbf{I} & \mathbf{O}\\ \mathbf{O} & \mathbf{O} & \mathbf{O}\end{bmatrix}\right)}{dt}\begin{bmatrix}\mathbf{0}\\\boldsymbol{\phi}^{n_a-b}\\\boldsymbol{\phi}^{N-n_a}\end{bmatrix} + \mathbf{z}^{\mathrm{T}}\, \text{v2m}(\mathbf{z})\, \dot{\boldsymbol{\phi}} \\[2mm]
&= \frac{d\left(\begin{bmatrix}(\mathbf{z}^b)^{\mathrm{T}}\text{v2m}(\mathbf{z}^b) & \mathbf{1}^{\mathrm{T}} & \mathbf{0}^{\mathrm{T}}\end{bmatrix}\right)}{dt}\begin{bmatrix}\mathbf{0}\\\boldsymbol{\phi}^{n_a-b}\\\boldsymbol{\phi}^{N-n_a}\end{bmatrix} + \mathbf{z}^{\mathrm{T}}\, \text{v2m}(\mathbf{z})\, \dot{\boldsymbol{\phi}} \\[2mm]
&= \begin{bmatrix}\dfrac{d\left((\mathbf{z}^b)^{\mathrm{T}}\text{v2m}(\mathbf{z}^b)\right)}{dt} & \mathbf{0}^{\mathrm{T}} & \mathbf{0}^{\mathrm{T}}\end{bmatrix}\begin{bmatrix}\mathbf{0}\\\boldsymbol{\phi}^{n_a-b}\\\boldsymbol{\phi}^{N-n_a}\end{bmatrix} + \mathbf{z}^{\mathrm{T}}\, \text{v2m}(\mathbf{z})\, \dot{\boldsymbol{\phi}} \\[2mm]
&= \mathbf{z}^{\mathrm{T}}\, \text{v2m}(\mathbf{z})\, \dot{\boldsymbol{\phi}},
\end{aligned}
\tag{14}
$$

where $\mathbf{1}$ and $\mathbf{0}$ are the column vectors of suitable size with all their elements equal to 1 and 0, respectively, and $\mathbf{I}$ and $\mathbf{O}$ are the identity and zero matrices of suitable size, respectively.

Substituting (13) in (14) yields:

$$
\dot{V} = \mathbf{z}^{\mathrm{T}}\, \text{v2m}(\mathbf{z})\, L_f \boldsymbol{\phi} - \mathbf{z}^{\mathrm{T}}\, \mathbf{W}\, \mathbf{z}\, u^+.
\tag{15}
$$

Since $\mathbf{z}^{N-n_a} = \mathbf{0}$ and the components of vector $\mathbf{z}^{n_a}$ range from 0 to 1, the *upper bound* of the first term in (15) is given by $z_i^{n_a} = 1$ when $L_f \phi_i^{n_a} > 0$

and $z_i^{n_a} = 0$ when $L_f \phi_i^{n_a} < 0$, that is:

$$\mathbf{z}^{\mathrm{T}} \, \mathrm{v2m}(\mathbf{z}) \, L_f \boldsymbol{\phi} \leq \sum_{i=1}^{n_a} \left(\max(L_f \phi_i, 0) \right). \tag{16}$$

Since matrix $\mathbf{W}$ is positive definite, the second term in (15) is negative and its upper bound is given by:

$$- \mathbf{z}^{\mathrm{T}} \, \mathbf{W} \, \mathbf{z} \, u^+ \leq -\mathrm{diag}_{\min}(\mathbf{W}) \, \|\mathbf{z}\|_2^2 \, u^+$$
$$\text{where } \|\mathbf{z}\|_2 \geq 1 \; \forall \, \boldsymbol{\phi} > \mathbf{0}, \tag{17}$$

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

From (16) and (17), the upper bound of the time derivative of the Lyapunov function V results in:

$$\dot{V} \leq \sum_{i=1}^{n_a} \left(\max(L_f \phi_i, 0) \right) - \mathrm{diag}_{\min}(\mathbf{W}) \, u^+. \tag{18}$$

Therefore, if u^+ fulfills (11) the Lyapunov function decays at a finite rate, it vanishes and collective SM in the intersection of the n_a active inequality constraints occurs after a finite time interval. $\qquad \square$

3.4. Assumption about the system initialization

The design procedure of a SMC includes two major steps encompassing the two main phases of SMC [47]: reaching phase (thick dashed line in Fig. 2, left), where the system state is driven from any initial state to reach the sliding surface or switching manifold; and SM phase (thick continuous line in Fig. 2, left), where the system is induced into the sliding motion on the switching manifold.

The one-side SMC proposed in this work is focused on the SM phase (thick continuous line in Fig. 2, right), i.e., it is used *to keep* the system within the limits of the allowed region. Therefore, in this work it is implicitly assumed that the *system starts in the allowed region* and, hence, the initial reaching phase is obviated.

If this assumption is not fulfilled, it is convenient to use a proper control method for the initial reaching phase in order to guarantee the entry of

the system state in the allowed region without overshoot[2]. Several control methods have been developed in the SMC literature to improve the reaching phase, see [46, 3] among others. However, this is out of the scope of this research.

3.5. Modified constraints

Approaching the boundary of the inequality 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 boundary. Therefore, the actual constraints σ_i will be modified to include the speed of movement as follows:

$$\phi_i = \sigma_i + K_i \dot{\sigma}_i \leq 0, \tag{19}$$

where K_i is a free design parameter that determines the rate of approach to the boundary of the original inequality constraint.

3.6. Chattering

Discrete-time implementations of the proposed one-side SMC makes the system leave the ideal SM and oscillate with finite frequency and amplitude inside a band around $\boldsymbol{\phi} = \mathbf{0}$, which is called *chattering* [9]. The upper bound for the chattering band $\triangle\boldsymbol{\phi}$ of the proposal can be obtained using the Euler-integration of the discontinuous control action given by Eq. (10), that is:

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

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

3.7. Adaptive switching gain

The selection of the switching gain u^+ is a common issue in SMC applications. A number of options for this purpose are discussed below.

---

[2]Note that the SMC used in the SM phase may result "too aggressive" for the initial reaching phase due to a choice of high gains, giving rise to large accelerations and velocities that cannot be promptly reduced once the allowed region has been reached.

Firstly, a *big number* could be chosen for u^+ in order to ensure that it is greater than the lower bound given by (11). However, such big numbers may induce *unnecessary control effort and chattering* amplitude, see (20).

Therefore, in order to avoid these drawbacks, a second option consists in *estimating* the mentioned lower bound and choosing a value slightly larger than the estimated one. However, since this lower bound depends on the system state, as usual in SMC applications, it may be difficult to estimate it a priori. Thus, many practical applications use a third option that consists in running the system application (either in simulation or experimentally) in order to *empirically tune* a proper value for u^+ so that the SM behavior is fulfilled at all times and the control effort and chattering amplitude are minimized.

Finally, taking into account that the switching gain u^+ can be varying in time, a fourth option consists in using an *adaptive switching gain $u^+(t)$* (ASG) on the basis that different parts of the system trajectory may require very different SM control actions. That is, the lower bound for u^+ may drastically change from one part of the trajectory to another, e.g., it is typically larger when the system trajectory executes abrupt maneuvers. The main advantage of this adaptive approach is that the control effort and chattering amplitude are minimized online according to the current part of the trajectory.

Many ASG laws can be found in the literature for conventional SMC, see [50, 8, 29, 6, 21, 26, 33, 37, 18, 43] among others. In the same spirit of the ASG discrete algorithm presented in [26, 45], this work considers the following ASG law for the proposed one-side SMC:

$$u^+(k) = u^+(k-1) + T_s \, \mu \, f_{ASG}(\phi(k), \phi(k-1)), \qquad (21)$$

where $u^+(k)$ and $u^+(k-1)$ are the values of the switching gain for the current and the previous time steps, respectively, $\phi(k)$ and $\phi(k-1)$ are values of the inequality constraint function for the current and the previous time steps, respectively, μ is a positive configuration parameter that determines the speed of the adaptation 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., $\phi(k-1) > 0$) the switching gain is decreased or increased 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., $\phi(k) < 0$) or not (i.e., $\phi(k) > 0$)

17

at the next time step. 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 (21) converges to a bounded region. The proof can be obtained straightforward considering the extension described in [26] 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 constraint boundary in just one time step. In theses cases, the commutation condition (21) 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 constraint boundary has been successfully crossed or not in order to decrease or increase the switching gain.

4. Proposed approach

4.1. Overview of the method

The objective of this work is to obtain a hybrid position/force robot control so that the robot and the human operator cooperatively manipulate and transport loads/objects. In particular, some coordinates of the robot pose are controlled to achieve the reference values imposed by the transport task, e.g., to keep a reference orientation, to keep a reference vertical position, etc. Meanwhile, the remaining coordinates of the robot pose are guided by the human operator using a *force sensor* located at the robot tool to detect the operator's forces.

Fig. 3 shows the block diagram of the proposed method. Four tasks with different priority levels are considered. Level 1 (high-priority task) is developed using the SMC theories described in Section 3 in order to fulfill a set of inequality constraints to ensure the required transport conditions for the object. Level 2 (medium-high-priority task) is also developed using the SMC described in Section 3 in order to fulfill the inequality constraints required for the robot to track the human operator's forces using the force sensor located at the robot tool. Note that the forces exerted by the human operator are detected by the force sensor either if they are applied to a handle located at the end-effector or directly to the transported object, labeled in
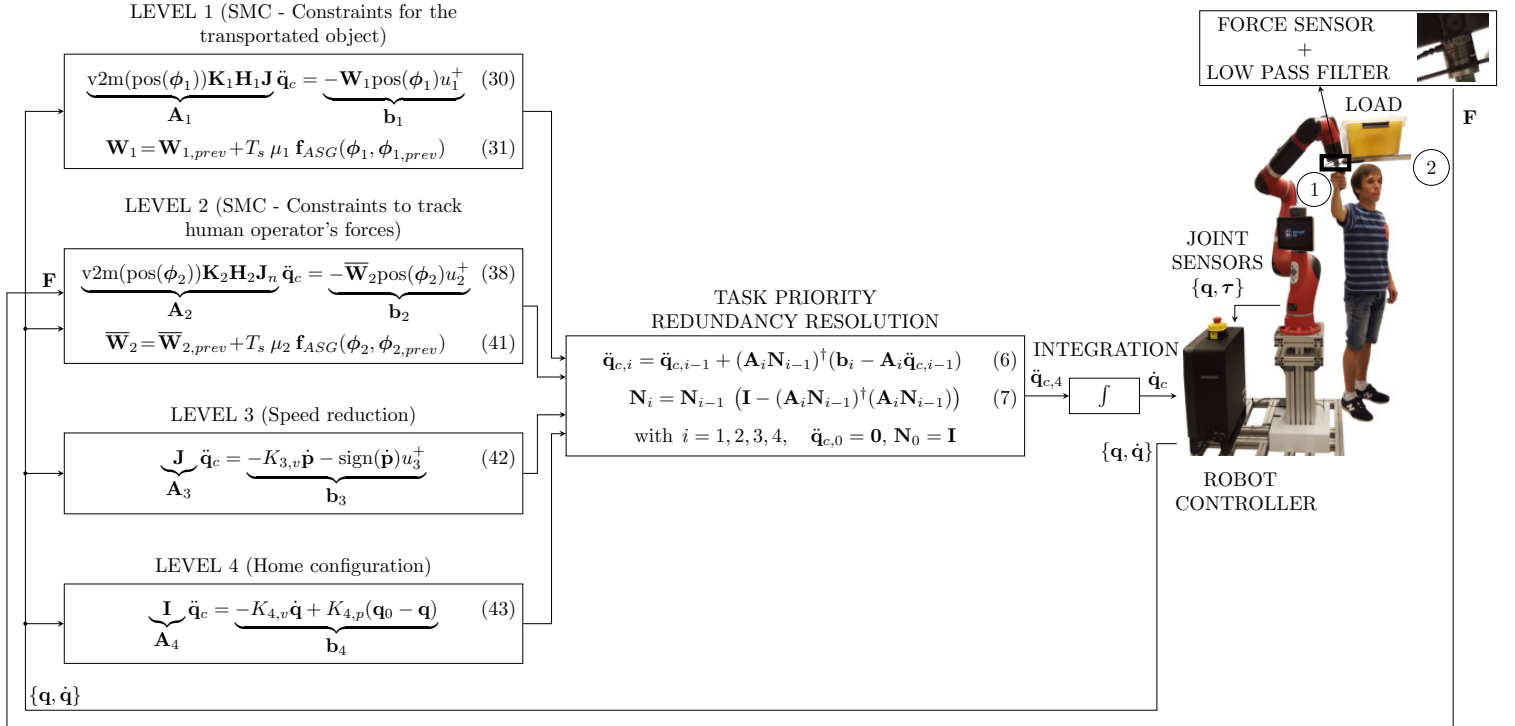
18

LEVEL 1 (SMC - Constraints for the transported object)

$$\underbrace{\text{v2m}(\text{pos}(\boldsymbol{\phi}_1))\mathbf{K}_1\mathbf{H}_1\mathbf{J}}_{\mathbf{A}_1}\ddot{\mathbf{q}}_c = \underbrace{-\mathbf{W}_1\text{pos}(\boldsymbol{\phi}_1)u_1^+}_{\mathbf{b}_1} \quad (30)$$

$$\mathbf{W}_1 = \mathbf{W}_{1,prev} + T_s\,\mu_1\,\mathbf{f}_{ASG}(\boldsymbol{\phi}_1,\boldsymbol{\phi}_{1,prev}) \quad (31)$$

LEVEL 2 (SMC - Constraints to track human operator's forces)

$$\underbrace{\text{v2m}(\text{pos}(\boldsymbol{\phi}_2))\mathbf{K}_2\mathbf{H}_2\mathbf{J}_n}_{\mathbf{A}_2}\ddot{\mathbf{q}}_c = \underbrace{-\overline{\mathbf{W}}_2\text{pos}(\boldsymbol{\phi}_2)u_2^+}_{\mathbf{b}_2} \quad (38)$$

$$\overline{\mathbf{W}}_2 = \overline{\mathbf{W}}_{2,prev} + T_s\,\mu_2\,\mathbf{f}_{ASG}(\boldsymbol{\phi}_2,\boldsymbol{\phi}_{2,prev}) \quad (41)$$

LEVEL 3 (Speed reduction)

$$\underbrace{\mathbf{J}}_{\mathbf{A}_3}\ddot{\mathbf{q}}_c = \underbrace{-K_{3,v}\dot{\mathbf{p}} - \text{sign}(\dot{\mathbf{p}})u_3^+}_{\mathbf{b}_3} \quad (42)$$

LEVEL 4 (Home configuration)

$$\underbrace{\mathbf{I}}_{\mathbf{A}_4}\ddot{\mathbf{q}}_c = \underbrace{-K_{4,v}\dot{\mathbf{q}} + K_{4,p}(\mathbf{q}_0 - \mathbf{q})}_{\mathbf{b}_4} \quad (43)$$

$\{\mathbf{q},\dot{\mathbf{q}}\}$

TASK PRIORITY
REDUNDANCY RESOLUTION

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

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

with $i = 1,2,3,4,\quad \ddot{\mathbf{q}}_{c,0} = \mathbf{0},\ \mathbf{N}_0 = \mathbf{I}$

INTEGRATION

$\ddot{\mathbf{q}}_{c,4}$ $\int$ $\dot{\mathbf{q}}_c$

$\{\mathbf{q},\dot{\mathbf{q}}\}$

Fig. 3. Block diagram of the method.

