Bimanual robot control for surface treatment tasks

Alberto García[a], J. Ernesto Solanes[a*], Luis Gracia[a], Pau Muñoz-Benavent[a], Vicent Girbés-Juan[b] and Josep Tornero[a]

[a]Instituto de Diseño y Fabricación, Universitat Politècnica de València, Camí de Vera s/n, 46022, València, Spain
[b]Departament d'Enginyeria Electrònica, Universitat de València, Avda de la Universitat s/n, 46100 Burjassot, Spain

ARTICLE HISTORY
Compiled May 5, 2021

ABSTRACT
This work develops a method to perform surface treatment tasks using a bimanual robotic system, i.e., two robot arms cooperatively performing the task. In particular, one robot arm holds the workpiece while the other robot arm has the treatment tool attached to its end-effector. Moreover, the human user teleoperates all the six coordinates of the former robot arm and two coordinates of the latter robot arm, i.e., the teleoperator can move the treatment tool on the plane given by the workpiece surface. Furthermore, a force sensor attached to the treatment tool is used to automatically attain the desired pressure between the tool and the workpiece and to automatically keep the tool orientation orthogonal to the workpiece surface. In addition, to assist the human user during the teleoperation, several constraints are defined for both robot arms in order to avoid exceeding the allowed workpsace, e.g., to avoid collisions with other objects in the environment. The theory used in this work to develop the bimanual robot control relies on sliding mode control and task prioritization. Finally, the feasibility and effectiveness of the method are shown through experimental results using two robot arms.

KEYWORDS
Bimanual Robotic System, Robust Control, Robotics

1. Introduction

1.1. Motivation

Automation is a growing tendency in all aspects of society, since it increases the speed, precision and efficiency with which tasks are solved, while also relieving human operators from hard, dangerous or repetitive aspects of said tasks.

Nonetheless, many operations cannot be fully automated yet, as they require the abilities of a human operator. In these cases, Human-Robot Interaction (HRI) appears as a step forward, trying to combine the accuracy and strength of a robotic system and the adaptability of a human operator.

A common situation where HRI is required takes place when an application relies totally or partially on the abilities of a human operator, but the humans presence in

*Corresponding author J.E. Solanes. Email: juasogal@isa.upv.es

the workspace or his or her direct intervention in the task is problematic, for instance, in hazardous environments or high precision operations.

In these situations, teleoperation permits different degrees of human implication in the application, while keeping a remote control architecture. In more general terms, robotic teleoperation is interesting for work environments, as it eliminates unnecessary human presence, thus increasing safety and also efficiency in regards to space requirements.

And regarding robotic systems, bimanual solutions, i.e., the collaboration of two robotic manipulators working on the same workpiece to accomplish a shared goal, are of deep interest, due to their higher versatility in comparison with single robot arm systems and the possibility of emulating human-like manual operations.

Thus, this contribution develops a control architecture for a bimanual robotic system in order to conduct a surface treatment task in which the human user partially teleoperates both robot arms. Note that the target task is complex and mainly non-automated (Kieselbach, Nöthen, & Heuer, 2019).

1.2. Literature review

1.2.1. Assisted teleoperation in robotics

Teleoperation, i.e. the remote control of a robot by a human operator, is one of the first manifestations of robotics (Niemeyer, Preusche, Stramigioli, & Lee, 2016) and is still an ongoing tendency in research. Its fields of interest are wide, including situations where the environment is either unsafe for humans, such as operations in space (Chen, Huang, & Liu, 2019) and in radioactive areas (Abi-Farraj, Pacchierotti, Arenz, Neumann, & Giordano, 2020; Bandala, West, Monk, Montazeri, & Taylor, 2019), or difficult to reach, such as aerial (Isop et al., 2019; Suarez et al., 2020) and subaquatic (Brantner & Khatib, 2021; Sivev, Coleman, Omerdi, Dooly, & Toal, 2018) tasks or rescue operations (Kono, Mori, Ji, Fujii, & Suzuki, 2019); but also situations where a non-invasive approach is required, mainly surgery (Y. Chen et al., 2020; Kapoor, Li, & Taylor, 2005; Saracino, Oude-Vrielink, Menciassi, Sinibaldi, & Mylonas, 2020; Yoon, Jeong, & Yi, 2018).

Moreover, even in cases of automated tasks with sophisticated AI, autonomous systems can be improved by working together with a human operator, that is, introducing shared-control architectures (Johnson & Vera, 2019). That being the case, a rich body of contributions in the field of teleoperation, of which this work is part of, has been developing, with its focus put on deepening and improving human-robot interaction in teleoperation (Clark et al., 2019; Girbés-Juan, Schettino, Demiris, & Tornero, 2021; Gorjup, Dwivedi, Elangovan, & Liarokapis, 2019; Lu, Huang, & Liu, 2018; Nicolis, Palumbo, Zanchettin, & Rocco, 2018; Selvaggio, Abi-Farraj, Pacchierotti, Giordano, & Siciliano, 2018), rather than solving specific problems such as the ones cited above.

Regarding the degree of shared-control between the human operator and the robotic system, two extremes of a spectre might be identified (Niemeyer et al., 2016), one being direct control, where the robotic systems limits itself to the direct execution of the humans commands, and the other being supervisory control, where the robotic system carries out highly automated tasks, and the human operator makes high-level decisions. Some examples can be found of these two extremes, for instance (Liu et al., 2019), a case of direct control where the contribution is focused on the processing of neural signals to teleoperate a dual arm system; (Bandala et al., 2019), where both cases are present: first, the direct control of the mobile platform of the robot and then,

the supervisory control of a dual-arm manipulator system, where the human operator selects points for the robotic system to grasp and cut automatically, and indicates the moment when the automated task should start; or (Isop et al., 2019), where the aerial exploration of indoor environments is carried out using a supervisory control approach, so the teleoperator chooses whether the aerial robot should inspect an object, explore an area or go to the next area, whereas the control algorithm manages every subtask making the task possible.

Still in the field of direct control, but with a more assisted approach of teleoperation, telepresence (Niemeyer et al., 2016) provides the human operator with an interface which makes the direct control task less dependent on his or her skills and concentration. Telepresence is a strong trend in recent research developments, with the introduction of virtual or augmented reality (Gorjup et al., 2019; Solanes et al., 2020), visual interfaces (Yoon et al., 2018) and haptic devices (Selvaggio, Ghalamzan, Moccia, Ficuciello, & Siciliano, 2019) or the combination of these different elements (Clark et al., 2019; Girbés-Juan et al., 2021; Saracino et al., 2020) to direct control teleoperation. A particular case can be found in (Nicolis et al., 2018), where one arm of a bimanual robot is teleoperated to grasp a target object, while the other develops an automatic task of visual-servoing to keep the object in sight of a camera and avoiding occlusions, thus making the teleoperation easier.

Since telepresence relies heavily on the users skills, a whole body of research focuses on imposing restrictions to the position references the human operator can command, by incorporating virtual barriers, such as Virtual Fixtures (Y. Chen et al., 2020) in surgery, an example of which can be found in (Kapoor et al., 2005), where the users position references are automatically modified to fit an allowed area, and Haptic Guidance applications, where a haptic system avoids the user from commanding reference positions beyond certain limits, as can be seen in (Abi-Farraj et al., 2020; Selvaggio et al., 2019).

Although this is a necessary approach in some fields, especially in surgery, where non-vision-based reactive control methods (such as force and impedance control) are hard to combine with a non invasive approach (small sensors, soft tissues) (Lopez, Zollo, & Guglielmelli, 2013), assisted teleoperation with telepresence interfaces and virtual barriers present inconveniences due to the fact that while control is still mainly carried out by the human operator, his or her maneuvers are limited by passive restrictions which are not evident in a perceptual sense, (Selvaggio et al., 2018).

As a result, shared-control architectures where the robotic system plays an active role in the task, while preserving the initiative of the human operator, are being addressed in various contributions. Whereas some of them are based on trajectory planners, such as (Lv et al., 2020), this offers some limitations regarding computational cost and real-time control in comparison to reactive control methods, such as the case of this work, among others (Selvaggio et al., 2018; Suarez et al., 2020).

In (Selvaggio et al., 2018), a system composed of two robot manipulators sharing the same workspace is developed, where one of the robot manipulators carries out an automatic visual task, while the other is teleoperated with a shared-control approach, in order to grasp a target object. The teleoperation is assisted by the control algorithm in the two defined situations: when the gripper is far from the target object, the control algorithm controls the gripper orientation in order to avoid the system constraints (joint limits, singularities and collisions with the other robot) and leaves the human user in control of the gripper translation, although this is also limited by a conventional haptic guidance which helps the user to avoid the constraints; whereas when the gripper is close to the target object, the control algorithm manages all the

necessary Degrees of Freedom (DoF) to make sure that the gripper always points towards the object, and so the teleoperator can use the remaining DoF to move the gripper in a sphere centered around the target object. The present work shares part of the approach of (Selvaggio et al., 2018), with an assisted teleoperation where the control algorithm manages different DoF depending on the evolution of certain parameters so as to ensure that necessary restrictions are met. However, in (Selvaggio et al., 2018) part of the shared-control relies on conventional haptic guidance, whereas the present work ensures the accomplishment of all the restrictions with a multitask control architecture with different priority levels.

In (Suarez et al., 2020), an aerial manipulation application is developed and tested. In this case, the human user is expected to carry out the teleoperation of two compliant manipulators attached to the aerial robot system, while the control application manages the variables of the flight using nested PID controllers and also applies a compliant joint and force control to the manipulators.

Although the present work shares with (Suarez et al., 2020) combining a reactive control approach with the dexterity and flexibility provided by a human teleoperator, there is a significant difference, for the present work assists the teleoperation by directly limiting the DoF available for the teleoperation of the bimanual robotic system and using those DoF to ensure certain constraints, while (Suarez et al., 2020) assists the teleoperation mainly by making sure that the conditions necessary for the teleoperation are met, but the human user exerts an almost completely direct teleoperation over the manipulators.

Finally, in (Brantner & Khatib, 2021) an assisted subaquatic teleoperation of a humanoid exploration robot with a bimanual manipulator system is developed. Similarly to the present work, a multitask control architecture with different priority levels is implemented, so a lower priority control law can use just the DoF that higher priority control laws are not using. In (Brantner & Khatib, 2021), this architecture is developed in order to make sure that certain restrictions are met (in this case, joint limits, self collision and obstacle avoidance) while the human operator teleoperates the robot arms to perform the manipulation task, and, in the lowest priority level, the robot body and arms postures are automatically controlled in order to optimize the position of the body with respect to the hands and to optimize the inertia of the robot hands by adjusting the position of the arms. However, (Brantner & Khatib, 2021) mainly relies on conventional continuos control laws, whereas the present work utilizes advanced Sliding Mode Control (SMC) techniques.

1.2.2. Bimanual robotics

The introduction of dual-arm robotic systems in industrial, domestic and assistive tasks is justified by their dexterity, flexibility, manipulability and their general resemblance to human behaviour when it comes to solving tasks, which makes it easier for human operators to relate to them and to design human-like applications more intuitively (Smith et al., 2012). As a result of sustained remarkable interest in dual-arm robotic systems from industry and academia (Makris et al., 2017), a wide variety of contributions in this field is available.

Some of these contributions can be described as goal-coordinated dual-arm robotic applications, where two robot manipulators work on the same task without physical interaction between them (Smith et al., 2012): for instance, in (Nicolis et al., 2018; Selvaggio et al., 2018) similar manipulation applications are presented, where one robotic arm is teleoperated to manipulate a workpiece while the other performs a fully auto-

mated visual servoing algorithm to keep track of the workpiece while avoiding collisions and other constraints, with no physical interaction between the two robot manipulators; and in (Sepúlveda, Fernández, Navas, Armada, & González-De-Santos, 2020), where two robots pick different aubergines simultaneously, and, when it is necessary, one arm helps the other by removing obstacles for the picking task.

However, a main trend in dual-arm robotics, including this work, focuses on bimanual robotics, which consists of dual-arm robotic systems where the two robot manipulators are coordinated to achieve a shared goal and interact physically in order to do so (Smith et al., 2012). Thus, bimanual robotics find a wide range of applications, such as: manipulation of deformable objects (Chatzilygeroudis et al., 2020; Garcia-Camacho et al., 2020; Sintov, Macenski, Borum, & Bretl, 2020), objects with unknown shape (Clark et al., 2019; Mitash, Shome, Wen, Boularias, & Bekris, 2020) or objects the geometry of which requires two grasping points (Salehian, Figueroa, & Billard, 2018; Suarez et al., 2020; Wu et al., 2019), emulation of human bimanual tasks (García, Rosell, & Suárez, 2019; Qu, Zhang, Wang, & Fu, 2019; Rakita, Mutlu, Gleicher, & Hiatt, 2019), assistive robotics (Joshi, Tarapure, & Shibata, 2020; Li, Guo, & Mukai, 2019), assembly operations (H. Chen, Li, Wan, Huang, & Harada, 2020; Parigi Polverini, Zanchettin, & Rocco, 2019; Zimmermann, Hakimifard, Zamora, Poranne, & Coros, 2020), surgery tasks (Zhong, Wang, Wang, & Liu, 2019) and simultaneous manipulation and cutting (Bandala et al., 2019), manipulation and fastening (Makris et al., 2017) or manipulation and surface treatment (Girbés-Juan et al., 2021), which is the case of this work.