Fig. 3 with numbers 1 and 2 inside a circle, respectively. Level 3 (medium-priority task) is used to reduce to zero the speed of the robot tool when no operator's forces is detected by the force sensor. Finally, Level 4 (low-priority task) is considered for the case of redundant robots to keep the robot close to the so-called *home* configuration.

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

Note that the above scheme represents a hybrid position/force control

19

since position control is carried out in Level 1, Level 3 and Level 4, whereas force control is carried out in Level 2.

4.2. Procedure to apply Theorem 1

The acceleration equalities for Level 1 and Level 2 are obtained below using the SMC presented in Section 3. In particular, the following steps must be followed to apply Theorem 1:

1) Define a dynamical system in the form of Eq. (8).
2) Define the inequality constraints $\phi_i = \sigma_i + K_i \dot{\sigma}_i \leq 0$ to be satisfied.
3) Compute the lie derivatives $\mathbf{L_g}\phi_i$.
4) Establish the control equality (10) to be fulfilled, where the switching gain u^+ and switching gain weights $\mathbf{W}$ are the control parameters and the control action $\mathbf{u}$ will be computed from this equality by applying the Moore-Penrose pseudoinverse, see Section 2.3.

For the first step 1), a dynamical system is considered with state vector $\mathbf{x} = \begin{bmatrix} \mathbf{q}^{\mathrm{T}} & \dot{\mathbf{q}}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$, 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:

$$\dot{\mathbf{x}} = \begin{bmatrix} \mathbf{O} & \mathbf{I} \\ \mathbf{O} & \mathbf{O} \end{bmatrix} \mathbf{x} + \begin{bmatrix} \mathbf{0} \\ \mathbf{d}_c \end{bmatrix} + \begin{bmatrix} \mathbf{O} \\ \mathbf{I} \end{bmatrix} \mathbf{u}, \tag{22}$$

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

$$\mathbf{L_g}\phi_i = (\partial \phi_i / \partial \dot{\mathbf{q}})^{\mathrm{T}} \tag{23}$$

$$L_f \phi_i = (\partial \phi_i / \partial \mathbf{q})^{\mathrm{T}} \dot{\mathbf{q}} + (\partial \phi_i / \partial \dot{\mathbf{q}})^{\mathrm{T}} \mathbf{d}_c. \tag{24}$$

The remaining steps 2), 3) and 4) for Level 1 and Level 2 are tackled below in Section 4.3 and Section 4.4, respectively.

4.3. Level 1: Constraints for the transported object

A typical constraint imposed by the transport task is to keep a reference orientation for the object when the robot and the human operator cooperatively transport it. To accomplish this, the following three inequality constraints are defined:

$$\sigma_{1,\alpha}(\mathbf{p}) = |\alpha - \alpha_{ref}| - \alpha_{\max} \leq 0 \tag{25}$$

$$\sigma_{1,\beta}(\mathbf{p}) = |\beta - \beta_{ref}| - \beta_{\max} \leq 0 \tag{26}$$

$$\sigma_{1,\gamma}(\mathbf{p}) = |\gamma - \gamma_{ref}| - \gamma_{\max} \leq 0, \tag{27}$$

20

where $\{\alpha, \beta, \gamma\}$ and $\{\alpha_{ref}, \beta_{ref}, \gamma_{ref}\}$ are the actual and reference values, respectively, for the roll-pitch-yaw orientation angles and $\{\alpha_{\max}, \beta_{\max}, \gamma_{\max}\}$ are the maximum absolute deviation allowed for these angles.

Taking into account (19), (23) and (25)–(27), the Lie derivative $\mathbf{L_g}\phi_1$ for the first level, which is required for the SMC in (10), is given by:

$$\mathbf{L_g}\phi_1 = (\partial\phi_1/\partial\dot{\mathbf{q}})^{\mathrm{T}} = \mathbf{K}_1 \, (\partial\boldsymbol{\sigma}_1/\partial\mathbf{q})^{\mathrm{T}} = \mathbf{K}_1\mathbf{H}_1\mathbf{J}, \qquad (28)$$

where $\boldsymbol{\sigma}_1$ and ϕ_1 are column vectors composed of the three inequality constraints above, $\mathbf{K}_1$ is a diagonal matrix composed of all approaching parameters $K_{1,i}$ for the first level and:

$$\mathbf{H}_1 = \begin{bmatrix} 0 & 0 & 0 & \text{sign}(\alpha - \alpha_{ref}) & 0 & 0 \\ 0 & 0 & 0 & 0 & \text{sign}(\beta - \beta_{ref}) & 0 \\ 0 & 0 & 0 & 0 & 0 & \text{sign}(\gamma - \gamma_{ref}) \end{bmatrix}, \qquad (29)$$

where $\text{sign}(\cdot)$ represents the *sign function*.

Therefore, the acceleration equality for the first level results in:

$$\text{v2m}\,(\text{pos}\,(\phi_1))\,\mathbf{K}_1\mathbf{H}_1\mathbf{J}\ddot{\mathbf{q}}_c = -\mathbf{W}_1\text{pos}\,(\phi_1)\,u_1^+ \quad \rightarrow \quad \mathbf{A}_1\ddot{\mathbf{q}}_c = \mathbf{b}_1, \qquad (30)$$

where $\mathbf{W}_1$ and u_1^+ are the switching gain weight matrix and switching gain, respectively, for the first level and $\mathbf{A}_1$ and $\mathbf{b}_1$ are the matrix and vector for the first task in (5).

Using the proposed ASG law given by (21), the switching gain weight matrix $\mathbf{W}_1$ is adapted as follows:

$$\mathbf{W}_1 = \mathbf{W}_{1,prev} + T_s\,\mu_1\,\mathbf{f}_{ASG}(\phi_1, \phi_{1,prev}), \qquad (31)$$

where $\mathbf{W}_{1,prev}$ and $\phi_{1,prev}$ are the weight matrix and constraint function vector of the previous sample time, respectively, μ_1 is the adaptation parameter for Level 1 and function $\mathbf{f}_{ASG}(\mathbf{v}_1, \mathbf{v}_2)$ gives a diagonal matrix with the i-th diagonal element given by $f_{ASG}(v_{1,i}, v_{2,i})$.

4.4. Level 2: Constraints to track human operator's forces

The second level includes the inequality constraints required for the robot to be guided by the human operator when both cooperatively transport the object. These constraints, defined below, depend on the vector of forces and torques measured by a force sensor located at the robot tool. In many

applications, the interaction forces between the tool and the environment (in this case the human operator) can be approximated by the ideal elastic model below [35]:

$$\mathbf{F}(\mathbf{q}, t) = \mathbf{K}_s \, \Delta\mathbf{s}(\mathbf{q}, \mathbf{p}_s) = \begin{bmatrix} F_x & F_y & F_z & F_\alpha & F_\beta & F_\gamma \end{bmatrix}^{\mathrm{T}}, \qquad (32)$$

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

The following constraint $\sigma_{2,l}$ is proposed to track the human operator's forces in the *linear* axes:

$$\sigma_{2,l}(\mathbf{F}) = \sigma_{2,l}(\mathbf{q}, t) = \sqrt{F_x^2 + F_y^2 + F_z^2} - F_{l,0} = F_l - F_{l,0} \le 0, \qquad (33)$$

where the constraint becomes active when the magnitude F_l of the linear forces detected by the sensor is larger than the threshold $F_{l,0}$, in which case the robot tool is moved by the proposed SMC in the direction of the detected linear forces in order to fulfill the constraint. Note that the smaller is threshold $F_{l,0}$, the less effort is for the human operator to move the transported object, i.e., the robot tool.

Taking into account (19), (23) and (33), the Lie derivative $\mathbf{L_g}\phi_{2,l}$ for the above constraint, which is required for the SMC in (10), is given by:

$$\mathbf{L_g}\phi_{2,l} = K_{2,l} \, (\partial\sigma_{2,l}/\partial\mathbf{q})^{\mathrm{T}} = K_{2,l} \begin{bmatrix} \overline{F}_x & \overline{F}_y & \overline{F}_z & 0 & 0 & 0 \end{bmatrix} \mathbf{K}_s \mathbf{J}_n, \qquad (34)$$

where $K_{2,l}$ is the approaching parameter for the constraint $\sigma_{2,l}$, $\overline{F}_i = F_i/F_l$ represents the normalized linear force in the i axis and $\mathbf{J}_n$ is the geometric Jacobian relative to the tool coordinate system [35], i.e., the Jacobian matrix relating the joint velocities $\dot{\mathbf{q}}$ and the linear and angular velocities of the end-effector relative to the tool coordinate system.

Similarly, other constraints can be considered in this level if the human operator is also allowed to change the orientation of the transported object. For instance, if the object can be rotated by the human operator in the yaw

axis, the following is considered:

$$\sigma_{2,\gamma}(\mathbf{F}) = \sigma_{2,\gamma}(\mathbf{q}, t) = |F_\gamma| - F_{\gamma,0} \le 0, \tag{35}$$

where $F_{\gamma,0}$ represents the torque threshold in the yaw axis and, hence, the Lie derivative $\mathbf{L_g}\phi_{2,\gamma}$ for this constraint results in:

$$\mathbf{L_g}\phi_{2,\gamma} = K_{2,\gamma} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \text{sign}(F_\gamma) \end{bmatrix} \mathbf{K}_s \mathbf{J}_n. \tag{36}$$

where $K_{2,\gamma}$ is the approaching parameter for the constraint $\sigma_{2,\gamma}$.

Note that the robot degree-of-freedom required to control the constraint given by (35) is only available at the second level if the constraint (27) in the first level is not included or not active.

The acceleration equality for the second level results in:

$$\text{v2m}\left(\text{pos}\left(\boldsymbol{\phi}_2\right)\right) \mathbf{L_g}\phi_2 \ddot{\mathbf{q}}_c = -\mathbf{W}_2 \,\text{pos}\left(\boldsymbol{\phi}_2\right) u_2^+, \tag{37}$$

where $\boldsymbol{\phi}_2$ is a column vector composed of the inequality constraints above, $\mathbf{L_g}\phi_2$ is a matrix composed of the row vectors $\mathbf{L_g}\phi_i$ of the constraints above, and $\mathbf{W}_2$ and u_2^+ are the switching gain weight matrix and switching gain, respectively, for the second level.

Since the stiffness coefficients $\mathbf{K}_s$ in $\mathbf{L_g}\phi_{2,l}$ (34) and $\mathbf{L_g}\phi_{2,\gamma}$ (36) may not be known, they can be included in the switching gain weight matrix $\mathbf{W}_2$, so that the SMC given by (37) is modified as follows:

$$\text{v2m}\left(\text{pos}\left(\boldsymbol{\phi}_2\right)\right) \mathbf{K}_2 \mathbf{H}_2 \mathbf{J}_n \ddot{\mathbf{q}}_c = -\overline{\mathbf{W}}_2 \text{pos}(\boldsymbol{\phi}_2) u_2^+ \quad \rightarrow \quad \mathbf{A}_2 \ddot{\mathbf{q}}_c = \mathbf{b}_2, \tag{38}$$

where $\mathbf{A}_2$ and $\mathbf{b}_2$ are the matrix and vector for the second task in (5), $\mathbf{K}_2$ is a diagonal matrix composed of all approaching parameters $K_{2,i}$ for the second level and:

$$\mathbf{H}_2 = \begin{bmatrix} \overline{F}_x & \overline{F}_y & \overline{F}_z & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \text{sign}(F_\gamma) \end{bmatrix} \tag{39}$$

$$\overline{\mathbf{W}}_2 = \begin{bmatrix} W_{2,l}/K_{s,l} & 0 \\ 0 & W_{2,\gamma}/K_{s,\gamma} \end{bmatrix} = \begin{bmatrix} \overline{W}_{2,l} & 0 \\ 0 & \overline{W}_{2,\gamma} \end{bmatrix}, \tag{40}$$

where it has been assumed the same stiffness coefficient $K_{s,l}$ for all the linear coordinates and, hence, the linear motion of the robot tool given by (38) is in the same direction as the human operator's forces, which are detected by

23

the tool sensor[3].

Using the proposed ASG law given by (21), the switching gain weight matrix $\overline{\mathbf{W}}_2$ is adapted as follows:

$$\overline{\mathbf{W}}_2 = \overline{\mathbf{W}}_{2,prev} + T_s \, \mu_2 \, \mathbf{f}_{ASG}(\boldsymbol{\phi}_2, \boldsymbol{\phi}_{2,prev}), \qquad (41)$$

where $\overline{\mathbf{W}}_{2,prev}$ and $\boldsymbol{\phi}_{2,prev}$ are the weight matrix and constraint function vector of the previous sample time, respectively, and μ_2 is the adaptation parameter for Level 2.

4.5. Level 3: Speed reduction

Note that Level 2 computes the joint accelerations required to track the forces exerted by the human operator. Thus, when the sensor detects no forces the accelerations given by Level 2 are zero, but in general the robot speed is not zero. Hence, it is required to reduce the speed to zero to stop the robot motion. In particular, the following equality is considered to reduce the tool speed to zero:

$$\mathbf{J}\ddot{\mathbf{q}}_c = -\, K_{3,v}\dot{\mathbf{p}} - \mathrm{sign}(\dot{\mathbf{p}})u_3^+ \;\; \rightarrow \;\; \mathbf{A}_3\ddot{\mathbf{q}}_c = \mathbf{b}_3, \qquad (42)$$

where the Cartesian speed of the robot tool $\dot{\mathbf{p}}$ is obtained from the first-order kinematics (2), $K_{3,v}$ is the velocity correction gain of the continuous term in the above control law, u_3^+ is the switching gain of the conventional SMC used in the second term, and $\mathbf{A}_3$ and $\mathbf{b}_3$ are the matrix and vector for the third task in (5).

Note that, basically, the above control law uses the first continuous term for speed reduction when $\dot{\mathbf{p}} \neq \mathbf{0}$ and the second switching term to compensate the term $\dot{\mathbf{J}}\dot{\mathbf{q}}$ of the robot second-order kinematics (3) when $\dot{\mathbf{p}} = \mathbf{0}$. This hybrid control law has several advantages: the time derivative of the robot Jacobian is not required; the switching gain u_3^+ can be relatively small, reducing the chattering effects, while a fast speed convergence to zero is obtained

---

[3]If the actual values of the linear stiffness coefficients are different, the optimal direction (i.e., the gradient vector) to reduce the value of the constraint function $\sigma_{2,l}$ in order to fulfill (33) is not the same as the human operator's forces. That is, the gradient tends to the direction of the axes that have larger stiffness coefficients in order to promptly reduce the value of the constraint function. However, since the purpose of this inequality constraint is to "follow" the human operator's forces, it is convenient from a practical point of view to consider the same stiffness coefficient for all the linear coordinates.

24

due to the continuous correction term.

4.6. Level 4: Home configuration

This level is considered only for the case of *redundant* robots (e.g., the one used in the experiments in Section 8) since otherwise there are no remaining degrees of freedom at this level. Among the different available options in literature, this work considers "pushing" the robot to a home configuration $\mathbf{q}_0$ for increasing safety, in order to avoid a bias robot self-motion which may lead achieving critical areas due to, for instance, joint limits, singular configurations or possible obstacles in the robot workspace. To accomplish this purpose, the following equality is considered:

$$\ddot{\mathbf{q}}_c = -K_{4,v}\dot{\mathbf{q}} + K_{4,p}(\mathbf{q}_0 - \mathbf{q}) \quad \rightarrow \quad \mathbf{A}_4\ddot{\mathbf{q}}_c = \mathbf{b}_4, \tag{43}$$

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

5. Additional remarks

5.1. 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 inequality constraints. 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.

5.2. Time derivatives

The proposed approach requires the derivatives of the roll-pitch-yaw orientation angles $\{\alpha, \beta, \gamma\}$ and force vector $\mathbf{F}$ for the SMC in the first and second levels, respectively. On the one hand, the derivative of the orientation angles can be computed from the robot Jacobian matrix and the joint velocities obtained from the robot controller, see (2). On the other hand, the derivative of the force vector can be obtained using numerical differentiation, e.g., the well-known backward Euler approximation. However, some kind

of filtering should be previously applied to the measured force vector when non-negligible noise is present. It is important to remark that the low-pass filter used for noise reduction must not limit the bandwidth of the control law, see Section 8.1.

5.3. Differentiability of the constraint functions

As mentioned in Theorem 1, the inequality constraint functions ϕ_i must be differentiable around the boundary of the allowed region (given by $\phi_i = 0$) in order to compute the required Lie derivatives $\mathbf{L_g}\phi_i$ for the SM control action in (10). If this assumption is not fulfilled at a given point around the mentioned boundary, when the system state is at this point the SM behavior of the proposed method is temporarily lost and the constraints may be unfulfilled.

For instance, constraint functions $\{\sigma_{1,\alpha}, \sigma_{1,\beta}, \sigma_{1,\gamma}, \sigma_{2,\gamma}\}$ in Section 4.3 and Section 4.4 are not differentiable at $\{\alpha = \alpha_{ref}, \beta = \beta_{ref}, \gamma = \gamma_{ref}, F_\gamma = 0\}$, respectively, see (25)–(27) and (35). This is due to the absolute value function $|\cdot|$ used in the constraint definition, i.e., the absolute value function has been used to combine two differentiable constraints into a single unified constraint: $x - x_{\max} \leq 0$ and $-x - x_{\max} \leq 0$ are combined into constraint $|x| - x_{\max} \leq 0$, which is not differentiable at point $x = 0$. However, the point remains within the allowed region and far away from its boundary[4], which is given by $x = x_{\max}$ and $x = -x_{\max}$. Therefore, the non-differentiable points described above do not represent a real concern to properly compute the SM control action given by (10) when the inequality constraints become active around the boundary of the allowed region.

5.4. Advantages of the proposed method

The first advantage of the proposed multitask SMC is *complementarity*: one task is in charge of achieving the reference values required by the transport task, while another task is in charge of controlling the robot force in order to track the human operator's forces when both cooperatively transport the object. Moreover, other advantages of the proposed one-side SMC to satisfy the inequality constraints in the first and second levels are:

---

[4]Note that if $x_{\max}$ tends to zero the allowed region is negligible and, hence, it makes little sense to consider an inequality constraint, and the *equality* constraint $x = 0$ should be considered instead, which can then be fulfilled using conventional SMC.

- *Smoothness*: firstly, the joint velocities are continuous since the SM control action are the joint accelerations; and, secondly, the constraints boundary is reached progressively depending on a free design parameter, i.e., the velocity perpendicular to the constraint manifold is progressively reduced to zero.

- *Robustness*: the algorithm is robust against the Lie derivatives $L_f\phi_i$ since they are collinear [9] 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 $\mathbf{p}_s$ of the robot environment (i.e., the human operator guiding the robot tool) and its derivative; the time derivative of the Jacobian matrix $\mathbf{J}$; 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 one-side SMC for Level 1 is given by (30) and only requires the robot Jacobian and the roll-pitch-yaw orientation angles of the robot tool and their time derivatives. Similarly, the one-side SMC for Level 2 is given by (38) and only requires the robot Jacobian and the force sensor measurement $\mathbf{F}$ and its time derivative. Hence, the proposed approach only requires a few program lines and has reduced computation time, see Section 7.

As is the case with other SMC-based controllers, the main disadvantage of the method is the chattering drawback, see Section 3.6, although this problem becomes negligible for reasonable fast sampling rates.

6. Simulation

In this section, the main features of the proposed adaptive one-side SMC are illustrated by simulating a DC motor model, as it allows to better highlighting the merits of the proposed controller with a traditional example, and show its performance with full command of the conditions. Note that this model represents a joint of the robot system and can be easily extrapolated to a given number of joints, as considered in the real experimentation of Section 8.

The DC motor model is given by:

$$\dot{\omega} = (K_t\, i - b\,\omega)/J \tag{44}$$

$$\dot{i} = (-R\, i + u_m - K_e\,\omega)/L, \tag{45}$$

were ω is the rotor angular velocity, i is the armature current, u_m is the voltage source, J is the moment of inertia of the rotor, b is the motor viscous friction constant, K_e is the electromotive force constant, K_t is the motor torque constant, R is the electric resistance and L is 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 u_m.

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

$$\sigma_m(\omega) = |e| - e_{\max} \leq 0, \qquad \phi_m(\omega, \dot{\omega}) = \sigma_m + K_m \dot{\sigma}_m \leq 0, \tag{46}$$

where K_m is the approaching parameter to the original constraints and $e_{\max}$ is the maximum allowed tracking error. Note that the relative degree between the constraint function ϕ_m and the control action u_m is equal to one, as required by SM control theory.

Therefore, the Lie derivative $L_g\phi$ and the control equality (10) for the one-side SMC result in:

$$L_g\phi_m = -(K_m K_t/L)\,\text{sign}(e) \tag{47}$$

$$\text{pos}(\phi_m)\text{sign}(e)\, u_m = \text{pos}(\phi_m)u^+, \tag{48}$$

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

By applying the Moore-Penrose pseudoinverse (see Section 2.3) in (48), the control action u_m results in:

$$\begin{aligned}
u_m &= (\text{pos}(\phi_m)\text{sign}(e))^{\dagger}\text{pos}(\phi_m)u^+ = (\text{sign}(e))^{\dagger}\text{pos}(\phi_m)u^+ \\
&= \text{sign}(e)\text{pos}(\phi_m)u^+,
\end{aligned} \tag{49}$$

where it has been taken into account that $(0)^{\dagger} = 0$.

It is interesting to remark that the constraint function in (46) is not differentiable at point $e = 0$. This however bears no significance to properly compute the SM control action with (49) when the inequality constraint

becomes active since this point is far away from the boundary of the allowed region, which is given by $e = e_{\max}$ and $e = -e_{\max}$, see Section 5.3.

6.1. Simulation conditions

The simulation was run under the following conditions:

- Parameters used for the DC motor model: $J(t_0) = 0.1$ kg·m^2, $b = 0.1$ N·m·s, $K_e = 0.01$ V·s, $K_t = 0.01$ N·m/A, $R = 1$ Ω and $L = 0.5$ H.

- The reference speed is a sinusoidal wave given by $\omega_{ref}(t) = 25 + 25\sin(1.33\pi t - \pi/2))$ deg/s.

- The approaching parameter K_m is equal to 0.1.

- The maximum allowed tracking error $e_{\max}$ for the inequality constraints is equal to 2 deg/s.

- The parameters used for the switching gain are:

    - FSG (fixed switching gain) High: $u^+ = 200$
    - FSG Low: $u^+ = 45$
    - ASG (adaptive switching gain): $u^+(0) = 45$ and $\mu_m = 3000$.

- The simulation period T_s was set to one millisecond

The simulation results presented below were obtained using MATLAB$^{\circledR}$.

6.2. Results for the adaptive one-side SMC

Fig. 4 and Fig. 5 show the behavior of the one-side SMC with a fixed switching gain (FSG), either using a *high* or a *low* gain, and with the proposed ASG algorithm. It can be seen in the top graph of Fig. 4 that the tracking error for the FSG case with low switching gain is larger than the maximum allowed error $e_{\max}$ during some phases of the simulation. This is because the SM is lost, as depicted by the commutation function $\text{pos}(\phi_m)$ in the third graph (from top) of Fig. 5(b). This is indicative of a value for the switching gain that is not large enough. In contrast, for both the case of FSG with high switching gain and the ASG method, tracking error is contained within the allowed band (refer to the top graph in Fig. 4), and the SM works properly, as indicated by the commutation function in the third graph of Fig. 5(a)

29
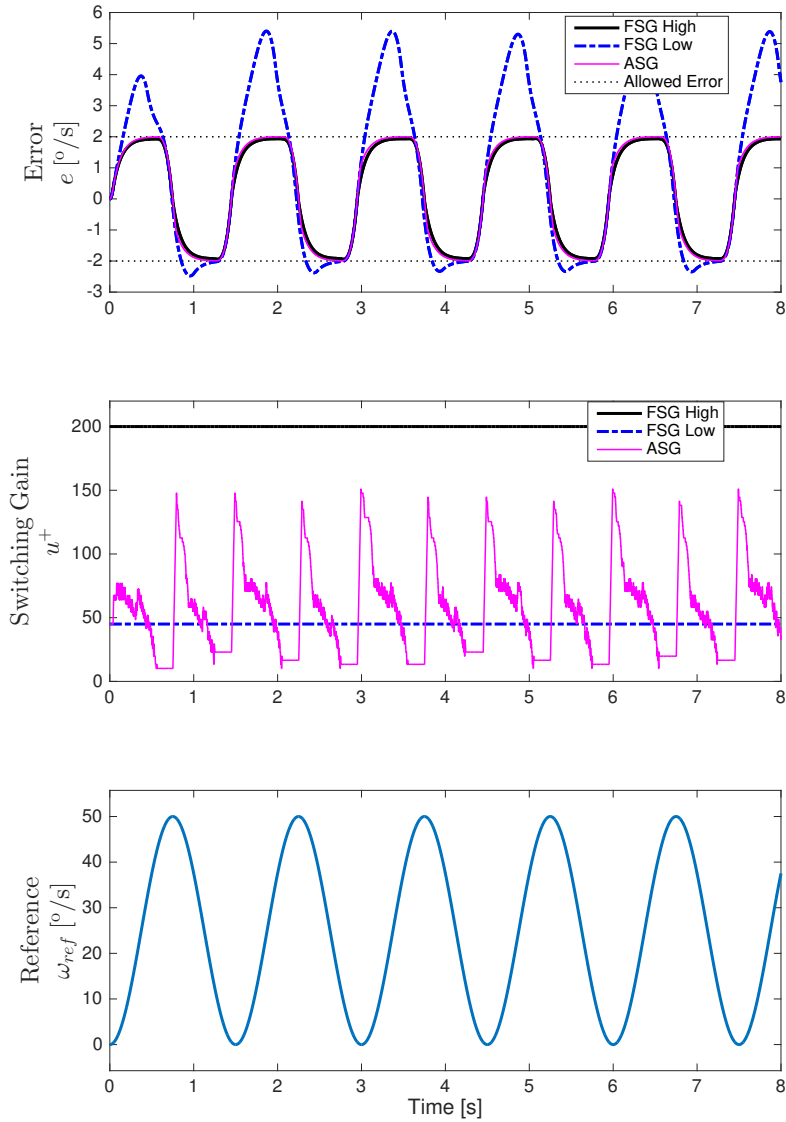
Fig. 4. Simulation of a motor control using the one-side SMC with a sinusoidal reference velocity: comparison of FSG and ASG methods in terms of error and switching gain.