A significant number of bimanual robotics contributions develop fully automated applications, presenting diverse approaches, such as offline training and machine learning (García et al., 2019; Joshi et al., 2020), solutions based on motion planning (Liang, Xu, Zhou, Li, & Ye, 2020; McConachie, Dobson, Ruan, & Berenson, 2020; Mitash et al., 2020; Sintov et al., 2020; Zimmermann et al., 2020), sometimes including specific situations where a reactive control method (for instance, force control or impedance control) is activated in order to improve the motion planner performance (H. Chen et al., 2020; Parigi Polverini et al., 2019; Zhong et al., 2019) and applications which rely mainly on closed-loop control methods (Li et al., 2019; Wu et al., 2019).

Nonetheless, Human-Robot Interaction is specially interesting for bimanual robotics not only because of the general reasons previously explained, but also because it exploits the humans intuitive understanding of bimanual configurations and movements, and, consequently so, it offers a fruitful framework for HRI. All these general and specific factors make HRI a main trend in bimanual robotics, trend to which this work belongs.

Of course, there exist contributions which include some degree of HRI and bimanual robotic configurations where the human does not interact with the bimanual task in itself, but rather with the robotic system as a whole. This is the case of (Makris et al., 2017), where the HRI consists of slowing down or stopping the operation of the bimanual robotic system in order to allow the presence of the human in the same workspace; or the supervisory control approaches involving bimanual tasks cited in Section 1.2.1.

However, the framework of the present work is that of Human-Robot Interaction involving a bimanual operation.

In this same framework, (Ibarguren, Eimontaite, Outn, & Fletcher, 2020) proposes an operation typically performed by two human operators, the transportation of a large workpiece, where one of the operators is substituted by a bimanual robotic system mounted on a mobile platform. To carry out the operation, the workpiece

can be moved along a pre-determined trajectory, and the human operator retains the initiative to start and stop the movement as well as to decide arbitrary adaptations of this trajectory, thanks to a conventional impedance control. Furthermore, certain limits for the arbitrary deformations of the trajectory are pre-determined too.

Thus, in a similar way to the present contribution, in (Ibarguren et al., 2020) the Human Robot Interaction benefits from the relatability and flexibility of a bimanual robotic configuration, and also the human operator retains the initiative for the main aspects of the operation, while being guided to perform it correctly, thanks to a reactive control. However, differently to (Ibarguren et al., 2020), in this contribution the human user (the teleoperator), can command totally arbitrary movements to both robot arms, the constraints are respected thanks to an advanced control architecture and there is a higher degree of coordination between the robot arms, since they are in close contact during the surface treatment operation.

On the other hand, a more similar related work can be found in (Brantner & Khatib, 2021). Although this contribution has already been addressed above in Section 1.2.1, there are some similarities to the present work regarding the bimanual task in itself which are worth recalling. Besides the existence of several layers of control with different levels of priority, the bimanual task is also subject of shared-control: a position control keeps the robot hands orientation constant and a force control keeps contact with the manipulated surface, thus leaving the teleoperator with 2 DoF to command, i.e., movement on a plane. This approach is similar to the one taken in the present contribution for one of the robot manipulators, as it will be explained in Section4.

1.2.3. Task optimization

In order to accomplish the robot task, some kind of optimization has to be considered when developing the control algorithm for the robot system. These optimization problems are well known in robotics and have been typically approached in two different ways, broadly classified as planning methods and "reactive" controllers (Siciliano & Khatib, 2008), as discussed below.

On the one hand, robot controllers based on high-level planning mainly solve the optimization problem taking into account the complete data of the robot task. This type of approach is suitable to cope with trap situations (Gracia, Sala, & Garelli, 2012) and singular configurations (Gracia, Andres, & Tornero, 2009), but it typically suffers from high computational cost and the difficulty to deal with task uncertainty. Examples of this type of approach can be found in (Da Silva, Dos Santos, Fernandes, Vilas Boas, & Garcia, 2020; Li, Li, Li, & Cao, 2019).

On the other hand, robot controllers based on reactive algorithms solve the optimization problem without prediction capabilities, i.e., considering only the data associated to the current time instant in order to compute the current control action to be applied to the robot system. This type of approach can be readily used in real-time to control the robot system, although it may suffer from trap situations and singular configurations. Examples of this type of approach are the navigation algorithms based on the well known artificial potential-fields (Li & Li, 2020; Park, Lee, & Kim, 2020) or, more recently, based on neural networks (Khan, Li, Chen, & Liao, 2020) and SMC (Fei, Shi, & Lim, 2020; Khan & Li, 2020), among others.

The method developed in this work to control the bimanual robotic system belongs to this category of reactive algorithms. In particular, several controllers based on SMC theory (Su & Zheng, 2020; Zhang, Ma, Luo, & Liu, 2020) are developed in this work in order to benefit from its inherent advantages, such as robustness and

low computational cost. Moreover, a task prioritization strategy is also used in this work to simultaneously address a set of objectives in order to properly perform the surface treatment task. Thus, the objectives are associated to a set of equalities whose square errors are hierarchically minimized according to the priority assigned to each objective/equality.

1.3. Proposed approach

This work develops a method to perform surface treatment tasks using a bimanual robotic system, i.e., two robot arms cooperatively performing the task. In particular, one robot arm, namely Workpiece Robot (WR), holds the workpiece while the other robot arm, namely Surface Treatment Robot (STR), has the treatment tool attached to its end-effector.

In this way, some robot coordinates are teleoperated by the human user, while the remaining robot coordinates are automatically controlled. In particular, the user teleoperates all the six coordinates of the WR in order to put the workpiece in a proper position and orientation for the task. Moreover, the teleoperator commands two linear coordinates of the STR to move the treatment tool on the workpiece surface in order to apply the surface treatment. Furthermore, a Force/Torque (F/T) sensor attached to the STR end-effector is used to automatically adapt the STR tool in order to attain the desired pressure between the tool and the workpiece as well as to keep the tool orientation orthogonal to the workpiece surface.

In addition, to assist the human user during the teleoperation, several constraints are defined for both robot arms in order to avoid exceeding the allowed workpsace, e.g., to avoid collisions with other objects in the environment.

As mentioned above, the theory used in this work to develop the bimanual robot control relies on SMC as well as task prioritization.

The content of the article is as follows. Section 2 and Section 3 present the theoretical basis used in this work. Subsequently, Section 4 develops the method proposed in this work for the bimanual robot control in order to perform surface treatment tasks, whose implementation is detailed in Section 5. The performance and effectiveness of the method are shown in Section 6 through experimental results, whereas some conclusions are outlined in Section 7.

2. Preliminaries

Kinematics. The kinematics of a robot system can be expressed as:

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

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

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

being $\mathbf{p} = \begin{bmatrix} x & y & z & \alpha & \beta & \gamma \end{bmatrix}^{\mathrm{T}}$ the pose of the robotic system, where α, β and γ represent the orientation angles (roll, pitch and yaw, respectively), $\mathbf{q} = \begin{bmatrix} q_1 & \cdots & q_n \end{bmatrix}^{\mathrm{T}}$ the configuration of the robotic system, $\mathbf{J}$ the Jacobian matrix and $\mathbf{l}$ the so-called kinematic function (Chiaverini, Oriolo, & Walker, 2008).

Task-priority based strategy. This method is useful to address a set of tasks with different priorities, where the error of the task equations has to be minimized. The recursive equations of this strategy are given below (Nakamura, Hanafusa, & Yoshikawa, 1987):

$$\mathbf{A}_i\overline{\mathbf{x}} = \mathbf{b}_i, \qquad i = 1, \ldots, M, \tag{4}$$

$$\overline{\mathbf{x}}_i = \overline{\mathbf{x}}_{i-1} + (\mathbf{A}_i\mathbf{N}_{i-1})^{\dagger}(\mathbf{b}_i - \mathbf{A}_i\overline{\mathbf{x}}_{i-1}), \qquad i = 1, \ldots, M, \tag{5}$$

$$\mathbf{N}_i = \mathbf{N}_{i-1}(\mathbf{I} - (\mathbf{A}_i\mathbf{N}_{i-1})^{\dagger}(\mathbf{A}_i\mathbf{N}_{i-1})), \qquad i = 1, \ldots, M, \tag{6}$$

being M the considered number of equalites or tasks, $\overline{\mathbf{x}}$ is the unknown vector to be computed (which, in this work, corresponds to the commanded joint accelerations $\ddot{\mathbf{q}}_c$), $\mathbf{A}_i$ and $\mathbf{b}_i$ are the matrix and vector, respectively, for the i-th task ($i = 1$ represents the highest priority), $\overline{\mathbf{x}}_i$ is the solution that hierarchically minimizes the error of the first i task equations, $\mathbf{N}_0 = \mathbf{I}$ is the identity matrix, $\overline{\mathbf{x}}_0 = \mathbf{0}$ is the null vector and superscript † denotes the pseudoinverse of a matrix (Golub & Van Loan, 1996) (a threshold can be used to neglect the small singular values).

3. Sliding Mode Control

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

3.1. Conventional SMC to satisfy equality constraints

For conventional SMC, see Fig. 1-left, the state space of the system is divided into two regions, **A** and **B**, separated by the sliding surface. When the system is in **A** the control action $\mathbf{u} = \mathbf{u}_B$ "pushes" the system into **B**. Similarly, the control action $\mathbf{u} = \mathbf{u}_A$ "pushes" the system into **A** when the system is in **B**. Thus, in both cases the system evolves to the sliding surface, which is called *reaching mode* (Utkin, Guldner, & Shi, 2009). Then, the control action $\mathbf{u}$ switches between $\mathbf{u}_A$ and $\mathbf{u}_B$ at a theoretically infinite frequency to keep the system on the sliding surface, which is known as *sliding mode* (SM) (Utkin et al., 2009). Moreover, there is a *continuous equivalent control* (Edwards & Spurgeon, 1998) for the SM phase that keeps the system on the sliding surface, although SMC produces such control action without explicitly computing it and with low computational cost (Edwards & Spurgeon, 1998).

A conventional SMC is developed in the theorem below to fulfill equality constraints.

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

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{d}) + \mathbf{g}(\mathbf{x})\,\mathbf{u}, \tag{7}$$

where $\mathbf{x}(t)$ is the state vector, $\mathbf{d}(t)$ is an unmeasured disturbance or model uncertainty, $\mathbf{u}(t)$ is the control input vector (possibly discontinuous), $\mathbf{f}$ is a vector field and $\mathbf{g}$ is a set of vector fields.

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

$$\Phi_{eq} = \{\mathbf{x} \mid \phi_{eq,i}(\mathbf{x}) = 0\}, \tag{8}$$

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

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

$$\mathbf{L_g}\boldsymbol{\phi}_{eq}\mathbf{u} = -\mathbf{W}_{eq} sign(\boldsymbol{\phi}_{eq})\, u_{eq}^{+} \tag{9}$$

$$u_{eq}^{+} > \|L_f\boldsymbol{\phi}_{eq}\|_1 / diag_{\min}(\mathbf{W}_{eq}), \tag{10}$$

where column vector $\boldsymbol{\phi}_{eq}$ *is composed of all the functions* $\phi_{eq,i}$, *the scalar* $L_f\phi_{eq,i} = \frac{\partial \phi_{eq,i}^T}{\partial \mathbf{x}}\mathbf{f}$ *and the row vector* $\mathbf{L_g}\phi_{eq,i} = \frac{\partial \phi_{eq,i}^T}{\partial \mathbf{x}}\mathbf{g}$ *represent the Lie derivatives of* $\phi_{eq,i}(\mathbf{x})$ *in the direction of* $\mathbf{f}$ *and* $\mathbf{g}$, *respectively,* $L_f\boldsymbol{\phi}_{eq}$ *is a column vector composed of the elements* $L_f\phi_{eq,i}$ *of all equality constraints,* $\mathbf{L_g}\boldsymbol{\phi}_{eq}$ *is a matrix composed of the row vectors* $\mathbf{L_g}\phi_{eq,i}$ *of all equality constraints, sign*$(\cdot)$ *denotes the* sign function, u_{eq}^{+} *is a positive scalar representing the switching gain, diagonal matrix* $\mathbf{W}_{eq}$ *is composed of the constraint switching gain weights,* $\|\cdot\|_1$ *denotes the Taxicab or 1-norm and* $diag_{\min}(\cdot)$ *returns the minimum value of the diagonal elements of a matrix.*

Proof. The proof is similar to the generalization of Proof 2.1 in (Utkin et al., 2009). See (Utkin et al., 2009) for further details. □

3.2. One-side SMC to satisfy inequality constraints

This work proposes the one-side SMC depicted in Fig. 1-right, which is useful to satisfy inequalities. For this approach, the system state space is divided into the allowed region $\mathbf{B}$ and non-allowed region $\mathbf{A}$, which are separated by the constraint boundary. As before, the control action $\mathbf{u} = \mathbf{u}_B$ pushes the system into $\mathbf{B}$ when the system sate is in $\mathbf{A}$. Nevertheless, no control action is applied ($\mathbf{u} = \mathbf{0}$) when the system state is in $\mathbf{B}$. Thus, the system evolves in reaching mode to the sliding surface when it starts in $\mathbf{A}$, although the system state can "freely" evolve according to some other criterion when it starts in $\mathbf{B}$. Hence, only when the state trajectory tries by itself to leave the allowed region, the one-side SMC will make $\mathbf{u}$ switch between $\mathbf{0}$ and $\mathbf{u}_B$ at a theoretically infinite frequency, which can be seen as an ideal SM behaviour (Edwards & Spurgeon, 1998).