30
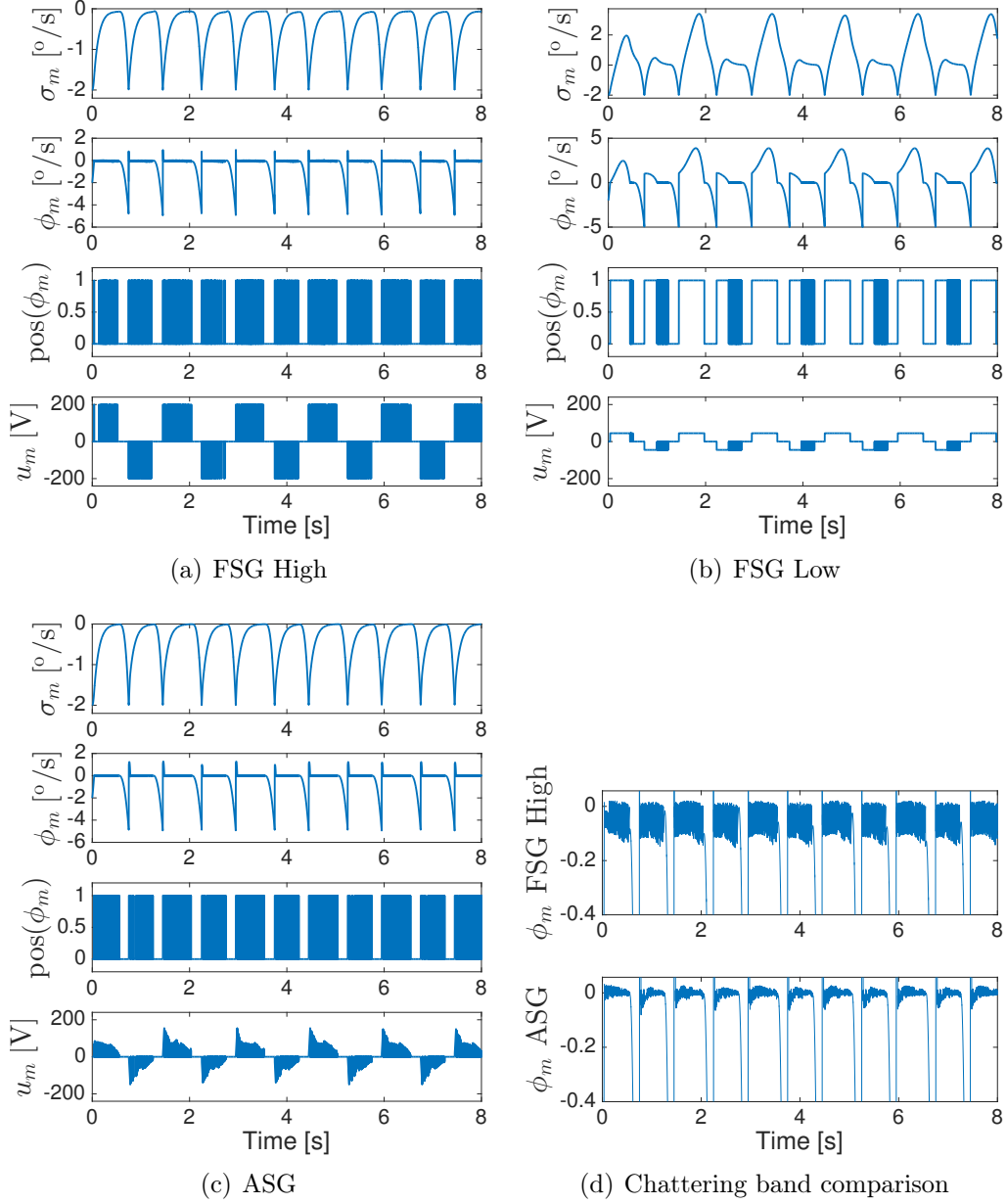
Fig. 5. Simulation of a motor control using the one-side SMC: Detail of constraint functions σ_m and ϕ_m, commutation function $\mathrm{pos}(\phi_m)$ and control action u_m for each method.

and Fig. 5(c). However, as illustrated by the middle graph in Fig. 4, the ASG method employs a lower value for the switching gain. As such, the final values for the control action u_m (49) and chattering band obtained with the ASG method are smaller than those required by the case of FSG with high gain. This effect can be seen in the bottom graphs of Fig. 5(a) and Fig. 5(c), and the detail view of the constraint functions ϕ_m in Fig. 5(d), respectively.

Therefore, according to the above simulation analysis it can be concluded that the proposed adaptive one-side SMC has a good performance and adapts effectively the switching gain in order to reduce the control effort and the chattering amplitude.

6.3. Comparison with conventional SMC

The proposed one-side SMC is compared in this section with the conventional SMC, whose control action $\bar{u}_m$ for the motor control is given by:

$$\bar{u}_m = \text{sign}(\phi_m)u^+. \tag{50}$$

The simulation was run under the same conditions indicated in Section 6.1 except for the following: reference speed $\omega_{ref}(t) = 5 + 25\sin(1.33\pi t)$ deg/s and switching gain parameters $u^+(0) = 250$ and $\mu_m = 0$, i.e., a FSG is considered for simplicity.

Fig. 6 shows the simulation results for both conventional and one-side SMCs. It can be seen that, once the initial error has been corrected (reaching mode), the conventional SMC keeps the error signal at zero (top graph) switching the control action value from positive to negative and vice versa (second graph), which is known as SM. In contrast, the one-side SMC only applies a correction (third graph), i.e., the value of the control signal is not zero, when the error signal is about to leave the allowed region (top graph) in order to keep the system within the limits. Therefore, it can be concluded that the proposed one-side SMC has a good performance and is especially suitable for non-negligible allowed regions, since otherwise the behavior is very similar to that obtained with conventional SMC. It is important to remark that when the one-side SMC applies no correction, the control signal can be used to achieve other goals of lower-priority tasks, see Section 2.

As in the case of conventional SMC, the proposed one-side SMC method suffers from the chattering drawback, see Section 3.6 and Section 5.4. However, this issue is partially mitigated in this work introducing a first-order low-pass filter between the original constraint function σ_i and the modified
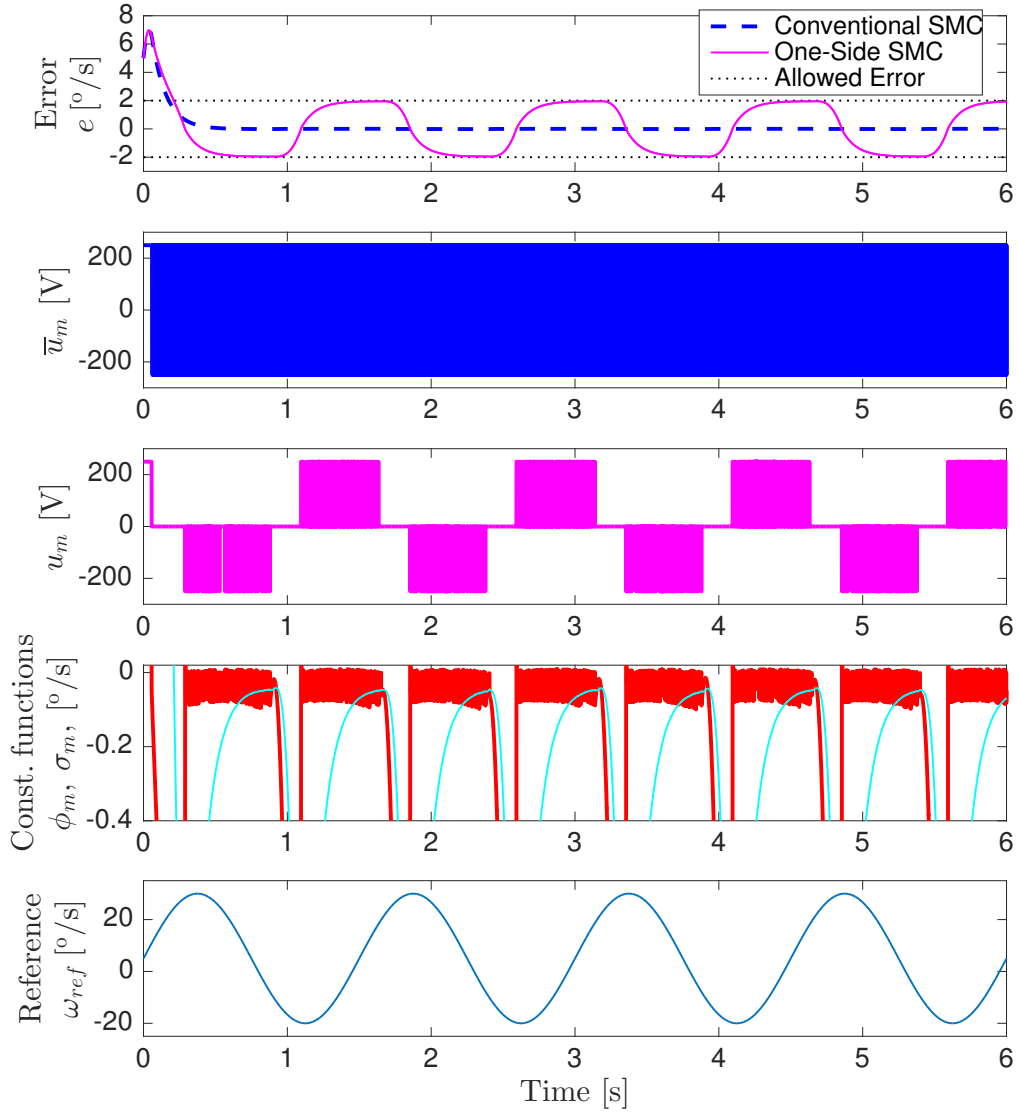
32

Fig. 6. Simulation of a motor control using conventional and one-side SMCs with a sinusoidal reference velocity. Top graph: errors. Second and third graphs: control action for the conventional and one-side SMCs, respectively. Fourth graph: constraint functions ϕ_m (thick line) and σ_m (thin line) for the one-side SMC. Bottom graph: reference velocity.

33

constraint function ϕ_i, see (19). Therefore, although ϕ_i theoretically switches its value every sampling period when the inequality constraint is active, the original constraint function σ_i remains *smooth*, see the fourth graph in Fig. 6.

7. Controller implementation

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

- Kinematic function $\mathbf{l}(\mathbf{q})$ and Jacobian matrices $\mathbf{J}$ and $\mathbf{J}_n$.

- Moore-Penrose pseudoinverse $(\cdot)^\dagger$ using a tolerance to set to zero the very small singular values, see Section 2.

- Robot sensors: *GetRobotStateAndForce*, which returns the current robot state $\{\mathbf{q}, \dot{\mathbf{q}}\}$ and the force $\mathbf{F}$ detected by the sensor, which has already been filtered by the sensor electronics.

- Actuators: *SendToJointControllers*$(\dot{\mathbf{q}}_c)$, which sends the current commanded joint velocity vector to the joint controllers.

The computation time per iteration of the algorithm using compiled C code in a computer with Intel Core i5-3470 processor at 3.2 GHz clock frequency was around 15 microseconds for the experiments in Section 8.

8. Real experimentation

The setup used for the experiments consists of (Fig. 7): a Sawyer collaborative robot in floor-mounted position; a Force/Torque Sensor Nano25 attached to the robot end-effector; a tool consisting of a joystick and a flat rectangular plastic object of 190x95x3mm attached to the sensor; and a shot glass with liquid.

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.

Algorithm executed at sampling time of T_s seconds	

1 $[\mathbf{q}, \dot{\mathbf{q}}, \mathbf{F}]$ =GetRobotStateAndForce;

2 $\mathbf{p} = \mathbf{l}(\mathbf{q})$; `// Eq. (1)`

3 $\dot{\mathbf{p}} = \mathbf{J}\dot{\mathbf{q}}$; `// Eq. (2)`

4 $\dot{\mathbf{F}} = (\mathbf{F} - \mathbf{F}_{prev})/T_s$; `// Derivative`

5 $\boldsymbol{\phi}_1 = \begin{bmatrix} |\alpha - \alpha_{ref}| - \alpha_{\max} + K_{1,\alpha}\,\mathrm{sign}(\alpha - \alpha_{ref})\,\dot{\alpha} \\ |\beta - \beta_{ref}| - \beta_{\max} + K_{1,\beta}\,\mathrm{sign}(\beta - \beta_{ref})\,\dot{\beta} \\ |\gamma - \gamma_{ref}| - \gamma_{\max} + K_{1,\gamma}\,\mathrm{sign}(\gamma - \gamma_{ref})\,\dot{\gamma} \end{bmatrix}$; `// Eqs. (19),(25)-(27)`

6 $\boldsymbol{\phi}_2 = \begin{bmatrix} \sqrt{F_x^2 + F_y^2 + F_z^2} - F_{l,0} + K_{2,l}\dfrac{\dot{F}_x F_x + \dot{F}_y F_y + \dot{F}_z F_z}{\sqrt{F_x^2 + F_y^2 + F_z^2}} \\ |F_\gamma| - F_{\gamma,0} + K_{2,\gamma}\,\mathrm{sign}(F_\gamma)\,\dot{F}_\gamma \end{bmatrix}$;

`// Eqs. (19),(33),(35)`

7 $\mathbf{W}_1 = \mathbf{W}_{1,prev} + T_s\,\mu_1\,\mathbf{f}_{ASG}(\boldsymbol{\phi}_1, \boldsymbol{\phi}_{1,prev})$; `// Eq. (31)`

8 $\overline{\mathbf{W}}_2 = \overline{\mathbf{W}}_{2,prev} + T_s\,\mu_2\,\mathbf{f}_{ASG}(\boldsymbol{\phi}_2, \boldsymbol{\phi}_{2,prev})$; `// Eq. (41)`

9 $\mathbf{A}_1 = \mathrm{v2m}\,(\mathrm{pos}\,(\boldsymbol{\phi}_1))\,\mathbf{K}_1\mathbf{H}_1\mathbf{J}$; `// Eq. (30)`

10 $\mathbf{b}_1 = -\mathbf{W}_1\,\mathrm{pos}\,(\boldsymbol{\phi}_1)\,u_1^+$; `// Eq. (30)`

11 $\mathbf{A}_2 = \mathrm{v2m}\,(\mathrm{pos}\,(\boldsymbol{\phi}_2))\,\mathbf{K}_2\mathbf{H}_2\mathbf{J}_n$; `// Eq. (38)`

12 $\mathbf{b}_2 = -\overline{\mathbf{W}}_2\,\mathrm{pos}\,(\boldsymbol{\phi}_2)\,u_2^+$; `// Eq. (38)`

13 $\mathbf{A}_3 = \mathbf{J}$; `// Eq. (42)`

14 $\mathbf{b}_3 = -K_{3,v}\dot{\mathbf{p}} - \mathrm{sign}(\dot{\mathbf{p}})u_3^+$; `// Eq. (42)`

15 $\mathbf{A}_4 = \mathbf{I}$; `// Eq. (43)`

16 $\mathbf{b}_4 = -K_{4,v}\dot{\mathbf{q}} + K_{4,p}(\mathbf{q}_0 - \mathbf{q})$; `// Eq. (43)`

17 $\ddot{\mathbf{q}}_{c,1} = \mathbf{A}_1^\dagger \mathbf{b}_1$; `// Eq. (6), $i = 1$`

18 $\mathbf{N}_1 = \mathbf{I} - \mathbf{A}_1^\dagger \mathbf{A}_1$; `// Eq. (7), $i = 1$`

19 $\ddot{\mathbf{q}}_{c,2} = \ddot{\mathbf{q}}_{c,1} + (\mathbf{A}_2\mathbf{N}_1)^\dagger(\mathbf{b}_2 - \mathbf{A}_2\ddot{\mathbf{q}}_{c,1})$; `// Eq. (6), $i = 2$`

20 $\mathbf{N}_2 = \mathbf{N}_1(\mathbf{I} - (\mathbf{A}_2\mathbf{N}_1)^\dagger(\mathbf{A}_2\mathbf{N}_1))$; `// Eq. (7), $i = 2$`

21 $\ddot{\mathbf{q}}_{c,3} = \ddot{\mathbf{q}}_{c,2} + (\mathbf{A}_3\mathbf{N}_2)^\dagger(\mathbf{b}_3 - \mathbf{A}_3\ddot{\mathbf{q}}_{c,2})$; `// Eq. (6), $i = 3$`

22 $\mathbf{N}_3 = \mathbf{N}_2(\mathbf{I} - (\mathbf{A}_3\mathbf{N}_2)^\dagger(\mathbf{A}_3\mathbf{N}_2))$; `// Eq. (7), $i = 3$`

23 $\ddot{\mathbf{q}}_{c,4} = \ddot{\mathbf{q}}_{c,3} + (\mathbf{A}_4\mathbf{N}_3)^\dagger(\mathbf{b}_4 - \mathbf{A}_4\ddot{\mathbf{q}}_{c,3})$; `// Eq. (6), $i = 4$`

24 $\dot{\mathbf{q}}_c = \dot{\mathbf{q}}_{c,prev} + T_s\ddot{\mathbf{q}}_{c,4}$; `// Integration`

25 SendToJointControllers($\dot{\mathbf{q}}_c$);

26 $[\dot{\mathbf{q}}_{c,prev}, \boldsymbol{\phi}_{1,prev}, \boldsymbol{\phi}_{2,prev}, \mathbf{W}_{1,prev}, \overline{\mathbf{W}}_{2,prev}, \mathbf{F}_{prev}] = [\dot{\mathbf{q}}_c, \boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \mathbf{W}_1, \overline{\mathbf{W}}_2, \mathbf{F}]$
; `// For next iteration`

8.1. Procedure to design the control parameters

Next, the various signal rates and bandwidths involved in the design of the proposed control method are discussed. For this purpose, the following

Fig. 7. Experimental setup: 7R serial manipulator in floor position with a force sensor rigidly attached to the robot end-effector, a tool consisting of a joystick and a flat rectangular plastic object of 190x95x3mm attached to the sensor and a shot glass with liquid.

frequency parameters are considered: $f_{SM} = (2T_s)^{-1}$ 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 and Level 2, which defines the approaching speed to the original constraint $\sigma_i = 0$, see (19), is equal to $-K_{1,i}^{-1}$ and $-K_{2,i}^{-1}$, respectively. The pole of the kinematic control in Level 3, which is used for speed reduction, is equal to $-K_{3,v}$. Finally, the poles of the kinematic control in Level 4, 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 $[1\ K_{4,v}\ K_{4,p}]$. Note that the bandwidth of the controlled system corresponds to the bandwidth of the aforementioned kinematic controller.

The following requirements must be fulfilled in the parameter design process. 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.

The steps and guidelines to design the control parameters 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 $f_{SM} = (2Ts)^{-1}$ is lower than the bandwidths of the sensor filter f_{FC} and robot controller f_{RC}.

3rd) The approaching parameters $K_{1,i}$ and $K_{2,i}$ should be as low as possible to reduced the chattering of the original constraint function σ_i, see (19), but it also has to guarantee that the bandwidths $f_{KC,1} = K_{1,i}^{-1}$ and $f_{KC,2} = K_{2,i}^{-1}$ are significantly lower than the SM frequency f_{SM}. The proposed relationship $f_{KC,1} = f_{KC,2} \approx f_{SM}/4$ has been used in the work hereby presented with successful results.

4th) The parameter $K_{3,v}$ is chosen to guarantee that the bandwidth $f_{KC,3} = K_{3,v}$ of the speed reduction fulfills the relationship $f_{KC,3} \approx f_{SM}/5$, since successful experimental results have been obtained using this relationship.

5th) It is proposed to use $K_{4,v} \approx 3\sqrt{K_{4,p}}$ to obtain a fast overdamped response (note that the critically damped response is given by $K_{4,v,\text{crit}} = 2\sqrt{K_{4,p}}$). Since Level 3 is used to control robot self-motion, its dynamics can be relatively slow without degrading robot performance. Hence, the parameter $K_{4,p}$ is selected to fulfill the relationship $f_{KC,4} \approx f_{SM}/15$.

6th) The threshold force $F_{l,0}$ and torque $F_{\gamma,0}$ are established small enough to be sensitive to the operator's forces and torques, but not too small to prevent the accidental activation of the algorithm, e.g., due to sensor noise. Some examples are given below for the practical scenarios presented.

7th) The switching gains and switching gain weights are empirically tuned as small as possible to reduce the chattering effect whilst guaranteeing that the SM behavior of the control action remains effective for a typical case of 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.

8.2. Experiment conditions

The control parameters shown below were selected using the steps and practical guidelines detailed above.

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 8.1), 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 8.1).

iii) Parameters used for Level 1 (see 3rd and 7th steps in Section 8.1): $\beta_{ref} = 0$, $\alpha_{ref} = 180°$, γ unconstrained, $\alpha_{\max} = \beta_{\max} = 1°$, $\mathbf{K}_1 = 0.05\,\mathbf{I}$, $\mathbf{W}_1 = \mathbf{I}$, $u_1^+ = 0.005$ and $\mu_{1,i} = 0$, i.e., no adaptation is considered for Level 1 since the required range of variation in this level for the SM control action is small.

iv) Parameters used for Level 2 (see 3rd, 6th and 7th steps in Section 8.1): $F_{l,0} = 2$ N, $F_{\gamma,0} = 0.2$ N·m, $\mathbf{K}_2 = 0.05\mathbf{I}$, $\overline{W}_{2,l} = 5$, $\overline{W}_{2,\gamma} = 4$, $\mu_{2,l} = 0.04$, $\mu_{2,\gamma} = 0.02$ and $u_2^+ = 0.05$.

v) Parameters used for Level 3 (see 4th and 7th steps in Section 8.1): $K_{3,v} = 5$ and $u_3^+ = 1$.

vi) Parameters used for Level 4 (see 5th step in Section 8.1): $K_{4,v} = 3$, $K_{4,p} = 1$ and $\mathbf{q}_0 = \begin{bmatrix} 0.17° & -67.15° & -0.12° & 124.73° & 0.12° & 32.32° & 190.22° \end{bmatrix}^{\mathrm{T}}$.

vii) For the experiments below the robot system has been initialized in the allowed region, as per the assumption stated in Section 3.4. For this purpose, a smooth continuous control has been previously used to take the robot to within the allowed region defined by the tool orientation angles.

8.3. Results

A first experiment has been conducted to verify the performance of the proposed approach. In this experiment the reference orientation for the flat rectangular object at the tool is perpendicular to the robot Z-axis, i.e., both roll and pitch angles are constrained (Level 1) to $\alpha_{ref} = 180°$ and $\beta_{ref} = 0$, whereas the yaw angle γ remains unconstrained and can be guided by the human operator (Level 2). A shot glass with liquid is placed on the flat object in order to show the smoothness of the proposed method.

A video for the first experiment can be played at https://media.upv.es/player/?id=266fcef0-feef-11e7-a69a-ed3f85977e27, whereas Fig. 8 shows several instants of this video. The results for this experiment are depicted at different figures. Fig. 9 shows the performance of the current approach in terms of constraint functions and the activation of the inequality constraints. In particular, on the one hand the roll or pitch constraints in the first level are active around 10% of the experiment time, whereas the maximum deviation for these angles is just about 0.2 degrees, see Fig. 10, which means that the orientation for the flat object is almost perpendicular to the Z-axis all the time, as required. Note that the constraint for angle β is activated more than that for angle α, which is due to the torque in the pitch axis exerted by the weight and inertia of the shot glass with liquid. On the other hand, the constraints in Level 2 are active around 65% of the time. Note that the main objective of Level 2 is to be very sensitive to the human operator's forces and torques, which is successfully attained as shown in the video: the robot is guided by the human operator using just one finger and very small forces and torques. In sum, as can be seen in the video, the robot control prevails over the human operator for the object orientation, while the human operator forces prevail over the robot for the object position, as required by the considered transport application.

Fig. 11 shows the ASG behavior of the constraints in Level 2, which are used to guide the robot. The figure shows the variation of the computed switching gains in order to follow the human operator's forces, which are detected by the tool sensor. In particular, it can be seen how the switching gains are increased by the ASG algorithm when large force magnitudes are detected.

Fig. 12 shows the tool speeds and force magnitudes, where it can be seen (top graph) that the linear velocities are increased in some phases to track the human operator's forces (second graph) and, subsequently, they are reduced
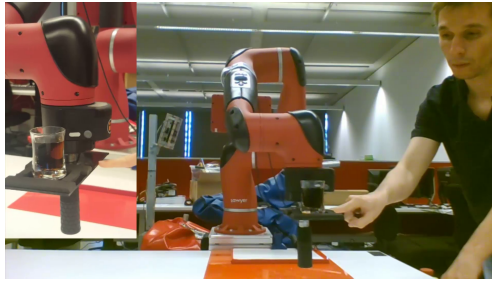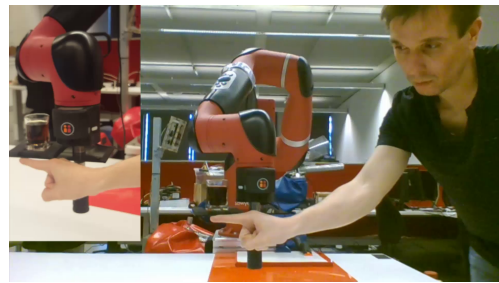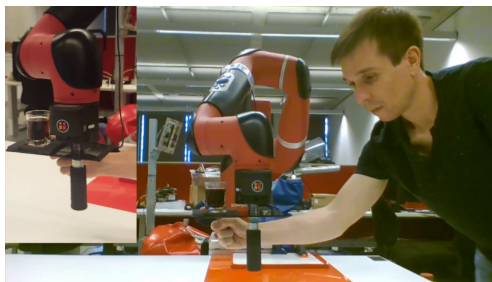
(a) video: 0m 27s; graph: 12s
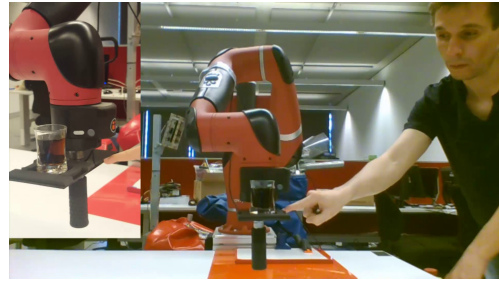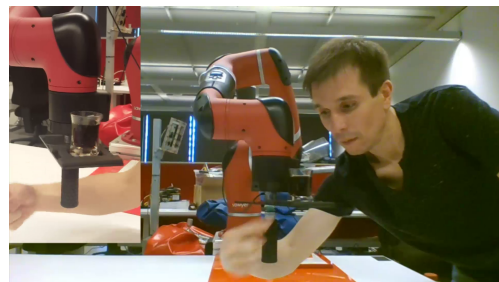
(b) video: 0m 30s; graph: 15s
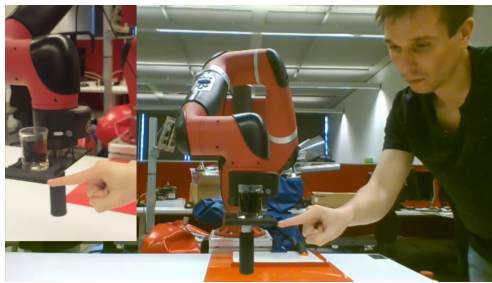
(c) video: 2m 01s; graph: 105s