Using this approach, the theorem below is developed to fulfill inequality constraints.

Theorem 3.2. *Consider the dynamical system given by* (7) *and consider also that the system state vector* $\mathbf{x}$ *is subject to* inequality *constraints* $\phi_{in,i}(\mathbf{x}) \leq 0$, $i = 1, \ldots, N_{in}$, *where* $\phi_{in,i}(\mathbf{x})$ *is the ith inequality constraint function. Thus, the region* Φ_{in} *of the state space compatible with the constraints on state* $\mathbf{x}$ *is given by:*

$$\Phi_{in} = \{\mathbf{x} \mid \phi_{in,i}(\mathbf{x}) \leq 0\}, \tag{11}$$

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

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

$$v2dm\left(pos\left(\boldsymbol{\phi}_{in}\right)\right)\mathbf{L_g}\boldsymbol{\phi}_{in}\mathbf{u} = -\mathbf{W}_{in}\; pos\left(\boldsymbol{\phi}_{in}\right)\; u_{in}^{+} \tag{12}$$

$$u_{in}^{+} > \sum_{i=1}^{n_a}\left(\max(L_f\phi_{in,i},0)\right)/diag_{\min}(\mathbf{W}_{in}), \tag{13}$$

where v2dm$(\cdot)$ *converts a vector into a diagonal matrix, pos*$(\cdot)$ *denotes the* positive function *(i.e., pos*(x) *is equal to* 0 *if* $x < 0$ *and equal to* 1 *if* $x > 0$*), function* $\max(a_1,\ldots,a_i)$ *returns the maximum value out of the arguments* $a1,\ldots,a_i$*, column vector* $\boldsymbol{\phi}_{in}$ *is composed of all the functions* $\phi_{in,i}$*,* $\mathbf{L_g}\boldsymbol{\phi}_{in}$ *is a matrix composed of the row vectors* $\mathbf{L_g}\phi_{in,i}$ *of all inequality constraints, the scalar* $L_f\phi_{in,i} = \frac{\partial\phi_{in,i}^T}{\partial\mathbf{x}}\mathbf{f}$ *and the row vector* $\mathbf{L_g}\phi_{in,i} = \frac{\partial\phi_{in,i}^T}{\partial\mathbf{x}}\mathbf{g}$ *represent the Lie derivatives of the inequality constraints in the direction of* $\mathbf{f}$ *and* $\mathbf{g}$*, respectively,* u_{in}^{+} *is a positive scalar representing the switching gain, diagonal matrix* $\mathbf{W}_{in}$ *is composed of the constraint switching gain weights and* n_a *denotes the number of active inequality constraints, i.e., those with* $\phi_{in,i} \geq 0$*.*

It is worth noting that v2dm(pos($\boldsymbol{\phi}_{in}$)) on the left-side of (12) yields the trivial scalar equation $0 = 0$ for the non-active inequality constraints (i.e., those with $\phi_{in,i} < 0$) and, thus, these constraints do not use system DoF.

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

Assuming[1] that $\boldsymbol{\phi}_{in}^{n_a}(0) > \mathbf{0}$, the objective of this proof is to show that convergence to point $\boldsymbol{\phi}_{in}^{n_a} = \mathbf{0}$ is achieved in finite time.

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

$$\dot{\boldsymbol{\phi}}_{in} = \frac{\partial\boldsymbol{\phi}^{\mathrm{T}}}{\partial\mathbf{x}}\mathbf{f}(\mathbf{x},\mathbf{d}) + \frac{\partial\boldsymbol{\phi}^{\mathrm{T}}}{\partial\mathbf{x}}\mathbf{g}(\mathbf{x})\,\mathbf{u} = L_f\boldsymbol{\phi}_{in} + \mathbf{L_g}\boldsymbol{\phi}_{in}\mathbf{u}. \tag{14}$$

Premultiplying (14) by v2dm (pos ($\boldsymbol{\phi}_{in}$)) and substituting (12) yields:

$$\mathrm{v2dm}(\mathbf{z}_{in})\dot{\boldsymbol{\phi}}_{in} = \mathrm{v2dm}(\mathbf{z}_{in})L_f\boldsymbol{\phi}_{in} - \mathbf{W}_{in}\,\mathbf{z}_{in}\,u_{in}^{+}, \tag{15}$$

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

Let $V_{in} = \mathbf{z}_{in}^{\mathrm{T}}\,\mathrm{v2dm}(\mathbf{z}_{in})\,\boldsymbol{\phi}_{in}$ be a Lyapunov function candidate. Vector $\boldsymbol{\phi}_{in}^{n_a}$ can be generically partitioned into two subvectors $\boldsymbol{\phi}_{in}^{n_a} = [\boldsymbol{\phi}_{in}^{b\,\mathrm{T}} \quad \boldsymbol{\phi}_{in}^{n_a-b\,\mathrm{T}}]^{\mathrm{T}}$, where SM occurs in the manifold given by $\boldsymbol{\phi}_{in}^{b} = \mathbf{0}$, whereas the components of vector $\boldsymbol{\phi}_{in}^{n_a-b}$ are greater than zero. Note that one of these two subvectors may be empty at a certain time. Taking into account that $\mathbf{z}_{in}^{n_a-b} = \mathbf{1}$ and $\mathbf{z}_{in}^{N_{in}-n_a} = \mathbf{0}$, the Lyapunov function

[1]Note that it has been assumed, without loss of generality, that the active inequality constraints do not initially fulfill the constraints, i.e., $\boldsymbol{\phi}_{in}^{n_a}(0) > \mathbf{0}$, since otherwise the controlled system simply remains at the starting point $\boldsymbol{\phi}_{in}^{n_a}(0) = \mathbf{0}$, which is the goal of the SMC. Therefore, $\boldsymbol{\phi}_{in}^{n_a}(0) > \mathbf{0}$ is the *general case* for the starting point and does not pose any requirement for the task of the robotic system.

derivative is given by:

$$\dot{V}_{in} = \frac{d}{dt}\left(\begin{bmatrix}\mathbf{z}_{in}^{b}\\ \mathbf{1}\\ \mathbf{0}\end{bmatrix}^{\mathrm{T}} \mathrm{v2dm}\left(\begin{bmatrix}\mathbf{z}_{in}^{b}\\ \mathbf{1}\\ \mathbf{0}\end{bmatrix}\right)\right)\begin{bmatrix}\mathbf{0}\\ \boldsymbol{\phi}_{in}^{n_a-b}\\ \boldsymbol{\phi}_{in}^{N_{in}-n_a}\end{bmatrix} + \mathbf{z}_{in}^{\mathrm{T}}\,\mathrm{v2dm}(\mathbf{z}_{in})\,\dot{\boldsymbol{\phi}}_{in}$$
$$= \mathbf{z}_{in}^{\mathrm{T}}\,\mathrm{v2dm}(\mathbf{z}_{in})\,\dot{\boldsymbol{\phi}}_{in}. \qquad (16)$$

Substituting (15) in (16) yields:

$$\dot{V}_{in} = \mathbf{z}_{in}^{\mathrm{T}}\,\mathrm{v2dm}(\mathbf{z}_{in})\,L_f\boldsymbol{\phi}_{in} - \mathbf{z}_{in}^{\mathrm{T}}\,\mathbf{W}_{in}\,\mathbf{z}_{in}\,u_{in}^{+}. \qquad (17)$$

Since $\mathbf{z}_{in}^{N_{in}-n_a} = \mathbf{0}$ and the components of vector $\mathbf{z}_{in}^{n_a}$ range from 0 to 1, the upper bound of the first term in (17) is given by $z_{in,i}^{n_a} = 1$ if $L_f\phi_{in,i}^{n_a} > 0$ and $z_{in,i}^{n_a} = 0$ if $L_f\phi_{in,i}^{n_a} < 0$, that is:

$$\mathbf{z}_{in}^{\mathrm{T}}\,\mathrm{v2dm}(\mathbf{z}_{in})\,L_f\boldsymbol{\phi}_{in} \leq \sum_{i=1}^{n_a}(\max(L_f\phi_{in,i}, 0)). \qquad (18)$$

Assuming[2] that $u_{in}^{+} > 0$, the second term in (17) is negative, since matrix $\mathbf{W}_{in}$ is positive definite, and its upper bound is given by:

$$-\mathbf{z}_{in}^{\mathrm{T}}\,\mathbf{W}_{in}\,\mathbf{z}_{in}\,u_{in}^{+} \leq -\mathrm{diag}_{\min}(\mathbf{W}_{in})\,\|\mathbf{z}_{in}\|_2^2\;u_{in}^{+}, \quad \text{where } \|\mathbf{z}_{in}\|_2 \geq 1\ \forall\,\boldsymbol{\phi}_{in} > \mathbf{0}, \qquad (19)$$

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

Taking into account (18) and (19), the Lyapunov function derivative has the following upper bound:

$$\dot{V}_{in} \leq \sum_{i=1}^{n_a}(\max(L_f\phi_{in,i}, 0)) - \mathrm{diag}_{\min}(\mathbf{W}_{in})\,u_{in}^{+}. \qquad (20)$$

Therefore, if u_{in}^{+} fulfills (13) the right-hand of Eq. (20) is negative and, hence, the derivative of the Lyapunov function (i.e., the left-hand of Eq. (20)) is negative. Thus, the Lyapunov function decreases at a finite rate, it vanishes and SM occurs in the intersection of the active inequality constraints. Note that this conclusion is analogous to that in the proof of conventional SMC in (Utkin et al., 2009). □

[2]Note that the switching gain u_{in}^{+} is a free design parameter that has been assumed to be positive in order to apply the control action of the SMC in the proper direction. This is analogous, for instance, to the case of the classical "proportional controller" where, for stability reasons, the correction gain is assumed to be positive in order to apply the corrective control action in the direction of the error signal.

3.3. Modified constraints

The original constraints $\sigma_{eq,i}$ and $\sigma_{in,i}$ are modified as follows to include also the velocity of the robotic system:

$$\phi_{eq,i} = \sigma_{eq,i} + K_{eq,i}\dot{\sigma}_{eq,i} = 0 \tag{21}$$

$$\phi_{in,i} = \sigma_{in,i} + K_{in,i}\dot{\sigma}_{in,i} \leq 0, \tag{22}$$

where design parameters $K_{eq,i}$ and $K_{in,i}$ establish the approaching speed to the equality constraint manifold and to the boundary of the inequality constraint, respectively.

4. Proposed approach

The goal of this work is to develop a robot control using the SMC in Section 3 so that two robot manipulators can be simultaneously teleoperated by a human user to conduct a surface treatment task. One of the robots, namely *workpiece robot* (WR), holds the workpiece while the other robot, namely *surface treatment robot* (STR), has the surface treatment tool and a F/T sensor attached to its end-effector, so the human operator can adapt the position and orientation of the workpiece and, at the same time, command the movement of the surface treatment tool over the workpiece surface.

In order to achieve this, some coordinates of both robot arms are controlled automatically:

– WR: The WR control keeps the workpiece center inside the allowed workspace whose boundary is given by a superellipsoid, which resembles a rectangular cuboid with smoothed corners. Moreover, for safety, the WR control limits the angular positions that the workpiece can achieve while the user teleoperates the WR. Additionally, in case that the WR is redundant (i.e., the WR has more than six DoF), the WR control uses the redundant DoF to "push" the WR to a home configuration for increased safety.

– STR: The STR control keeps the robot tool center inside the allowed area on the workpiece, whose boundary is given by a modified superellipse, which resembles a rectangle with smoothed corners. Moreover, the STR control maintains the desired pressure and orthogonality to the workpiece needed to perform the treatment task, using the data from an F/T sensor, coined as treatment sensor. The remaining DoF of the STR are available for the human user to teleoperate the STR on the surface of the workpiece.

In the equations below, subscripts "w" and "s" are used to denote WR and STR, respectively.

4.1. System tasks

A different control architecture is developed for each robot arm of the bimanual application proposed in this work.

On the one hand, the following four prioritized tasks are considered for the WR control:

W1) The first level (high-priority task) includes the *inequality constraints* that must

be satisfied to keep the workpiece center within the allowed region, whose boundary is given by a superellipsoid as mentioned above.

W2) The second level (medium-high-priority task) includes the *inequality constraints* required for the workpiece to keep its angular position within previously specified limits.

W3) The third level (medium-low-priority task), a hybrid SMC, ensures a reference tracking so the human user can teleoperate the WR.

W4) The fourth level (low-priority task), which only applies for the case of *redundant* robots, is used to keep the robot close to a *home* configuration.

The input to these tasks is the WR state $\{\mathbf{q}_w, \dot{\mathbf{q}}_w\}$ (and $\mathbf{p}_w$, which is obtained from the WR kinematic function) and the reference $\mathbf{p}_{w,ref}$ generated by the human teleoperator for the WR pose vector, whereas each task gives an acceleration equality whose square error must be minimized.

On the other hand, three tasks with the same approach of priority levels are considered for the STR control:

S1) The first level (high-priority task) includes the *inequality constraints* that must be satisfied to keep the surface treatment tool center in the allowed area on the workpiece surface, whose boundary is given by a modified superellipse as mentioned above.

S2) The second level (medium priority task) includes the *equality constraints* that must be satisfied at all times to properly perform the treatment on the workpiece with the STR tool. In particular, equality constraints are defined to exert the desired pressure between the STR tool and the workpiece being treated, and to keep the tool orientation orthogonal to the workpiece surface.

S3) The third level (low priority task), a hybrid SMC, ensures a reference tracking so the human user can teleoperate the STR.

The input to these tasks is the STR state $\{\mathbf{q}_s, \dot{\mathbf{q}}_s\}$ (and $\mathbf{p}_s$, which is obtained from the STR kinematic function), the force vector $\mathbf{F}$ with the measurements of the F/T sensor and the reference $\overline{\mathbf{p}}_{sb,ref}$ generated by the human teleoperator for the 2D position of the STR tool on the workpiece, whereas each task gives an acceleration equality whose square error must be minimized.

In particular, the acceleration equality for the first and second levels of the WR control and the first level of the STR control are obtained below using the one-side SMC presented in Section 3.2, whereas the acceleration equality for the second level of the STR control are obtained below using the conventional SMC presented in Section 3.1.

Fig. 2 depicts the overview block diagram of the proposed control for the WR and STR.

It is worth noting that the collaboration between the human user and the bimanual robotic system in the surface treatment task consists in the *teleoperation* of both robot arms by the user. That is, the teleoperator commands all the six coordinates of the WR in order to put the workpiece in a proper position and orientation for the task. Moreover, the teleoperator commands two linear coordinates of the STR to move the treatment tool on the workpiece surface in order to apply the surface treatment, while the STR control automatically maintains the desired pressure and perpendicularity to the surface using the data from an F/T sensor.

Furthermore, in order to assist the human user during the teleoperation, several constraints are defined for both robot arms in order to avoid exceeding the allowed workpsace. In particular, a boundary constraint is considered for the WR to confine

the workpiece center inside the allowed 3D area in order to avoid collisions with other objects in the environment, see Section 4.5.1 for further details. Moreover, a second constraint is considered for the WR to limit the angular positions that the workpiece can achieve while the user teleoperates the WR in order to avoid an excessive tilt of the workpiece, see Section 4.5.2 for further details. Finally, another boundary constraint is considered for the STR to confine the STR tool inside the allowed 2D area on the workpiece surface in order to avoid exceeding the workpiece limits, see Section 4.6.1 for further details.

Note that other similar constraints could also be considered for both robot arms, e.g., the typical robot constraints for the maximum joint ranges. Details omitted for brevity.

4.2. Lie derivatives

To use the SMC in Section 3, a dynamical system in the form of Eq. (7) is considered for both WR and STR with the state vector $\mathbf{x} = \begin{bmatrix}\mathbf{q}^{\mathrm{T}} & \dot{\mathbf{q}}^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}}$, the disturbance vector $\mathbf{d} = \mathbf{d}_c$, where $\mathbf{d}_c$ represents inaccuracies of the low-level controller developed by the robot manufacturer, and the input vector $\mathbf{u} = \ddot{\mathbf{q}}_c$. Thus, the state equation of the system, which is a double integrator, is given by:

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

and, hence, the Lie derivatives for the constraint function ϕ_i result in:

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

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

4.3. Boundary model

The first levels of the control algorithms of both robot arms include inequality constraints required to keep the center of the workpiece and the surface treatment tool within the allowed workspace, whose boundary has the shape of a superellipsoid and a superellipse, respectively.

The equation that defines a superellipsoid is given by:

$$\left|\frac{x}{W}\right|^m + \left|\frac{y}{H}\right|^m + \left|\frac{z}{M}\right|^m = 1, \tag{26}$$

where $\{W, H, M\}$ are the superellipsoid axes and exponent m, whose value has to be greater than one, defines the smoothing of the superellipsoid. That is, the shape of the superellipsoid ranges from an ellipsoid to a rectangular cuboid as m ranges from 2 to infinity. In this work, the chosen value for m is 4.

In the case of a superellipse, the third dimension is removed, while maintaining the same exponent m. Therefore, the equation for the superellipse, which resembles a $2W \times 2H$ rectangle with smoothed corners, is given by:

$$\left|\frac{x}{W}\right|^m + \left|\frac{y}{H}\right|^m = 1. \tag{27}$$

However, in this work a modified superellipse is used instead of the one described above in order to obtain a more homogeneous control action to keep the robot tool within the allowed workspace when one side of the superellipse is significantly longer than the other, as it is further explained in Section 4.6.1.

The equation that describes the proposed modified superellipse is:

$$\left|\frac{x}{W}\right|^m + \left(\frac{\max(|y| - (H - W), 0)}{W}\right)^m = 1, \tag{28}$$

where it has been assumed $H > W$, although it can be readily modified for the analogous case $H < W$, details omitted for brevity.

As shown in Fig. 3, the equation above describes a rectangle with smoothed corners, with $2W$ for its short side and $2H$ for its long side, by attaching a $2W \times 2(H - W)$ rectangle to two offseted halfs of an even-sided $2W \times 2W$ superellipse.

4.4. Force model

The second level of control for the STR includes several equality constraints to properly perform the treatment task. These constraints are defined in Section 4.6.2 depending on the vector $\mathbf{F}$ of force and torque measured by the treatment sensor. In many cases, the vector force $\mathbf{F}$ between the environment and the robotic system can be approximated by the ideal elastic model below (Siciliano, Sciavicco, Villani, & Oriolo, 2009):

$$\mathbf{F}(\mathbf{p}_s, t) = \mathbf{K}_{ts}(t)(\mathbf{p}_s(t) - \mathbf{p}_e(t)) = \begin{bmatrix} F_x & F_y & F_z & F_\alpha & F_\beta & F_\gamma \end{bmatrix}^{\mathrm{T}}, \tag{29}$$

where diagonal matrix $\mathbf{K}_{ts}$ is composed of the treatment sensor stiffness coefficients in each axis, $\mathbf{p}_s$ is the STR pose vector and $\mathbf{p}_e$ is the environment pose vector. Matrix $\mathbf{K}_{ts}$ and vector $\mathbf{p}_e$ are, in general, variable.

4.5. Control for the Workpiece robot

4.5.1. Level 1: Boundary control

An inequality constraint is defined based on the superellipsoid equation described in (26) as follows:

$$\sigma_{w1}(\mathbf{p}_w) = -1 + \left|\frac{x_{wb}}{W}\right|^m + \left|\frac{y_{wb}}{H}\right|^m + \left|\frac{z_{wb}}{M}\right|^m \leq 0, \tag{30}$$

where σ_{w1} defines the boundary of the allowed workspace for the workpiece center position as a superellipsoid (see Section4.3), $\{W, H, M\}$ and m are the axes and smoothing factor, respectively, of the superellipsoid and:

$$\overline{\mathbf{p}}_{wb}(\mathbf{p}_w) = \begin{bmatrix} x_{wb} \\ y_{wb} \\ z_{wb} \end{bmatrix} = {}^b\mathbf{R}_w \left(\begin{bmatrix} \mathbf{I}_3 & \mathbf{O}_3 \end{bmatrix} \mathbf{p}_w - \overline{\mathbf{p}}_{wc} \right), \tag{31}$$

where $\overline{\mathbf{p}}_{wb}$ is the 3D position of the workpiece center relative to the boundary (i.e., the superellipsoid coordinate system), $\mathbf{p}_w$ is the WR pose vector[3], $\overline{\mathbf{p}}_{wc}$ is the 3D position of the boundary/superellipsoid center relative to the WR coordinate system, ${}^b\mathbf{R}_w$ represents the orientation of the boundary/superellipsoid (i.e., the rotation matrix that transforms WR coordinates into superellipsoid coordinates) and $\mathbf{I}_3$ and $\mathbf{O}_3$ denote the identity and zero matrix, respectively, of dimension 3×3.

Note that, as intended, in order for the inequality constraint above to be respected, the workpiece center must be kept within the limits of the superellipsoid. Otherwise, the constraint will become active.

Taking into account (22) and (30)–(31), the modified constraint function ϕ_{w1} for this level results in:

$$\begin{aligned}\phi_{w1}(\mathbf{p}_w) &= \sigma_{w1} + K_{w1}\,\dot{\sigma}_{w1} = \sigma_{w1} + K_{w1}\,(\partial\sigma_{w1}/\partial\mathbf{q}_w)^{\mathrm{T}}\,\dot{\mathbf{q}}_w \\ &= \sigma_{w1} + K_{w1}\,\left((\partial\mathbf{p}_w/\partial\mathbf{q}_w)^{\mathrm{T}}\,(\partial\overline{\mathbf{p}}_{wb}/\partial\mathbf{p}_w)^{\mathrm{T}}\,(\partial\sigma_{w1}/\partial\overline{\mathbf{p}}_b)\right)^{\mathrm{T}}\,\dot{\mathbf{q}}_w \\ &= -1 + \left|\frac{x_{wb}}{W}\right|^m + \left|\frac{y_{wb}}{H}\right|^m + \left|\frac{z_{wb}}{M}\right|^m + K_{w1}\,\mathbf{H}_{w1}\,\mathbf{J}_w\,\dot{\mathbf{q}}_w, \end{aligned} \tag{32}$$

where K_{w1} is the approaching parameter for the WR first level, $\mathbf{J}_w$ is the WR Jacobian matrix and:

$$\mathbf{H}_{w1} = m \begin{bmatrix} \dfrac{\mathrm{sign}(x_{wb})\ |x_{wb}|^{(m-1)}}{W^m} \\ \dfrac{\mathrm{sign}(y_{wb})\ |y_{wb}|^{(m-1)}}{H^m} \\ \dfrac{\mathrm{sign}(z_{wb})\ |z_{wb}|^{(m-1)}}{M^m} \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} {}^b\mathbf{R}_w & \mathbf{O}_3 \end{bmatrix}. \tag{33}$$

Furthermore, from (24) and (32), the required Lie derivative $\mathbf{L_g}\phi_1$ results in:

$$\mathbf{L_g}\phi_{w1} = (\partial\phi_{w1}/\partial\dot{\mathbf{q}}_w)^{\mathrm{T}} = K_{w1}(\partial\sigma_{w1}/\partial\mathbf{q}_w)^{\mathrm{T}} = K_{w1}\,\mathbf{H}_{w1}\,\mathbf{J}_w. \tag{34}$$

From Eqs. (12) and (34), the control equation for WR Level 1 results in:

$$\mathrm{pos}(\phi_{w1})\,K_{w1}\,\mathbf{H}_{w1}\,\mathbf{J}_w\,\ddot{\mathbf{q}}_{wc} = -\mathrm{pos}(\phi_{w1})\,u^+_{w1} \;\rightarrow\; \mathbf{A}_{w1}\,\ddot{\mathbf{q}}_{wc} = b_{w1}, \tag{35}$$

where u^+_{w1} represents the switching gain of the SMC and $\mathbf{A}_{w1}$ and b_{w1} correspond to the matrix and vector (a row vector and a scalar in this case), respectively, for the WR first task in (4).

[3]Note that it has been assumed that the WR grasps the workpiece at its center and, hence, the workpiece center position corresponds to the first three components of the WR pose vector $\mathbf{p}_w$. However, if that would not be the case, the formulation can be easily modified, details omitted for brevity.

4.5.2. Level 2: Orientation control

In order to keep the orientation of the workpiece around a reference value, three inequality constraints are defined as follows:

$$\sigma_{w2,\alpha}(\mathbf{p}_w) = |\alpha_w - \alpha_{w,mid}| - \alpha_{w,max} \leq 0 \tag{36}$$
$$\sigma_{w2,\beta}(\mathbf{p}_w) = |\beta_w - \beta_{w,mid}| - \beta_{w,max} \leq 0 \tag{37}$$
$$\sigma_{w2,\gamma}(\mathbf{p}_w) = |\gamma_w - \gamma_{w,mid}| - \gamma_{w,max} \leq 0, \tag{38}$$

where $\{\alpha_w, \beta_w, \gamma_w\}$ are the actual values for the WR roll, pitch and yaw orientation angles, $\{\alpha_{w,mid}, \beta_{w,mid}, \gamma_{w,mid}\}$ are the mid-range values for these angles and $\{\alpha_{w,max}, \beta_{w,max}, \gamma_{w,max}\}$ are the maximum absolute allowed deviation angles with respect to the mid-range values.