(d) video: 2m 10s; graph: 114s

(e) video: 2m 14s; graph: 118s

(f) video: 2m 19s; graph: 123s

Fig. 8. Frames of the video of the first experiment, where a glass with liquid is transported cooperatively between the operator and the robot. The time instant is indicated for each frame.
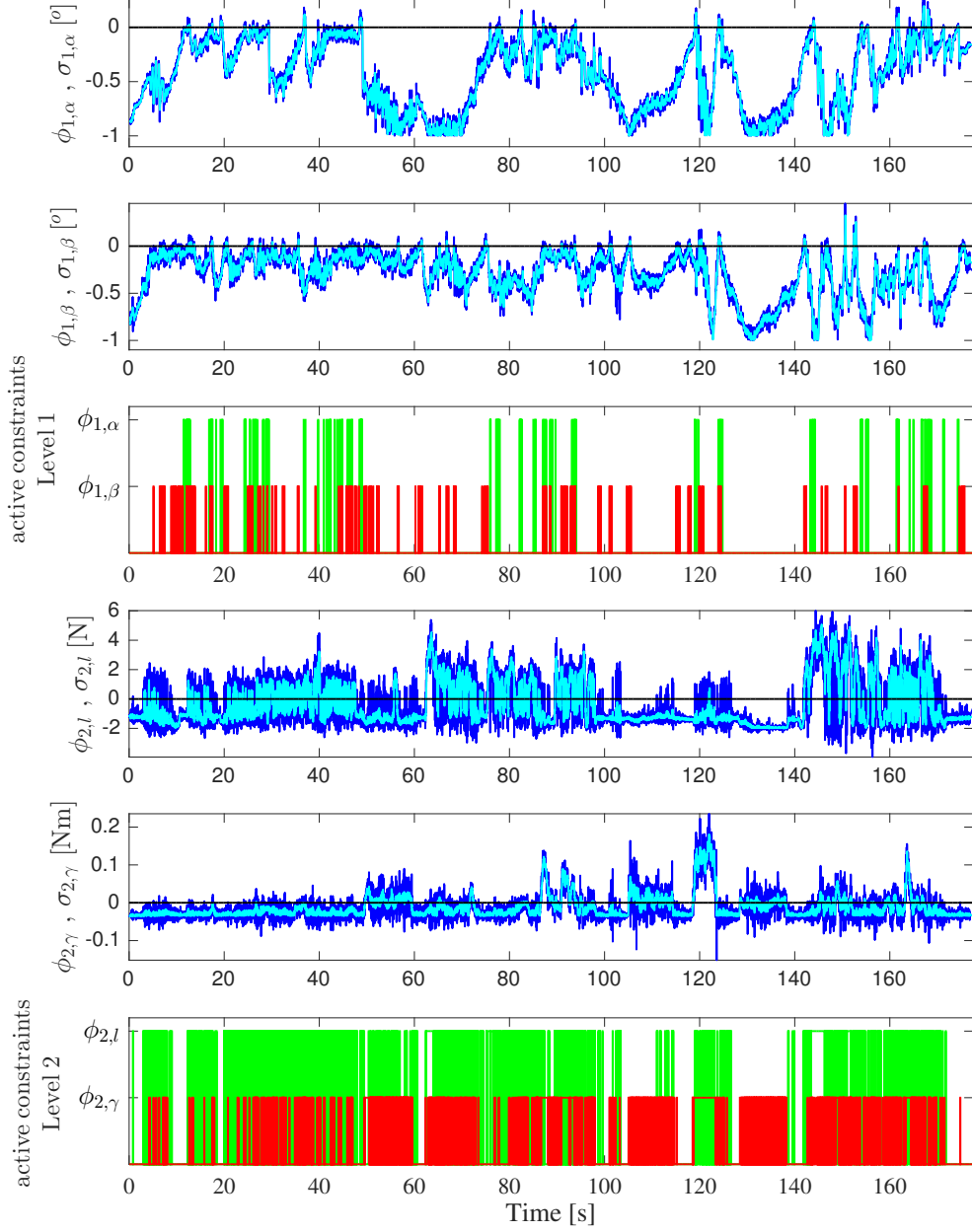
Fig. 9. Constraint functions and active constraints for the first two levels in the first experiment. The modified constraint function ϕ_i is in dark-blue, whereas the original constraint function σ_i is in light-cyan.

41
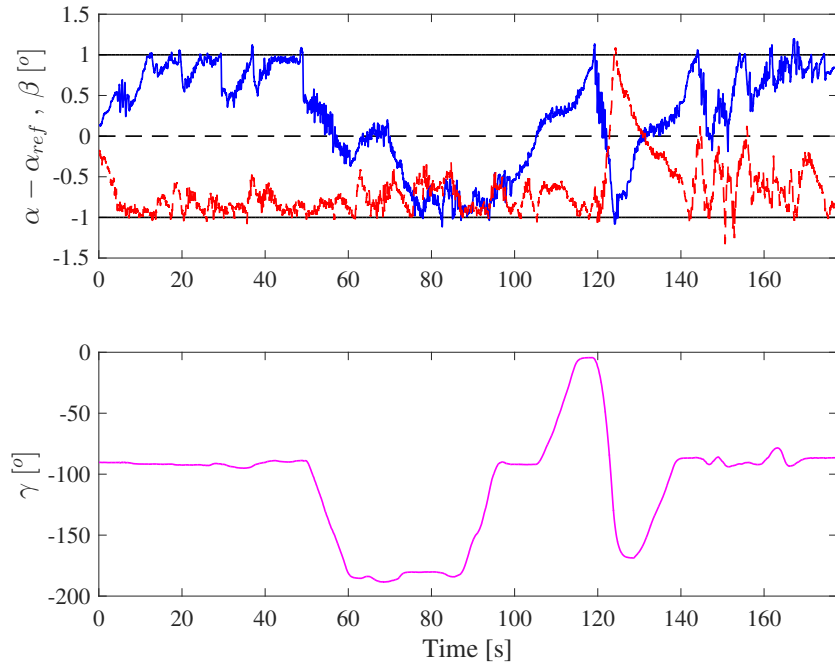
Fig. 10. Angles for the robot tool in the first experiment. Top: constrained roll (continuous-blue line) and pitch (dashed-red line) angles, where the continuous horizontal lines indicate the boundary of the allowed region, whereas the dashed horizontal lines represent the reference value for the angles. Bottom: yaw angle, which is modified to track the human operator's torques in this axis.

42
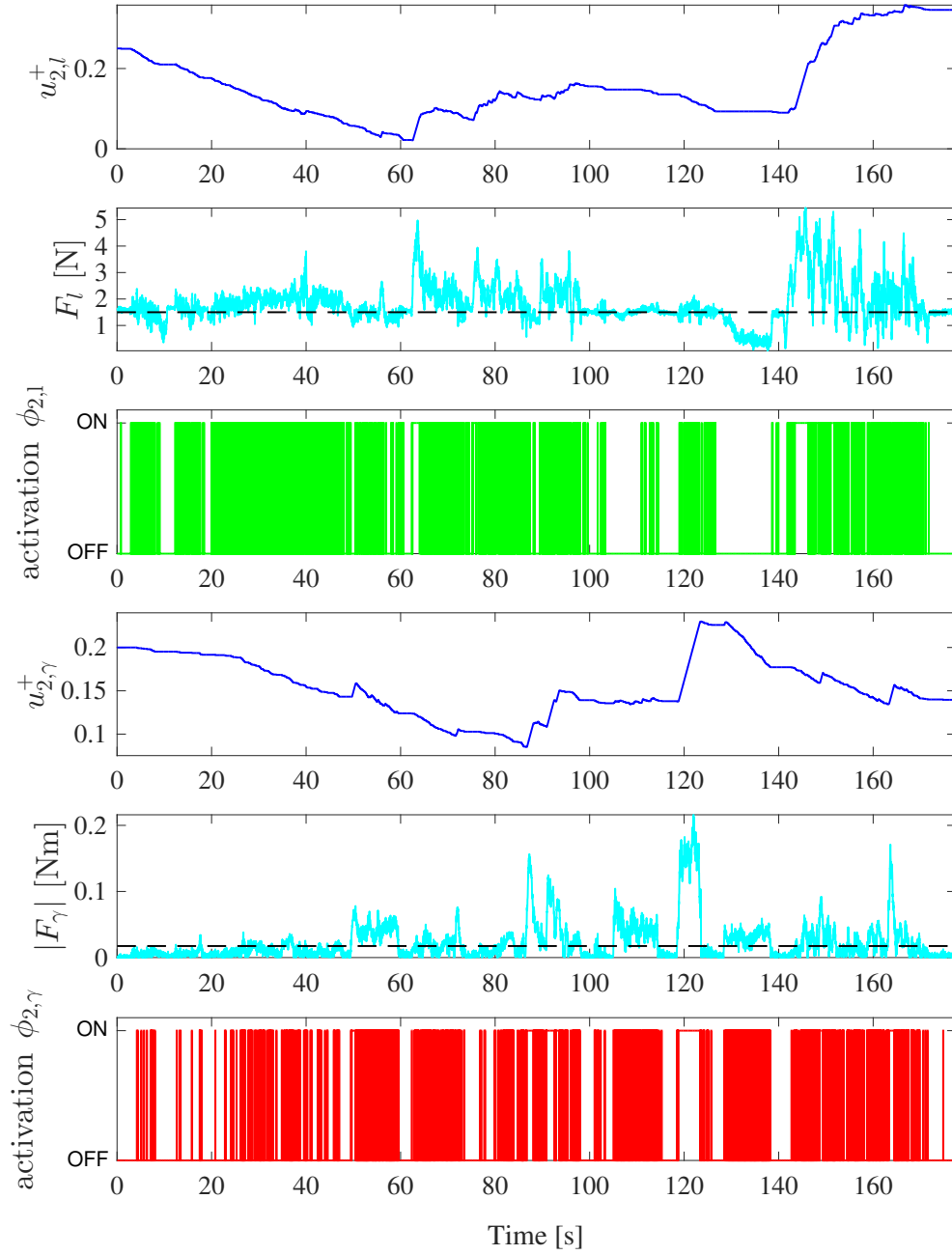
Fig. 11. Switching gain ($u_{2,i}^+ = \overline{W}_{2,i} u_2^+$), force/torque magnitude and constraint activation for the two inequality constraints in Level 2. The dashed horizontal lines represent the thresholds for the force/torque magnitudes.

43
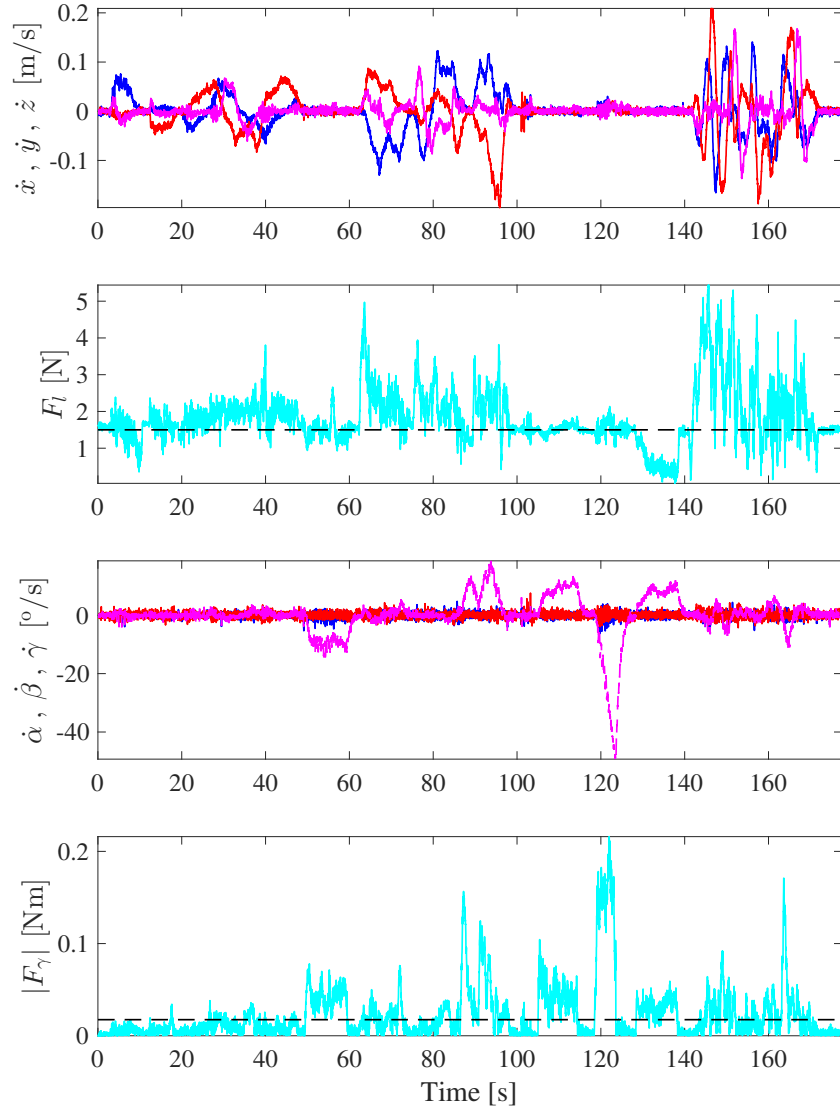
Fig. 12. Time derivative of the tool pose and force magnitudes for the first experiment. From top to bottom: linear velocities; linear force magnitude; angular velocities ($\dot{\gamma}$ dashed-magenta line); and torque magnitude for the Z-axis. The dashed horizontal lines represent the thresholds for the force/torque magnitudes.

44
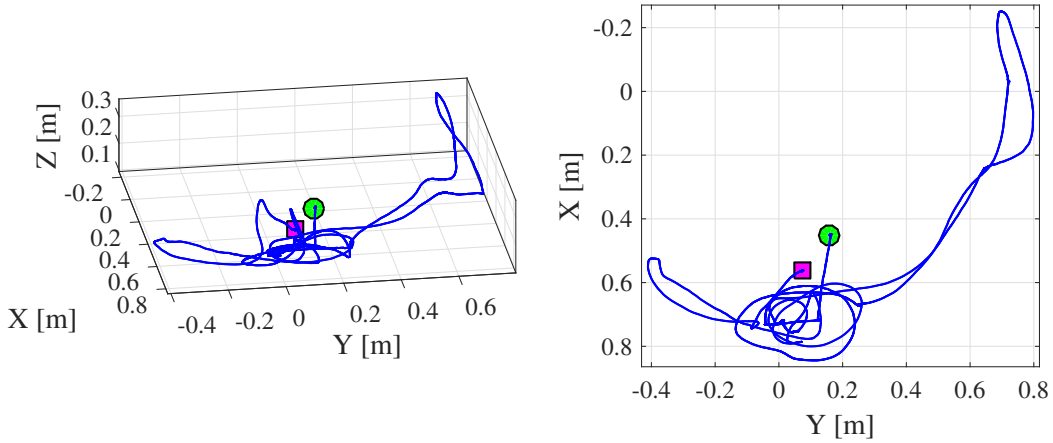
Fig. 13. Trajectory followed by the robot tool in the first experiment: left, 3D view; right, top view. Start and end points are indicated with a green circle and a magenta square, respectively.

when no forces are detected by the sensor. The same applies to the angular velocity $\dot{\gamma}$ (third graph), which is also guided by the operator (bottom graph), whereas the other two angular velocities, i.e., $\dot{\alpha}$ and $\dot{\beta}$, remain close to zero since only small corrections are needed to keep the reference orientation of the flat object.

Fig. 13 shows the trajectory followed by the robot tool, where it can be seen that a large area has been covered by the robot during the experiment: around 1 meter in both X- and Y-axes and 0.25 meters in Z-axis. Fig. 14 shows the control commands computed during the experiment. Note that all four levels contribute to the commanded joint accelerations, although those for Level 1 are smaller, since only small corrections are needed to keep the reference orientation of the flat object, as mentioned above.

A *second experiment* has been conducted in order to show the capability of the proposed approach to keep a generic reference orientation as imposed by a hypothetical collaborative handling task, while the sensed commanded action from the operator is used to guide the location of the end-effector. In particular, for this experiment the reference value for roll angle of the flat object has been changed to $\alpha_{ref} = 180° + 30°$, whereas the yaw angle has been constrained to $\gamma_{ref} = 0$ and the pitch angle remains constrained to $\beta_{ref} = 0$. The same sensor arrangement without the shot glass has also been used for this experiment.
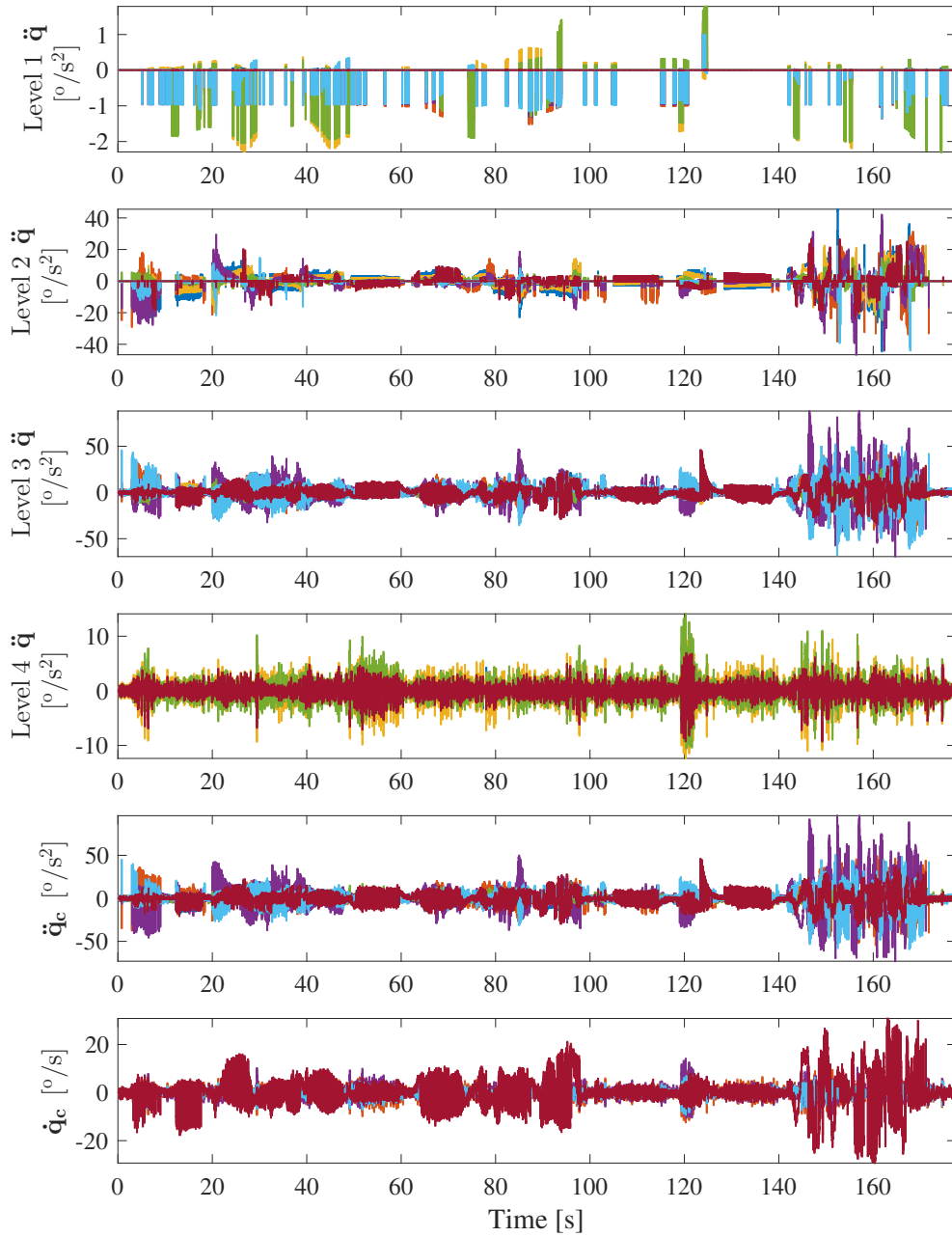
The video for the second experiment can be played at

Fig. 14. Control actions for the first experiment: contribution of each priority level to the commanded joint accelerations in the first four graphs, the commanded joint accelerations in the fifth graph, and commanded joint velocities to be sent to the robot controller in the sixth graph.
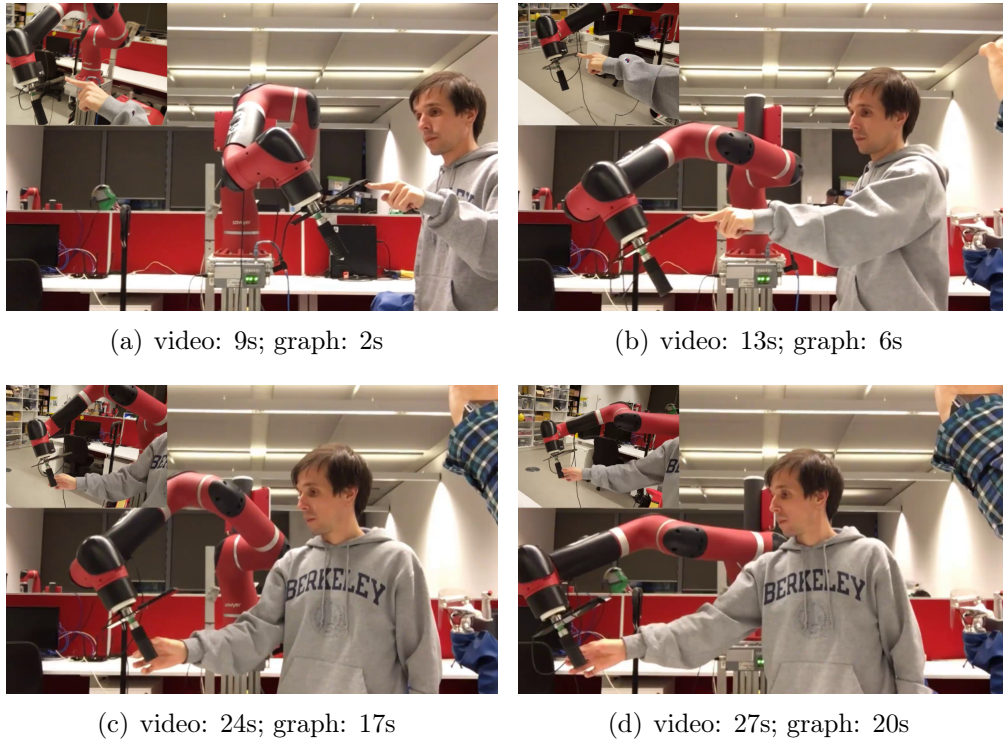
46

(a) video: 9s; graph: 2s   (b) video: 13s; graph: 6s

(c) video: 24s; graph: 17s   (d) video: 27s; graph: 20s

Fig. 15. Frames of the video of the second experiment, where a desired robot tool orientation is maintained regardless of the user indications, which drive the location of the end-effector under the action of the proposed cooperative controller. The time instant is indicated for each frame.
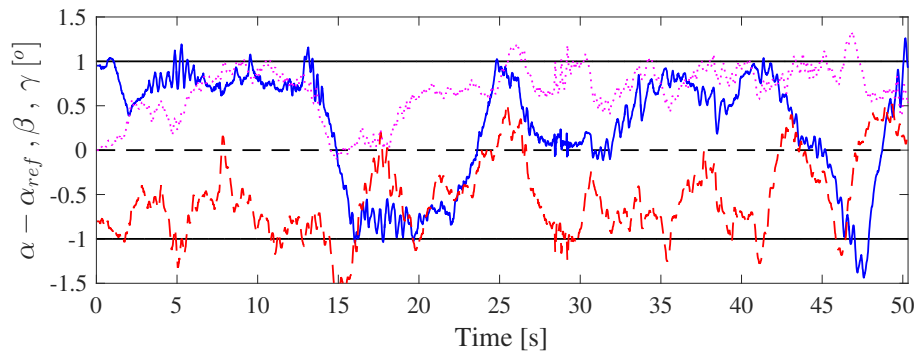

Fig. 16. Second experiment: angles for the robot tool, constrained roll (continuous-blue line), pitch (dashed-red line) and yaw (dotted-magenta line) angles. The continuous horizontal lines indicate the boundary of the allowed region, whereas the dashed horizontal lines represent the reference value for the angles.
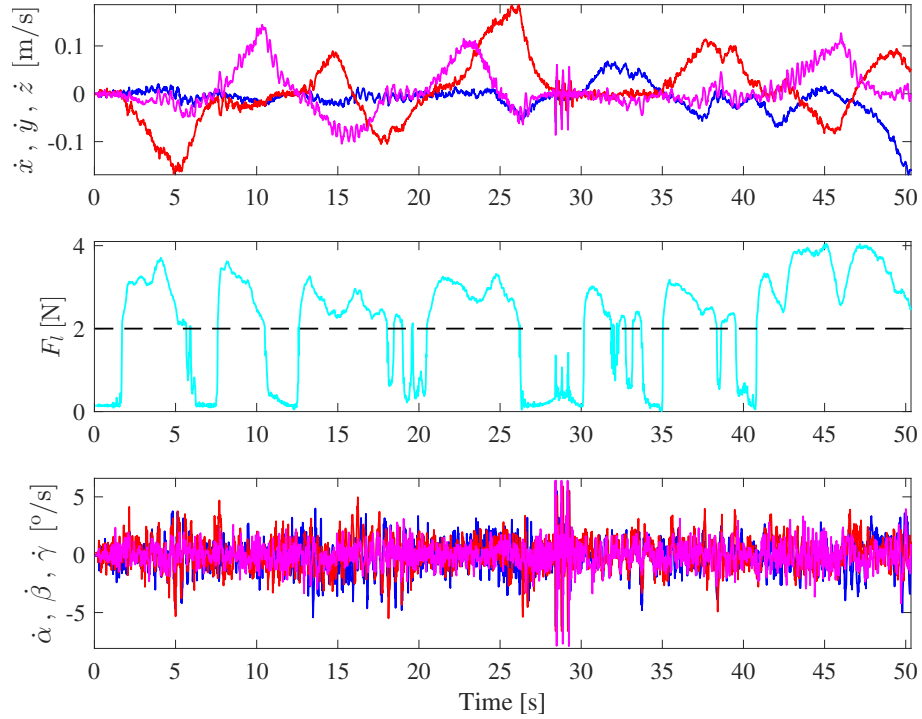
47

Fig. 17. Second experiment: Tool speeds and force magnitude. From top to bottom: linear velocities; linear force magnitude; and angular velocities. The dashed horizontal line represents the threshold for the force magnitude.
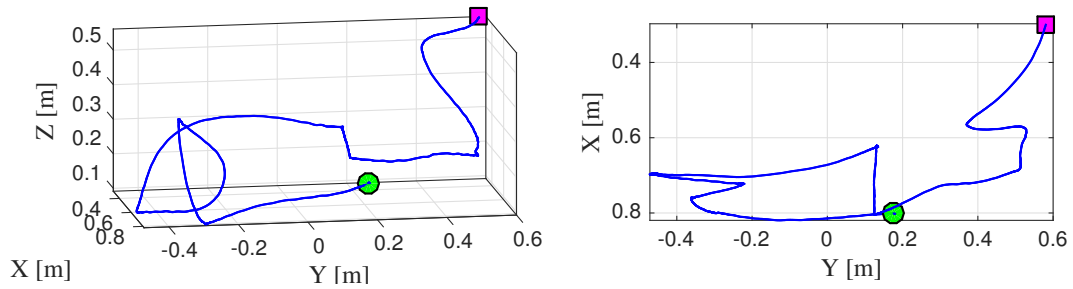

Fig. 18. Second experiment: Trajectory followed by the robot tool: left, 3D view; right, top view. Start and end points are indicated with a green circle and a magenta square, respectively.