Taking into account (22) and (36)–(38), the modified constraint function vector $\boldsymbol{\phi}_{w2}$ for this level results in:

$$\begin{aligned}\boldsymbol{\phi}_{w2}(\mathbf{p}_w) &= \boldsymbol{\sigma}_{w2} + \mathbf{K}_{w2}\,\dot{\boldsymbol{\sigma}}_{w2} = \boldsymbol{\sigma}_{w2} + \mathbf{K}_{w2}\,(\partial\boldsymbol{\sigma}_{w2}/\partial\mathbf{q}_w)^{\mathrm{T}}\,\dot{\mathbf{q}}_w \\ &= \boldsymbol{\sigma}_{w2} + \mathbf{K}_{w2}\,(\partial\boldsymbol{\sigma}_{w2}/\partial\mathbf{p}_w)^{\mathrm{T}}(\partial\mathbf{p}_w/\partial\mathbf{q}_w)\,\dot{\mathbf{q}}_w \\ &= \begin{bmatrix} |\alpha_w - \alpha_{w,mid}| - \alpha_{w,max} \\ |\beta_w - \beta_{w,mid}| - \beta_{w,max} \\ |\gamma_w - \gamma_{w,mid}| - \gamma_{w,max} \end{bmatrix} + \mathbf{K}_{w2}\,\mathbf{H}_{w2}\,\mathbf{J}_w\,\dot{\mathbf{q}}_w,\end{aligned} \tag{39}$$

where $\boldsymbol{\phi}_{w2}$ and $\boldsymbol{\sigma}_{w2}$ are column vectors composed of the modified and original constraint functions $\phi_{w2,i}$ and $\sigma_{w2,i}$, respectively, see (36)-(38), $\mathbf{K}_{w2}$ is a diagonal matrix composed of all approaching parameters for the WR second level, $\mathbf{J}_w$ is the WR Jacobian matrix and:

$$\mathbf{H}_{w2} = \begin{bmatrix} 0 & 0 & 0 & \mathrm{sign}(\alpha_w - \alpha_{w,mid}) & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathrm{sign}(\beta_w - \beta w, ref) & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathrm{sign}(\gamma_w - \gamma w, ref) \end{bmatrix}. \tag{40}$$

Furthermore, from (24) and (39), the required Lie derivative $\mathbf{L_g}\boldsymbol{\phi}_{w2}$ results in:

$$\mathbf{L_g}\boldsymbol{\phi}_{w2} = (\partial\boldsymbol{\phi}_{w2}/\partial\dot{\mathbf{q}}_w)^{\mathrm{T}} = \mathbf{K}_{w2}(\partial\boldsymbol{\sigma}_{w2}/\partial\mathbf{q}_w)^{\mathrm{T}} = \mathbf{K}_{w2}\,\mathbf{H}_{w2}\,\mathbf{J}_w. \tag{41}$$

Thus, the control equality for the WR second level is:

$$\mathrm{v2dm}(\mathrm{pos}(\boldsymbol{\phi}_{w2}))\,\mathbf{K}_{w2}\,\mathbf{H}_{w2}\,\mathbf{J}_w\,\ddot{\mathbf{q}}_{wc} = -\mathbf{W}_{w2}\,\mathrm{pos}(\boldsymbol{\phi}_{w2})\,u^+_{w2} \;\rightarrow\; \mathbf{A}_{w2}\,\ddot{\mathbf{q}}_{wc} = \mathbf{b}_{w2}, \tag{42}$$

where u^+_{w2} is the switching gain, $\mathbf{W}_{w2}$ is the switching gain weight matrix, and $\mathbf{b}_{w2}$ and $\mathbf{A}_{w2}$ are the vector and matrix, respectively, for the WR third task in (4).

4.5.3. Level 3: Teleoperation for the workpiece robot

In order to add flexibility to the surface treatment operation, the third control level of the WR allows the user to teleoperate the position and orientation of the workpiece.

Taking into account (3), the following control equation is obtained to achieve this

teleoperation:

$$\mathbf{J}_w\ddot{\mathbf{q}}_{wc} = \ddot{\mathbf{p}}_{w,ref} + K_{w3,v}\,\dot{\mathbf{e}}_w + K_{w3,p}\,\mathbf{e}_w + \text{sign}\left(\dot{\mathbf{e}}_w + \frac{K_{w3,p}}{K_{w3,v}}\mathbf{e}_w\right) u^+_{w3}$$
$$\rightarrow \quad \mathbf{A}_{w3}\,\ddot{\mathbf{q}}_{wc} = \mathbf{b}_{w3}, \tag{43}$$

where $\mathbf{J}_w$ is the WR Jacobian matrix, $\mathbf{e}_w = \mathbf{p}_{w,ref} - \mathbf{p}_w$ and $\dot{\mathbf{e}}_w = \dot{\mathbf{p}}_{w,ref} - \dot{\mathbf{p}}_w$ represent the WR pose and velocity error, respectively, $K_{w3,p}$ and $K_{w3,v}$ are their correspondent correction gains, and $\mathbf{b}_{w3}$ and $\mathbf{A}_{w3}$ are the vector and matrix for the WR third task in (4). It must be noted that $\dot{\mathbf{p}}_w$ is obtained from the first order robot kinematics in (2), and that the reference $\mathbf{p}_{w,ref}$ is generated by the human teleoperator.

Note that a hybrid control approach is developed for this control level, since the term $\dot{\mathbf{J}}_w\dot{\mathbf{q}}_w$ from Eq. (3) is substituted by the conventional SMC switching term $\text{sign}\left(\dot{\mathbf{e}}_w + \frac{K_{w3,p}}{K_{w3,v}}\mathbf{e}_w\right)$. This hybrid control presents two advantages: the Jacobian derivative is not needed, so its computation is avoided; and, thanks to the continuous control terms in the control action, the switching gain $\mathbf{u}^+_{w3}$ can be relatively small, achieving a reduced chattering effect.

4.5.4. Level 4: Home configuration

If the STR robot is *redundant*, e.g., like the 7R cobot used in the experiments below, the DoF that remain at this point can be used for other objectives, avoiding also a bias self-motion. In this research, a home configuration $\mathbf{q}_{w0}$ is considered to "push" the STR to it for increasing safety. In particular, the following control law is used:

$$\ddot{\mathbf{q}}_{wc} = -K_{w4,v}\,\dot{\mathbf{q}}_w + K_{w4,p}(\mathbf{q}_{w0} - \mathbf{q}_w) \;\rightarrow\; \mathbf{A}_{w4}\,\ddot{\mathbf{q}}_{wc} = \mathbf{b}_{w4}, \tag{44}$$

where $K_{w4,v}$ and $K_{w4,p}$ are the velocity and position correction gains, respectively, and $\mathbf{b}_{w4}$ and $\mathbf{A}_{w4}$ are the vector and matrix for the WR fourth task in (4)

4.6. Control for the surface treatment robot

4.6.1. Level 1: Boundary control

Similarly to Level 1 of the WR control (see Section 4.5.1), an inequality constraint is defined based on the modified superellipse equation described in (28) as follows:

$$\sigma_{s1}(\mathbf{p}_s, t) = -1 + \left|\frac{x_{sb}}{W}\right|^m + \left(\frac{\max(|y_{sb}| - (H - W), 0)}{W}\right)^m \leq 0, \tag{45}$$

where σ_{s1} defines the boundary of the allowed surface on the workpiece for the STR tool as a modified superellipse (see Section4.3), $\{W, H\}$ and m are the axes and smoothing factor, respectively, of the modified superellipse and:

$$\overline{\mathbf{p}}_{sb}(\mathbf{p}_s, t) = \begin{bmatrix} x_{sb} \\ y_{sb} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} {}^b\mathbf{R}_s(t)\, \left(\begin{bmatrix} \mathbf{I}_3 & \mathbf{O}_3 \end{bmatrix} \mathbf{p}_s - \overline{\mathbf{p}}_{sc}(t) \right), \tag{46}$$

where $\overline{\mathbf{p}}_{sb}$ is the 2D position of the STR tool with respect to the boundary (i.e., the superellipse coordinate system), $\mathbf{p}_s$ is the STR pose vector, $\overline{\mathbf{p}}_{sc}$ is the 3D position

of the superellipse center relative to the STR coordinate system, ${}^b\mathbf{R}_s$ represents the orientation of the boundary/superellipse (i.e., the rotation matrix that transforms STR coordinates into superellipse coordinates) and $\mathbf{I}_3$ and $\mathbf{O}_3$ denote the identity and zero matrix, respectively, of dimension 3×3.

Note that the values of the position $\overline{\mathbf{p}}_{sc}$ and orientation ${}^b\mathbf{R}_s$ of the modified superellipse are readily obtained from the WR pose vector $\mathbf{p}_w$, since the WR holds the workpiece and the superellipse is virtually attached to the workpiece surface.

As before, in order for the inequality constraint above to be respected, the STR tool position must be kept within the limits of the modified superellipse. Otherwise, the constraint will become active.

Taking into account (22), the modified constraint function ϕ_{s1} for this level results in:

$$\phi_{s1}(\mathbf{p}_s, t) = \sigma_{s1} + K_{s1}\, \dot{\sigma}_{s1} \tag{47}$$

where K_{s1} is the approaching parameter for the STR first level.

Note that the modified superellispe is non-static due to the workpiece movement, i.e., both its position $\overline{\mathbf{p}}_{sc}$ and orientation ${}^b\mathbf{R}_s$ are variable. Therefore, the value of $\dot{\sigma}_{s1}$ in (47) cannot be computed from the STR joint velocity vector $\dot{\mathbf{q}}_s$ alone and, hence, numerical differentiation of σ_{s1} is used in the practical implementation below.

Taking into account (46) and (47), the use of a modified superellipse as described in (28) is justified as follows. If $H > W$ and σ_{s1} was based on the regular superellipse given by (27), a movement in the X-axis of the superellipse would cause a much faster increase in σ_{s1} and ϕ_{s1} than a movement in the Y-axis with the same speed, which would, in turn, cause the control action to activate sooner in the first case and, thus, make the approach to the boundary smoother than in the second case, where the control action would be later and, as a consequence, rougher.

In contrast, by using the modified superellipse equation (28), the allowed workspace remains a $2W \times 2H$ rectangle with smoothed corners, see Fig. 3, while the way in which σ_{s1} and ϕ_{s1} evolve depending on the direction of the tool movement is more homogeneous, since both terms $|x_{sb}|$ and $\max(|y_{sb}| - (H - W), 0)$ are bounded between 0 and W.

From (24), (47) and (46), the required Lie derivative $\mathbf{L}_{\mathbf{g}}\phi_{s1}$ results in:

$$\begin{aligned}
\mathbf{L}_{\mathbf{g}}\phi_{s1} &= \left(\partial\phi_{s1}/\partial\dot{\mathbf{q}}_s\right)^{\mathrm{T}} = K_{s1}\left(\partial\sigma_{s1}/\partial\mathbf{q}_s\right)^{\mathrm{T}} \\
&= K_{s1}\left(\left(\partial\mathbf{p}_s/\partial\mathbf{q}_s\right)^{\mathrm{T}}\left(\partial\overline{\mathbf{p}}_{sb}/\partial\mathbf{p}_s\right)^{\mathrm{T}}\left(\partial\sigma_{s1}/\partial\overline{\mathbf{p}}_s\right)\right)^{\mathrm{T}} \\
&= K_{s1}\,\mathbf{H}_{s1}\,\mathbf{J}_s,
\end{aligned} \tag{48}$$

where $\mathbf{J}_s$ is the STR Jacobian matrix and:

$$\mathbf{H}_{s1} = m \begin{bmatrix} \dfrac{\operatorname{sign}(x_{sb})\ |x_{sb}|^{(m-1)}}{W^m} \\ \dfrac{(\max(|y_{sb}| - (H-W), 0))^{(m-1)}}{H^m} \\ 0 \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} {}^b\mathbf{R}_s & \mathbf{O}_3 \end{bmatrix}. \tag{49}$$

From (12) and (48), the control equation for STR Level 1 results in:

$$\text{pos}(\phi_{s1})\, K_{s1}\, \mathbf{H}_{s1}\, \mathbf{J}_s\, \ddot{\mathbf{q}}_{sc} = -\text{pos}(\phi_{s1})\, u_{s1}^{+} \;\rightarrow\; \mathbf{A}_{s1}\, \ddot{\mathbf{q}}_{sc} = b_{s1}, \tag{50}$$

where u_{s1}^{+} represents the switching gain of the SMC and $\mathbf{A}_{s1}$ and b_{s1} correspond to the matrix and vector (a row vector and a scalar in this case), respectively, for the STR first task in (4).

4.6.2. Level 2: Treatment task constraints

The following equality constraints are considered for the robot surface treatment:

$$\sigma_{s2,z}(\mathbf{F}) = \sigma_{s2,z}(\mathbf{p}_s, t) = F_z - F_{z,ref} = 0 \tag{51}$$
$$\sigma_{s2,\alpha}(\mathbf{F}) = \sigma_{s2,\alpha}(\mathbf{p}_s, t) = F_\alpha = 0 \tag{52}$$
$$\sigma_{s2,\beta}(\mathbf{F}) = \sigma_{s2,\beta}(\mathbf{p}_s, t) = F_\beta = 0, \tag{53}$$

where F_z is the measurement of the F/T sensor in the linear Z-axis of STR tool, F_α and F_β are the measurements of the F/T sensor in angular X- and Y-axes of the STR tool, and $F_{z,ref}$ is the reference value for the force F_z. Thus, the first equality above is used to accomplish the reference pressure $F_{z,ref}$ between the workpiece and the STR tool, and the last two equalities are used to maintain the tool orthogonality to the workpiece, i.e., the above angular torques are zero if the STR tool is completely orthogonal to the workpiece. Note that the human operator could teleoperate the torque in the angular Z-axis because it is not restricted.