48

https://media.upv.es/player/?id=d4f6d6b0-2e53-11e8-b43a-51b816915a74.
Fig. 15 shows several instants of the video. The maximum deviation for the angles is, as before, about 0.5 degrees, see Fig. 16. Fig. 17 shows the tool speeds, where it can be seen that the linear velocities (top graph) are increased to track the human operator's forces detected by the sensor and reduced to zero when those forces are lower than the threshold (see middle graph), whereas the angular velocities (bottom graph) remain close to zero since only small corrections are needed to keep the reference orientation of the flat object. Fig. 18 shows the trajectory followed by the robot tool, where it can be seen again that a large area has been covered by the robot during this experiment: around 0.5m in X-axis, 1.1m in Y-axis and 0.55m in Z-axis.

The experimental results shown above allow to validate the proposed control strategy and to appreciate its generality, effectiveness and main features. The method could be extrapolated to a specific industrial application using an experimental platform with adequate capabilities: robot payload; robot velocity; sensor accuracy, etc.

8.4. Discussion

A main advantage of SMC techniques over classical continuous approaches is its inherent *robustness* [9], see Section 5.4. In this regard the above experiments have revealed how the robot is able to automatically adapt the tool position to the operator's forces while maintaining the required tool orientation regardless of the possible inaccuracies accounted for at the low-level joint controllers and, in general, all the reasonable perturbations and unmodeled dynamics (non-linearities, friction forces, etc.) that could be expected in a practical setting for the collaborative task at hand, as shown by the results presented here (please also refer to the video links included).

Another typical advantage of SMC strategies over classical continuous approaches is the *low computational cost* [9], see Section 5.4. In particular, the proposed SMC algorithm only requires the Lie derivatives $\mathbf{L_g}\phi_i$, which are given by the robot Jacobian and the constraint functions computed from the roll-pitch-yaw orientation angles and the force sensor measurements. 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, e.g., 15 microseconds for the above experiments using a modern computer.

9. Conclusions

An approach for human-robot collaboration in transport applications has been developed in this work. In particular, some coordinates of the robot pose were controlled to achieve the reference values imposed by a handling task, whilst the remaining coordinates were guided by the human operator using a force sensor located at the robot tool. There are two main distinctive features of the proposal: the multi-layered nature of the strategy, where a set of four tasks are defined with different priorities; and an adaptive one-side sliding mode control developed to fulfill inequality constraints, which were defined to track both the reference values and the operator's forces.

The main advantages of the proposed approach are robustness and low computational cost. The feasibility and effectiveness of the proposed approach were substantiated by simulation and experimentation using a redundant 7R manipulator.

Acknowledgements

References

[1] Amini, M., Shahbakhti, M., Pan, S., and Hedrick, J. (2017). Bridging the gap between designed and implemented controllers via adaptive robust discrete sliding mode control. *Control Engineering Practice*, 59:1–15.

[2] Ansari, Y., Manti, M., Falotico, E., Mollard, Y., Cianchetti, M., and Laschi, C. (2017). Towards the development of a soft manipulator as an assistive robot for personal care of elderly people. *International Journal of Advanced Robotic Systems*, 14(2):DOI 10.1177/1729881416687132.

[3] Bartoszewicz, A. and Lesniewski, P. (2014). Reaching law approach to the sliding mode control of periodic review inventory systems. *IEEE Transactions on Automation Science and Engineering*, 11(3):810–817.

[4] Begnini, M., Bertol, D., and Martins, N. (2017). A robust adaptive fuzzy variable structure tracking control for the wheeled mobile robot:

Simulation and experimental results. *Control Engineering Practice*, 64:27–43.

[5] Bigelow, F. and Kalhor, A. (2017). Robust adaptive controller based on evolving linear model applied to a Ball-Handling mechanism. *Control Engineering Practice*, 69:85–98.

[6] Chen, X., Shen, W., Cao, Z., and Kapoor, A. (2014). A novel approach for state of charge estimation based on adaptive switching gain sliding mode observer in electric vehicles. *Journal of Power Sources*, 246(Supplement C):667 – 678.

[7] Chiaverini, S., Oriolo, G., and Walker, I. (2008). Kinematically redundant manipulators. *Springer Handbook of Robotics*, pages 245–268.

[8] Cong, B., Chen, Z., and Liu, X. (2014). On adaptive sliding mode control without switching gain overstimation. *International Journal of Robust and Nonlinear Control*, 24(3):515–531.

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

[10] Golub, G. and Van Loan, C. (1996). *Matrix Computations*. The Johns Hopkins University Press, Baltimore, MD, 3rd edition.

[11] Jiang, Z., Sun, Y., Lei, L., Hu, Y., Xiao, C., and Zhang, J. (2016). Compliance control based on particle swarm optimization approach for physical human-robot interaction. In *2016 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, pages 117–122.

[12] Jin, X., Zhu, S., Zhu, X., Chen, Q., and Zhang, X. (2017). Single-input adaptive fuzzy sliding mode control of the lower extremity exoskeleton based on human-robot interaction. *Advances in Mechanical Engineering*, 9(2):DOI 10.1177/1687814016686665.

[13] Jlassi, S., Tliba, S., and Chitour, Y. (2014). An event-controlled online trajectory generator based on the human-robot interaction force processing. *Industrial Robot: An International Journal*, 41(1):15–25.

[14] Kashiri, N., Tsagarakis, N. G., Van Damme, M., Vanderborght, B., and Caldwell, D. G. (2016). *Proxy-Based Sliding Mode Control of Compliant*

Joint Manipulators, pages 241–257. Springer International Publishing, Cham.

[15] Khan, A. M., Yun, D.-w., Zuhaib, K. M., Iqbal, J., Yan, R.-J., Khan, F., and Han, C. (2017). Estimation of desired motion intention and compliance control for upper limb assist exoskeleton. *International Journal of Control, Automation and Systems*, 15(2):802–814.

[16] Koustoumpardis, P. N., Chatzilygeroudis, K. I., Synodinos, A. I., and Aspragathos, N. A. (2016). *Human Robot Collaboration for Folding Fabrics Based on Force/RGB-D Feedback*, pages 235–243. Springer International Publishing, Cham.

[17] Kunusch, C., Puleston, P., and Mayosky, M. (2012). *Fundamentals of Sliding-Mode Control Design*, pages 35–71. Springer London, London.

[18] Leung, T.-P., Zhou, Q.-J., and Su, C.-Y. (1991). An adaptive variable structure model following control design for robot manipulators. *IEEE Transactions on Automatic Control*, 36(3):347–353.

[19] Li, Y. and Ge, S. S. (2016). Force tracking control for motion synchronization in human-robot collaboration. *Robotica*, 34(6):1260–1281.

[20] Lu, L. and Wen, J. T. (2015). Human-directed robot motion/force control for contact tasks in unstructured environments. In *2015 IEEE International Conference on Automation Science and Engineering (CASE)*, pages 1165–1170.

[21] Lu, Y. S. (2009). Sliding-mode disturbance observer with switching-gain adaptation and its application to optical disk drives. *IEEE Transactions on Industrial Electronics*, 56(9):3743–3750.

[22] Maurtua, I., Ibarguren, A., Kildal, J., Susperregi, L., and Sierra, B. (2017). Human-robot collaboration in industrial applications. *International Journal of Advanced Robotic Systems*, 14(4):DOI 10.1177/1729881417716010.

[23] Meziane, R., Otis, M. J.-D., and Ezzaidi, H. (2017). Human-robot collaboration while sharing production activities in dynamic environment: Spader system. *Robotics and Computer-Integrated Manufacturing*, 48:243–253.

[24] Mitra, A. and Behera, L. (2015). Development of a fuzzy sliding mode controller with adaptive tuning technique for a MRI guided robot in the human vasculature. In *2015 IEEE 13th International Conference on Industrial Informatics (INDIN)*, pages 370–377.

[25] Mohammed, A., Schmidt, B., and Wang, L. (2017). Active collision avoidance for human-robot collaboration driven by vision sensors. *International Journal of Computer Integrated Manufacturing*, 30(9):970–980.

[26] Monsees, G. and Scherpen, J. (2002). Adaptive switching gain for a discrete-time sliding mode controller. *International Journal of Control*, 75(4):242–251.

[27] Nakamura, Y., Hanafusa, H., and Yoshikawa, T. (1987). Task-priority based redundancy control of robot manipulators. *The Int. Journal of Robotics Research*, 6(2):3–15.

[28] Nikoleizig, S., Vick, A., and Krüger, J. (2017). Compensating human feedback oscillation in compliance control for industrial robots. In *2017 3rd International Conference on Control, Automation and Robotics (ICCAR)*, pages 221–224.

[29] Plestan, F., Shtessel, Y., Bregeault, V., and Poznyak, A. (2010). New methodologies for adaptive sliding mode control. *International Journal of Control*, 83(9):1907–1919.

[30] Rahman, N. and Lee, M. (2013a). Roboticslab based simulation for extraction of reaction force in multi DOF surgical robot dynamic system and analysis. In *2013 10th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, pages 560–564.

[31] Rahman, N. and Lee, M. C. (2013b). Reaction force separation method of surgical tool from unknown dynamics and disturbances by fuzzy logic and perturbation observer of SMCSPO algorithm. In *The SICE Annual Conference 2013*, pages 2536–2541.

[32] Rashid Husain, A., Noh Ahmad, M., and Halim Mohd Yatim, A. (2008). Chattering-free sliding mode control for an active magnetic bearing system. volume 59, pages 385–391.

[33] Shen, Q., Wang, D., Zhu, S., and Poh, E. K. (2015). Integral-type sliding mode fault-tolerant control for attitude stabilization of spacecraft. *IEEE Transactions on Control Systems Technology*, 23(3):1131–1138.

[34] Shtessel, Y., Edwards, C., Fridman, L., and Levant, A. (2014). Introduction: Intuitive theory of sliding mode control. In *Sliding Mode Control and Observation*, pages 1–42. Springer.

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

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

[37] Taleb, M., Plestan, F., and Bououlid, B. (2015). An adaptive solution for robust control based on integral high-order sliding mode concept. *International Journal of Robust and Nonlinear Control*, 25(8):1201–1213.

[38] Tsarouchi, P., Matthaiakis, A.-S., Makris, S., and Chryssolouris, G. (2017). On a human-robot collaboration in an assembly cell. *International Journal of Computer Integrated Manufacturing*, 30(6):580–589.

[39] Utkin, V. (2016). Discussion aspects of high-order sliding mode control. *IEEE Transactions on Automatic Control*, 61(3):829–833.

[40] Utkin, V., Guldner, J., and Shi, J. (2009). *Sliding Mode Control in Electro-Mechanical Systems*. Taylor & Francis, London, 2nd edition.

[41] Vogel, J., Haddadin, S., Jarosiewicz, B., Simeral, J., Bacher, D., Hochberg, L., Donoghue, J., and van der Smagt, P. (2015). An assistive decision-and-control architecture for force-sensitive hand-arm systems driven by human-machine interfaces. *The International Journal of Robotics Research*, 34(6):763–780.

[42] Wang, C., Li, Y., Ge, S. S., Tee, K. P., and Lee, T. H. (2013). Continuous critic learning for robot control in physical human-robot interaction. In *2013 13th International Conference on Control, Automation and Systems (ICCAS 2013)*, pages 833–838.

[43] Wheeler, G., Su, C., and Stepanenko, Y. (1998). A sliding mode controller with improved adaption laws for the upper bounds on the norm of uncertainties. *Automatica*, 34(12):1657–1661.

[44] Wu, Q., Wang, X., Du, F., and Zhu, Q. (2015). Fuzzy sliding mode control of an upper limb exoskeleton for robot-assisted rehabilitation. In *2015 IEEE International Symposium on Medical Measurements and Applications (MeMeA) Proceedings*, pages 451–456.

[45] Xu, Q. (2013). Adaptive discrete-time sliding mode impedance control of a piezoelectric microgripper. *IEEE Transactions on Robotics*, 29(3):663–673.

[46] Yilmaz, C. and Hurmuzlu, Y. (2000). Eliminating the reaching phase from variable structure control. *ASME J Dyn Syst, Meas, Control*, 122(04):753–757.

[47] Yu, X. and Kaynak, O. (2009). Sliding-mode control with soft computing: A survey. *IEEE Transactions on Industrial Electronics*, 56(9):3275–3285.

[48] Yun, D., Khan, A. M., Yan, R.-J., Ji, Y., Jang, H., Iqbal, J., Zuhaib, K. M., Ahn, J. Y., Han, J., and Han, C. (2016). Handling subject arm uncertainties for upper limb rehabilitation robot using robust sliding mode control. *International Journal of Precision Engineering and Manufacturing*, 17(3):355–362.

[49] Zhou, J., Zhou, Z., and Ai, Q. (2016). Impedance control of the rehabilitation robot based on sliding mode control. In Li, X., editor, *Mechanical Engineering and Control Systems (MECS2015)*, pages 135–140.

[50] Zhu, J. and Khayati, K. (2017). On a new adaptive sliding mode control for MIMO nonlinear systems with uncertainties of unknown bounds. *International Journal of Robust and Nonlinear Control*, 27(6):942–962.