From (21) and (51)–(53), the modified constraint function vector $\boldsymbol{\phi}_{s2}$ for this level results in:

$$\boldsymbol{\phi}_{s2}(\mathbf{F}, \dot{\mathbf{F}}) = \boldsymbol{\sigma}_{s2} + \mathbf{K}_{s2}\, \dot{\boldsymbol{\sigma}}_{s2} = \begin{bmatrix} F_z - F_{z,ref} + K_{s2,z}\, \dot{F}_z \\ F_\alpha + K_{s2,\alpha}\, \dot{F}_\alpha \\ F_\beta + K_{s2,\beta}\, \dot{F}_\beta \end{bmatrix}, \tag{54}$$

where $\boldsymbol{\phi}_{s2}$ and $\boldsymbol{\sigma}_{s2}$ are column vectors composed of the modified and original constraint functions $\phi_{s2,i}$ and $\sigma_{s2,i}$, respectively, and diagonal matrix $\mathbf{K}_{s2}$ with the approaching parameters for the STR second level.

Furthermore, taking into account (21), (24), (29) and (51) –(53), the required Lie derivative $\mathbf{L_g}\boldsymbol{\phi}_{s2}$ results in:

$$\begin{aligned} \mathbf{L_g}\boldsymbol{\phi}_{s2} &= (\partial \boldsymbol{\phi}_{s2}/\partial \dot{\mathbf{q}}_s)^{\mathrm{T}} = \mathbf{K}_{s2}(\partial \boldsymbol{\sigma}_{s2}/\partial \mathbf{q}_s)^{\mathrm{T}} = \mathbf{K}_{s2}(\partial \boldsymbol{\sigma}_{s2}/\partial \mathbf{p}_s)^{\mathrm{T}}(\partial \mathbf{p}_s/\partial \mathbf{q}_s) \\ &= \mathbf{K}_{s2} \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \mathbf{K}_{ts}\mathbf{J}_{sn} = \mathbf{K}_{s2}\mathbf{H}_{s2}\mathbf{K}_{ts}\mathbf{J}_{sn}, \end{aligned} \tag{55}$$

where diagonal matrix $\mathbf{K}_{ts}$ is associated to the treatment sensor stiffness and $\mathbf{J}_{sn}$ is the STR geometric Jacobian relative to the tool coordinate system (Siciliano et al., 2009), that is, the Jacobian that relates the velocity vector $\dot{\mathbf{q}}_s$ and the STR end-effector velocities with respect to the treatment tool frame.

The stiffness parameters $\mathbf{K}_{ts}$ in $\mathbf{L_g}\boldsymbol{\phi}_{s2}$ (55) are usually not known, but, without loss of generality, they can be incorporated in the switching gain weight matrix $\mathbf{W}_{s2}$.

Thus, the SMC given by (9) results in:

$$\mathbf{K}_{s2}\,\mathbf{H}_{s2}\,\mathbf{J}_{sn}\,\ddot{\mathbf{q}}_{sc} = -\overline{\mathbf{W}}_{s2}\,\mathrm{sign}(\boldsymbol{\phi}_{s2})\,u_{s2}^{+} \;\rightarrow\; \mathbf{A}_{s2}\,\ddot{\mathbf{q}}_{sc} = \mathbf{b}_{s2}, \tag{56}$$

where u_{s2}^{+} represents the switching gain of the SMC and $\mathbf{b}_{s2}$ and $\mathbf{A}_{s2}$ are the vector and matrix for the STR second task in (4) and:

$$\overline{\mathbf{W}}_{s2} = \begin{bmatrix} W_{s2,z}/K_{ts,z} & 0 & 0 \\ 0 & W_{s2,\alpha}/K_{ts,\alpha} & 0 \\ 0 & 0 & W_{s2,\beta}/K_{ts,\beta} \end{bmatrix} = \begin{bmatrix} \overline{W}_{s2,z} & 0 & 0 \\ 0 & \overline{W}_{s2,\alpha} & 0 \\ 0 & 0 & \overline{W}_{s2,\beta} \end{bmatrix}. \tag{57}$$

Note that the conventional SMC given by (56) only requires: the control parameters $\{u_{s2}^{+}, \overline{\mathbf{W}}_{s2}, \mathbf{K}_{s2}, F_{z,ref}\}$; the robot Jacobian $\mathbf{J}_{sn}$; and the treatment sensor measurement $\mathbf{F}$ and its derivative.

4.6.3. Level 3: Surface treatment tool teleoperation

In order to carry out the surface treatment operation, the third control level of the STR allows the user to teleoperate the tool position on the workpiece surface with respect to the superellipse coordinate system.

Similarly to Level 3 of the WR control (see Sec. 4.5.3), the following control equation is obtained to achieve this teleoperation:

$$\begin{aligned} \mathbf{J}_{sb}\ddot{\mathbf{q}}_{sc} &= \ddot{\overline{\mathbf{p}}}_{sb,ref} + K_{s3,v}\,\dot{\overline{\mathbf{e}}}_{sb} + K_{s3,p}\,\overline{\mathbf{e}}_{sb} + \mathrm{sign}\left(\dot{\overline{\mathbf{e}}}_{sb} + \frac{K_{s3,p}}{K_{s3,v}}\overline{\mathbf{e}}_{sb}\right) u_{s3}^{+} \\ &\rightarrow\; \mathbf{A}_{s3}\,\ddot{\mathbf{q}}_{sc} = \mathbf{b}_{s3}, \end{aligned} \tag{58}$$

where $\overline{\mathbf{p}}_{sb,ref}$ is the reference generated by the human teleoperator for the 2D position of the STR tool with respect to the boundary/superellipse coordinate system, $K_{s3,p}$ and $K_{s3,v}$ are the position and velocity correction gains, respectively, u_{s3}^{+} represents the switching gain of the SMC, $\mathbf{b}_{s3}$ and $\mathbf{A}_{s3}$ are the vector and matrix for the STR third task in (4) and:

$$\mathbf{J}_{sb} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} {}^{b}\mathbf{R}_{s} & \mathbf{O}_{3} \end{bmatrix} \mathbf{J}_{s} \tag{59}$$

$$\overline{\mathbf{e}}_{sb} = \overline{\mathbf{p}}_{sb,ref} - \overline{\mathbf{p}}_{sb} \tag{60}$$

$$\dot{\overline{\mathbf{e}}}_{sb} = \dot{\overline{\mathbf{p}}}_{sb,ref} - \dot{\overline{\mathbf{p}}}_{sb} \tag{61}$$

where $\overline{\mathbf{e}}_{sb}$ and $\dot{\overline{\mathbf{e}}}_{sb}$ are the 2D position and velocity errors, respectively, of the STR tool with respect to the boundary/superellipse coordinate system.

Note that the derivative $\dot{\overline{\mathbf{p}}}_{sb}$ in (61) cannot be computed from the STR joint velocity vector $\dot{\mathbf{q}}_{s}$ alone since, as mentioned above, the boundary/superellipse is non-static due to the workpiece movement, see (46). Hence, numerical differentiation of $\overline{\mathbf{p}}_{sb}$ is used in the practical implementation below.

In the same way as in Level 3 for the WR control, note that a hybrid control approach is developed for this control level, substituting $\dot{\mathbf{J}}_{sb}\dot{\mathbf{q}}_{s}$ by the conventional SMC switching term $\mathrm{sign}\left(\dot{\overline{\mathbf{e}}}_{sb} + \frac{K_{s3,p}}{K_{s3,v}}\overline{\mathbf{e}}_{sb}\right)$.

4.7. Limitations of the proposed approach

The main limitations of the proposed method are discussed as follows:

- *High-level planning*: The proposed robot control does not include high-level planning, i.e., the robot controller "reacts" to the teleoperator commands and robot constraints with no kind of prediction or long-term anticipation. Therefore, the algorithm may be blocked in trap situations (Gracia et al., 2012) or achieve singular configurations (Gracia et al., 2009) where the robot system "loses" DoF, which may degrade the performance of the robot task. In some cases, these situations could be avoided by employing a high-level planner (see the discussion about task optimization in Section 1.2.3) to solve the problem using a large prediction horizon and taking into account the complete data, including a priori knowledge of the teleoperator commands, which may not be possible in practice. Nevertheless, the complexity and computational cost of this high-level planner would be significantly greater than those of the proposed method.

  However, singular configurations could also be avoided in the proposed method by including, similarly to the boundary constraints above, another inequality constraint in order to prevent the robot from entering workspace regions where the condition number of the robot Jacobian is above a given threshold (note that, at singular configurations, the condition number tends to infinity). However, this is out of the scope of this research and remains as further work.
- *Chattering*: SMC discrete-time implementations make the system oscillate with finite amplitude and frequency within a band around $\boldsymbol{\phi} = \mathbf{0}$, namely *chattering* (Edwards & Spurgeon, 1998). For the proposed method, the chattering band $\triangle\boldsymbol{\phi}$ has an upper bound that can be computed as follows using the Euler-integration of the SMC action in (9):

$$\triangle\boldsymbol{\phi} = T\,|\mathbf{L_g}\boldsymbol{\phi}\,\mathbf{u}| = T\,u^{+}\,\mathrm{dm2v}(\mathbf{W}), \tag{62}$$

  where function $\mathrm{dm2v}(\cdot)$ converts a diagonal matrix into a column vector and T^{-1} is the sampling frequency of the robotic system.

  Nevertheless, for reasonably fast sampling the chattering drawback can be neglected. For instance, in the real experimentation of Section 6, sampling frequencies of 75Hz and 200Hz gave rise to negligible chattering effect.

  However, it is important to remark that there are several approaches in the literature to reduce the chattering effect of the SMC: using a super-twisting or twisting SMC (Levant, 2003); using a quasi-continuous SMC (Levant, 2005); using the hyperbolic tangent curve as switching function (Edwards & Spurgeon, 1998); etc. However, this is out of the scope of this research.

5. Controller implementation

Table 1 and Table 2 show the pseudo-code of the controllers developed in this work for the WR and STR, respectively. Note that it is assumed that these controllers are implemented in a modern computer, although they could also be easily implemented in other platforms such as: an industrial workstation; an embedded processor; a Programmable Logic Controller (PLC); or even directly in the robot controller.

The algorithms of the WR and STR control are executed at T_w and T_s seconds

sampling periods, respectively, and incorporate the auxiliary functions below:

- Kinematic functions and Jacobian matrices: $\mathbf{l}_w(\mathbf{q}_w)$, $\mathbf{l}_s(\mathbf{q}_s)$, $\mathbf{J}_w(\mathbf{q}_w)$, $\mathbf{J}_s(\mathbf{q}_s)$ and $\mathbf{J}_{sn}(\mathbf{q}_s)$.
- Moore-Penrose pseudoinverse (a tolerance is used to discard small singular values, see Section 2): $(\cdot)^\dagger$.
- Function *Pose2PositionAndRot*$(\mathbf{p}_w)$ that converts the WR pose vector into the position $\overline{\mathbf{p}}_{sc}$ and orientation ${}^b\mathbf{R}_s$ of the modified superellipse.
- Robot sensors:
  - *GetWRState*() for the WR algorithm, which returns the current WR state $\{\mathbf{q}_w, \dot{\mathbf{q}}_w\}$.
  - *GetSTRStateAndForces*() for the STR algorithm, which returns the current STR state $\{\mathbf{q}_s, \dot{\mathbf{q}}_s\}$ and forces $\mathbf{F}$ measured by the treatment sensor, which is assumed that has been filtered by the electronics of the treatment sensor.
- Robot communications:
  - *GetUserReferenceWR*() which reads the reference pose $\mathbf{p}_{w,ref}$ determined by the human teleoperator for the workpiece, in the WR algorithm.
  - *GetUserReferenceSTR*() which reads the reference position $\overline{\mathbf{p}}_{sb,ref}$ determined by the human teleoperator for the STR tool.
  - *GetWRPose*() which obtains the current WR pose $\mathbf{p}_w$ by reading data received from the WR, in the STR algorithm.
- Actuators:
  - *SendToWRJointControllers*$(\mathbf{q}_{wc})$, which commands the desired joint values to the WR joint controllers.
  - *SendToSTRJointControllers*$(\mathbf{q}_{sc})$, which commands the desired joint values to the STR joint controllers.

Note that, in the code of Table 1 and Table 2, numerical differentiation (i.e., the well-known backward Euler approximation) is used to obtain the derivatives of the signals $\{\mathbf{p}_{w,ref}\}$ and $\{\sigma_{s1}, \mathbf{F}, \overline{\mathbf{p}}_{sb}, \overline{\mathbf{p}}_{sb,ref}\}$ in the WR and STR control algorithms, respectively. However, depending on the application, this approach could lead to excessive noise in the signals. In order to avoid this issue, the sampling period of the corresponding control algorithm should be chosen large enough in order to avoid noisy signals[4]. For instance, in the specific case of the experimentation in Section 6, a synchronous sampling period of 5 ms and 13 ms for the WR and STR control algorithms, respectively, gave rise to negligible noise in numerical differentiation.

From a computational complexity point of view, the algorithms of the WR and STR have 31 and 35 lines of code, respectively, see Table 1 and Table 2. Moreover, the computational cost of both algorithms implemented in a modern computer is around 12 microseconds for the experiments in Section 6, although this value could change if the algorithms are implemented in a different platform, e.g., an industrial workstation, a PLC, etc.

[4]Note that if a small sampling period is chosen, a low-pass filter would be required to remove the noise from the derivatives. However, the bandwidth of this approach is approximately equivalent to use a larger sampling period with no filtering.

6. Real experimentation

6.1. Setup

The experimental platform of this work can be seen in Fig. 4. Its components are detailed below.

– STR: A 6R robot arm (KUKA KR6 Agilus), whose Denavit-Hartenberg parameters are displayed in Table 3; an Axia80 F/T sensor attached to the STR end-effector; and a self-developed brush of 14 cm in diameter to carry out a demonstrative surface treatment, which is attached to the F/T sensor.
– WR: A 7R cobot (KUKA LBR iiwa 14 r820), whose DH parameters are shown in Table 4; and a methacrylate board measuring 28x38 cm (the workpiece) rigidly attached to the WR end-effector by means of a self developed adaptor.

An external computer using *Ubuntu 16.04* ans O.S. and *ROS Kinetic* distribution has been used to implement the different algorithms specified in Section 5, and an XBox controller allows the user to teleoperate both robot arms simultaneously. The robot arms, F/T sensor and external computer were communicated by an Ethernet switch. It must be noted that, the KUKA LBR iiwa robot has two available Ethernet ports, of which, particularly, the one correspondent to the FRI (Fast Robot Interface) was used to communicate the robot with the external computer, as it allows sampling periods up to 2 ms. On the other hand, the remote controller is communicated with the computer via a USB connection.

6.2. Experiment conditions and parameter values

The values of the parameters of the two control algorithms were empirically tuned to obtain a proper performance of both robot arms.

The parameter values for the WR are as follows:

- WR control rate: $T_w^{-1} = 200\text{Hz}$.
- Parameters of WR Level 1: $K_{w1} = 1.0$, $u_{w1}^{+} = 4.0$, $W = H = M = 0.06$, $m = 4$ and the position and orientation of the superellipsoid match the initial value of the position and orientation of workpiece (which are given by the WR end-effector pose), respectively, i.e., $\overline{\mathbf{p}}_{wc} = \overline{\mathbf{p}}_w(0)$ and ${}^b\mathbf{R}_w = {}^{wp}\mathbf{R}_w(0)$.
- Parameters of WR Level 2: $\mathbf{K}_{w2} = \text{v2dm}([0.5\ 0.5\ 2])$, $\mathbf{W}_{w2} = \mathbf{I}_3$, $u_{w2}^{+} = 1.0$, $\alpha_{w,max} = \beta_{w,max} = \gamma_{w,max} = 15^\circ$ and the mid-range values for the workpiece orientation angles correspond to their initial values, i.e., $\alpha_{w,mid} = \alpha_w(0)$, $\beta_{w,mid} = \beta_w(0)$ and $\gamma_{w,mid} = \gamma_w(0)$.
- Parameters of WR Level 3: $K_{w3,v} = 1.5$, $K_{w3,p} = 1.0$ and $u_{w3}^{+} = 0.001$.
- Parameters of WR Level 4: $K_{w4,v} = 0.8$ and $K_{w4,p} = 0.5$.

The parameter values for the STR are as follows:

- STR control rate: $T_s^{-1} = 75\text{Hz}$.
- Parameters of STR Level 1: $K_{s1} = 0.6$, $u_{s1}^{+} = 6.0$, $W = 0.07$, $H = 0.12$ and $m = 4$.
- Parameters of STR Level 2: $\mathbf{K}_{s2} = \text{v2dm}([1.5\ 2\ 2])$, $\mathbf{W}_{s2} = \text{v2dm}([0.4\ 0.6\ 0.6])$, $u_{s2}^{+} = 1.0$ and $F_{z,ref} = 20\text{N}$
- Parameters of STR Level 3: $K_{s3,v} = 1.8$, $K_{s3,p} = 1.3$ and $u_{s3}^{+} = 0.01$.

Note that the difference between both control rates is not problematic, since the STR control algorithm needs information sent by the WR control algorithm, which is ensured, as the former is slower than the latter, but the WR control algorithm does not need information coming from the STR control algorithm.

6.3. Results

Seven experiments were conducted in order to study and verify the performance of the two control algorithms working separately and together, and of the different control laws of which they are composed. These experiments are arranged in three groups:

- Two experiments to analyze the behavior of the WR control algorithm.
- Four experiments to analyze the behavior of the STR control algorithm.
- One experiment to analyze the behavior of the whole bimanual application.

Note that, in the videos below for each experiment, the upper right corner shows a virtual image of the bimanual robot workspace with the following information to assist the user during the teleoperation: the *blue* dots represent the boundary of the allowed area on the workpiece for the STR tool (i.e., the 2D modified superellipse), which turn *green* when the corresponding boundary constraint becomes active; the *static yellow* dot corresponds to the initial position of the workpiece center; the *purple* dot is the current position of the workpiece center, i.e., of the WR end-effector; the *cyan* dot is the reference position commanded by the user to the workpiece center, i.e., to the WR end-effector; the *red* dot is the current position of the STR tool; and the *moving yellow* dot is the reference position commanded by the user to the STR tool. Note that some of these dots do not apply for some experiments. For instance, in the video of Experiment 1, which is focused only on the WR, the red and moving yellow dots are not shown since they are related to the STR.

6.3.1. Experiments for the WR control algorithm

There are two experiments belonging to this section: the first experiment (Experiment 1) focuses mainly on the study of the first control level, i.e., the boundary constraint, whereas the second experiment (Experiment 2) focuses mainly on the study of the second control level, i.e., the angular restriction. Moreover, both experiments study the behavior of the remaining control levels (third, reference tracking, and fourth, redundancy resolution), as in both cases the WR is teleoperated.

In Experiment 1 the WR is teleoperated (WR Level 3) so that the reference position commanded by the human operator is outside the allowed workspace, i.e., the 3D superellipsoid, thus activating the boundary constraint (WR Level 1) and keeping the workpiece inside the intended area, see the `video` (Video of Experiment 1, 2021).

Several graphs are presented below in order to verify the quantitative performance of Experiment 1. In Fig. 5, the functions and activation of the boundary constraint in WR Level 1 are shown. Observe that this constraint is activated during two intervals, approximately from 19s to 50s and from 105s and 179s, and during the whole time of the experiment, the workpiece center is kept within the boundary, as can be seen in Fig. 6. Particularly, in Fig. 6 it can be appreciated that the reference position often surrounds the boundary, while the workpiece center travels along it without surpassing it. It also can be seen that this boundary is a superellipsoid relative to the initial position of the center of the workpiece.

In Experiment 2 the WR is teleoperated (WR Level 3) so that the angular references for the workpiece are beyond the allowed maximum angles (relative to its initial angular position), thus activating the angular restriction (WR Level 2) and keeping the workpiece angular position within the permitted interval, see the `video` (Video of Experiment 2, 2021).

Fig. 7 shows the constraint functions of each one of the three restrictions. The restrictions are activated as follows to prevent the constraint functions $\sigma_{w2,i}$ surpassing the zero value: first, the pitch restriction is activated twice, then the roll restriction is also activated twice, then the yaw restriction is activated once and, finally, all restrictions are activated simultaneously. During the second activation of the roll restriction and the activation of the yaw restriction, the pitch angle reference remains close to its limit, so its restriction is intermittently activated and deactivated. All this information is completed by observing Fig. 8, which shows how the angular reference goes beyond the permitted limits, while the actual position of the workpiece center always respects those limits.

6.3.2. Experiments for the STR control algorithm

As explained at the beginning of Section 6.3, there are four experiments belonging to this section: Experiment 3 and Experiment 4 study the behavior of the STR boundary constraint (STR Level 1), while Experiment 5 and Experiment 6 focus on the force and torque (F/T) control in order to keep contact and perpendicularity at all times, respectively (STR Level 2). Moreover, Experiment 3, Experiment 5 and Experiment 6 also include user teleoperation, so they also explore the behavior of the reference tracking control law (STR Level 3).

Experiment 3 shows the performance of the boundary constraint when it remains static, in a similar fashion to Experiment 1 in Section 6.3.1. In this experiment, the STR is teleoperated (STR Level 3) so that the reference position for the tool tip in local coordinates of the workpiece frame tries to guide the tool beyond its permitted area of work, thus activating the boundary constraint and keeping the tool inside the allowed area, see the `video` (Video of Experiment 3, 2021).

Several graphs are presented below in order to verify the quantitative performance of Experiment 3. In Fig. 9, the functions and activation of the boundary constraint in STR Level 1 are shown. Observe that this constraint is activated for an interval of approximately 131s (from 31s to 132s) and during the whole time of the experiment, the STR tool is kept within the boundary, as can be seen in Fig. 10. Thus, despite that the reference tries to take the tool tip out of the allowed area (i.e., the 2D modified superellipse), the actual tool tip always stays within due to the activation of the control law of the boundary constraint (STR Level 1), which has a higher priority than the reference tracking control law (STR Level 3).

Several frames of the video of Experiment 3 depicted in Fig. 11 show how the surface treatment tool never leaves the workpiece area. This is due to the fact that the allowed area, a modified superellipse (28), has been calculated taking into account both the workpiece dimensions and the tool diameter, see Fig. 10.

Additionally, the quantitative performance of the surface treatment task can be seen in Fig. 12, where it is shown the behavior of the constraint functions $\{\sigma_{s2,z}, \sigma_{s2,\alpha}, \sigma_{s2,\beta}\}$, see (51)–(53). In particular, it can be seen that, thanks to the control law in Level 2 of the STR, all three functions are switching around zero, i.e., $\{F_z \approx F_{z,ref}, F_\alpha \approx 0, F_\beta \approx 0\}$, which means that the surface treatment is being performed properly. That is, the tool pressure and perpendicularity are kept regardless of the changes on the

workpiece and treatment tool positions, which are commanded by the human user, and even though the sudden deformations of the methacrylate workpiece.

It should be noted that the STR control algorithm must take into account the movement of the workpiece, since the boundary is defined locally to the workpiece, so it will move during the bimanual task. Thus, another experiment (Experiment 4) in order to fully explore the first level of control is necessary, so as to study the behavior of the STR boundary constraint when it is the boundary itself that is moving and the surface treatment tool is trying to remain static.

Being so, in Experiment 4, the WR moves the workpiece along its end-effector X-axis (note that the end-effector Z-axis is along the last link of the robot, as usual) following a sinusoidal movement of amplitude 12cm and time period 30s, while the STR tries to keep the treatment tool still, so the boundary approaches the tool and its constraint goes active, pushing the tool and keeping it inside the permitted area, see the `video` (Video of Experiment 4, 2021).

Observe that in Fig. 13, which presents a set of frames from the video of Experiment 4, the tool is pushed away from the workpiece limits when those limits reach the tool. In fact, the boundary constraint is activated before the tool reaches the limits, thanks to the variation in the distance between the tool and the boundary being computed in the modified constraint function ϕ_{s1} (see Section 3.3), allowing the STR to anticipate, as can be seen in graphs in Fig. 14, which show the functions and activation of the first control law, where ϕ_{s1} surpasses the threshold and activates the constraint well before σ_{s1} is in risk of surpassing it.

Note that, compared to the same functions in Fig. 9, in this case σ_{s1} is kept further away from the limit, since, in this case, the boundary is moving, so σ_{s1} changes faster, which is reflected in higher values of ϕ_{s1}. This anticipation is critical when the boundary is moving, as the STR has no control over how fast the boundary moves, so its reaction must be quick enough to avoid breaking the restriction. This is reflected in Fig. 15 as well, where the distance from the STR tool to the limits of the modified superellipse tends to be bigger than that displayed in Fig. 10 of Experiment 3, where the boundary remains static and, so, the surface treatment tool can approach the boundary more slowly.

Up next, a more detailed study of the performance of STR Level 2, i.e., force control in order to keep contact between the tool and the workpiece and torque control in order to keep perpendicularity between them, is necessary.

Experiment 5 poses a more challenging situation for STR Level 2 than the two previous ones. In this experiment, the WR moves the workpiece back and forth following its end-effector Z-axis with a sinusoidal movement of amplitude 5cm and time period 20s, thus pushing the surface treatment tool while the STR is teleoperated while it tries to keep contact without impeding the workpiece advance. Eventually, the WR changes the course, forcing the STR control algorithm to adapt so as to not detach from the workpiece, see the `video` (Video of Experiment 5, 2021). Note that during this experiment, the user is still able to teleoperate the STR and the boundary constraint works as expected, so the experiment offers a more complete outlook on the STR control algorithm and its capabilities.

Several frames of the video of Experiment 5 have been selected in Fig. 16 to show the moment of the change of course in the movement of the workpiece, and how contact is correctly maintained.

Graphs shown in Fig. 17 back this visual observation: during the whole experiment, pressure over the surface of the workpiece oscillates around the 20N reference, being its higher value 52.51 N and its lower value 2.33N, over 0N, so contact is ensured.

Although it is not the main purpose of this experiment, these graphs also show how perpendicularity is kept thanks to the constant correction of the torque exerted over the F/T sensor, so it always oscillates about 0, without deviating to one side or the other.

Fig. 18 shows the roll and pitch angles of the surface treatment tool. Note that the orientation of the STR tool is kept relatively still, with variations between peaks of 9.25° and 5.72° in roll and pitch angles, respectively, which are small deviations taking into account that the workpiece is not a totally rigid surface, so the orientation must be adapted, especially when the tool approaches the limits of the workpiece, where it is less rigid.

As explained above, the STR boundary restriction is also respected, as can be seen in Fig. 19

Finally, Experiment 6 in this section presents a situation similar to that of Experiment 5, but instead of moving the workpiece back and forth, the WR describes simultaneous angular movements, so the torque control behavior of STR Level 2 is explored more deeply. In this experiment, the workpiece describes an automatic sinusoidal movement of amplitude 15° for roll and pitch angles and 30° for yaw angle, and time period 40s for all three orientation angles. As it can be observed in the `video` (Video of Experiment 6, 2021), the STR is able to adjust successfully to these changes while being teleoperated.

A set of frames from the video of Experiment 6 is presented in Fig. 20, showing different orientations of the workpiece. It must be noted that in all cases, the STR has been able to adapt correctly, keeping the tool orthogonal to the surface of the workpiece without detaching from it.

Studying graphs in Fig. 21, which show F/T measurements, it is observed a similar situation to that of the former experiment, where perpendicularity is kept and that is reflected in the control law being able to keep the torque values oscillating around 0. Note that torque values do not stay in a negative or positive torque value for intervals longer than 3s (roll) and 4s (pitch) in the worst cases, which means the control law is able to adjust the orientation even when the workpiece orientation is changing and the STR is being teleoperated simultaneously.

Moreover, similarly to Experiment 5 above, the STR boundary constraint is also activated and able to keep the surface treatment tool inside the allowed modified superellipse even though the human teleoperator tries to guide it beyond the boundary, as can be seen in Fig. 22.

6.3.3. Experiment for the bimanual application

This section focuses on the practical study of this contribution as a whole, i.e., the bimanual assisted teleoperation for a surface treatment task. In this last experiment (Experiment 7) every factor studied in the last two sections is present: the user teleoperates the WR in global coordinates (i.e., 3D Cartesian position $[x\, y\, z]$ and orientation angles $[\alpha\ \beta\ \gamma]$) and the STR in local coordinates of the workpiece (i.e., 2D Cartesian position $[x\ y]$), and even though the task is complex, the different restrictions introduced by both WR and STR control algorithms and the coordination between them, allows the user to successfully carry out the task without detaching the tool from the workpiece, losing perpendicularity between them or leaving the intended workspace for both robot arms, see the `video` (Video of Experiment 7, 2021). The surface treatment operation is represented by the brush of the surface treatment tool cleaning a translucent liquid spilled over the surface of the workpiece.

Although the fundamental quantitative aspects of the performance of the application have already been studied in Section 6.3.1 and Section 6.3.2, there are some specific challenges which arise from the interaction between the two robots and the simultaneous teleoperation of both robot arms by the human user.

First, it must be noted that the angular movement limitation of the workpiece (WR Level 2) is still able to achieve its desired behavior even when the STR is exerting pressure in the most unfavorable area (that is, close to the workpiece limits on the side to which it is tilted). This situation is highlighted in the first and third frames in Fig. 23, and its quantitative performance can be observed in graphs depicted in Fig. 24, which show how, during the whole experiment, the angular positions of the workpiece are kept within the allowed limits.

Second, it is also remarkable that the WR movements are now arbitrary, differently to what happened in Experiment 5 and Experiment 6 in Section 6.3.2, but the tool is still kept in contact with the workpiece and perpendicular to it, as it can be seen in the detail view (lower right corner) of the second and fourth frames in Fig. 23. The same as in Experiment 5 and Experiment 6, this performance can be checked by observing in Fig. 25 how force and torque keep oscillating around their reference points (20N and 0Nm, respectively), without significant continued deviations from them. This is thanks to the control law of STR Level 2 being always active.

And third, an operation which would be virtually impossible to complete by a direct control bimanual teleoperation is correctly carried out thanks to the action of the control algorithms. In order to fully observe this, the application as a whole must be studied, but additionally to what has already been shown, it can be noted that nor the workpiece neither the surface treatment tool abandon their allowed workspace even though their teleoperation reference positions try to command them to do so, see Fig. 26 and Fig. 27, due to the WR Level 1 and STR Level 1, respectively.

Finally, information about the control actions is presented in Fig. 28 and Fig. 29, corresponding to the WR and STR control algorithms, respectively.

In Fig. 28, note that WR Level 1 and Level 2 (boundary constraint and angular position restriction) only register activity when the teleoperator tries to break those restrictions (see Fig. 26 and Fig. 24), while WR Level 3 and Level 4 (reference tracking and redundancy resolution) area always active, although the reference tracking level is noticeably more active when it is trying to follow a reference that is blocked by WR Level 1 and Level 2, because the position error cannot be overcome.

In Fig. 29, observe that STR Level 2 (F/T control) is always active because it is a conventional SMC, see Section 3.1 and Eq. (21), and STR Level 3 (reference tracking) is also active, since it is a continuous control. However, STR Level 1 only activates when the teleoperator tries to surpass the boundary, see Fig. 27.

7. Conclusions

A solution to perform surface treatment tasks has been developed in this work using a bimanual robotic system, i.e., two robot arms cooperatively performing the task. In particular, one robot arm holds the workpiece while the other robot arm has the treatment tool attached to its end-effector. In order to properly accomplish the surface treatment tasks, some robot coordinates were teleoperated by the human user, while the remaining robot coordinates were automatically controlled.

Furthermore, to assist the human user during the teleoperation, several constraints were defined for both robot arms in order to avoid exceeding the allowed workpsace. In

particular, a boundary constraint was defined for each robot arm, as well as maximum orientation angles were considered for the robot arm that holds the workpiece.

A distinguishing feature of the bimanual robot control developed in this work is that not only conventional but also a one-side sliding mode control was used.

Furthermore, the feasibility and effectiveness of the method were shown through experimental results using two robot arms: a 6R industrial manipulator and a 7R cobot.

It is interesting to remark that, during the experimentation, the user found it difficult to teleoperate the robot arms, mainly because it was difficult for the user to figure out, in real-time, the spatial correspondence between the teleoperation commands and the robot movements. Therefore, in order to improve the user teleoperation, it is suggested as further work to develop an *advanced teleoperation* system based on mixed reality (e.g., an augmented reality headset) and haptic devices (e.g. a Phantom Omni), in order to make the robot teleoperation more intuitive to the user.

Acknowledgements

This research was funded by the Spanish Government under Grant PID2020-117421RB-C21 and the Generalitat Valenciana under Grants ACIF/2019/007 and GV/2021/181.

References

Abi-Farraj, F., Pacchierotti, C., Arenz, O., Neumann, G., & Giordano, P. R. (2020). A haptic shared-control architecture for guided multi-target robotic grasping. *IEEE Transactions on Haptics*, *13*(2), 270-285.

Bandala, M., West, C., Monk, S., Montazeri, A., & Taylor, C. J. (2019). Vision-based assisted tele-operation of a dual-arm hydraulically actuated robot for pipe cutting and grasping in nuclear environments. *Robotics*, *8*(2).

Brantner, G., & Khatib, O. (2021). Controlling ocean one: Humanrobot collaboration for deep-sea manipulation. *Journal of Field Robotics*, *38*(1), 28-51.

Chatzilygeroudis, K., Fichera, B., Lauzana, I., Bu, F., Yao, K., Khadivar, F., & Billard, A. (2020). Benchmark for bimanual robotic manipulation of semi-deformable objects. *IEEE Robotics and Automation Letters*, *5*(2), 2443-2450.

Chen, H., Huang, P., & Liu, Z. (2019). Mode switching-based symmetric predictive control mechanism for networked teleoperation space robot system. *IEEE/ASME Transactions on Mechatronics*, *24*(6), 2706-2717.

Chen, H., Li, J., Wan, W., Huang, Z., & Harada, K. (2020). Integrating combined task and motion planning with compliant control. *International Journal of Intelligent Robotics and Applications*, *4*, pages149163.

Chen, Y., Zhang, S., Wu, Z., Yang, B., Luo, Q., & Xu, K. (2020). Review of surgical robotic systems for keyhole and endoscopic procedures: state of the art and perspectives. *Frontiers of Medicine*, *14*, 382403.

Chiaverini, S., Oriolo, G., & Walker, I. (2008). Kinematically redundant manipulators. *Springer Handbook of Robotics*, 245-268.

Clark, J. P., Lentini, G., Barontini, F., Catalano, M. G., Bianchi, M., & O'Malley, M. K. (2019). On the role of wearable haptics for force feedback in teleimpedance control for dual-arm robotic teleoperation. In *2019 International Conference on Robotics and Automation (ICRA)* (p. 5187-5193).

Da Silva, A., Dos Santos, D., Fernandes, A., Vilas Boas, J., & Garcia, L. (2020). High-level path planning for an autonomous sailboat robot using Q-Learning. *Sensors*, *20*(6), 1550.

Edwards, C., & Spurgeon, S. (1998). *Sliding mode control: Theory and applications* (1st ed.). UK: Taylor & Francis.

Fei, Y., Shi, P., & Lim, C.-C. (2020). Neural network adaptive dynamic sliding mode formation control of multi-agent systems. *International Journal of Systems Science*, *51*(11), 2025-2040.

García, N., Rosell, J., & Suárez, R. (2019). Motion planning by demonstration with human-likeness evaluation for dual-arm robots. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *49*(11), 2298-2307.

Garcia-Camacho, I., Lippi, M., Welle, M. C., Yin, H., Antonova, R., Varava, A., ... Kragic, D. (2020). Benchmarking bimanual cloth manipulation. *IEEE Robotics and Automation Letters*, *5*(2), 1111-1118.

Girbés-Juan, V., Schettino, V., Demiris, Y., & Tornero, J. (2021). Haptic and visual feedback assistance for dual-arm robot teleoperation in surface conditioning tasks. *IEEE Transactions on Haptics*, *14*(1), 44-56.

Golub, G., & Van Loan, C. (1996). *Matrix computations* (3rd ed.). Baltimore, MD: The Johns Hopkins University Press.

Gorjup, G., Dwivedi, A., Elangovan, N., & Liarokapis, M. (2019). An intuitive, affordances oriented telemanipulation framework for a dual robot arm hand system: On the execution of bimanual tasks. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (p. 3611-3616).

Gracia, L., Andres, J., & Tornero, J. (2009). Trajectory tracking with a 6R serial industrial robot with ordinary and non-ordinary singularities. *International Journal of Control, Automation and Systems*, *7*(1), 85-96.

Gracia, L., Sala, A., & Garelli, F. (2012). A path conditioning method with trap avoidance. *Robotics and Autonomous Systems*, *60*(6), 862-873.

Ibarguren, A., Eimontaite, I., Outn, J. L., & Fletcher, S. (2020). Dual arm co-manipulation architecture with enhanced humanrobot communication for large part manipulation. *Sensors*, *20*(21).

Isop, W. A., Gebhardt, C., Ngeli, T., Fraundorfer, F., Hilliges, O., & Schmalstieg, D. (2019). High-level teleoperation system for aerial exploration of indoor environments. *Frontiers in Robotics and AI*, *6*, 95.

Johnson, M., & Vera, A. (2019, Mar.). No ai is an island: The case for teaming intelligence. *AI Magazine*, *40*(1), 16-28.

Joshi, R. P., Tarapure, J. P., & Shibata, T. (2020). Electric wheelchair-humanoid robot collaboration for clothing assistance of the elderly. In *2020 13th International Conference on Human System Interaction (HSI)* (p. 300-306).

Kapoor, A., Li, M., & Taylor, R. H. (2005). Spatial motion constraints for robot assisted suturing using virtual fixtures. In J. S. Duncan & G. Gerig (Eds.), *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2005* (pp. 89–96). Berlin, Heidelberg: Springer Berlin Heidelberg.

Khan, A. H., & Li, S. (2020). Sliding mode control with PID sliding surface for active vibration damping of pneumatically actuated soft robots. *IEEE Access*, *8*, 88793–88800.

Khan, A. H., Li, S., Chen, D., & Liao, L. (2020). Tracking control of redundant mobile manipulator: An RRNN based metaheuristic approach. *Neurocomputing*, *400*, 272-284.

Kieselbach, K. K., Nöthen, M., & Heuer, H. (2019). Development of a visual inspection system and the corresponding algorithm for the detection and subsequent classification of paint defects on car bodies in the automotive industry. *Journal of Coatings Technology and Research*, *16*, 10331042.

Kono, H., Mori, T., Ji, Y., Fujii, H., & Suzuki, T. (2019). Development of perilous environment estimation system using a teleoperated rescue robot with on-board lidar. In *2019 IEEE/SICE International Symposium on System Integration (SII)* (p. 7-10).

Levant, A. (2003). Higher-order sliding modes, differentiation and output-feedback control.

Int. Journal of Control, *76*(9-10), 924-941.

Levant, A. (2005). Quasi-continuous high-order sliding-mode controllers. *IEEE Transactions on Automatic Control*, *50*(11), 1812-1816.

Li, Y., Guo, S., & Mukai, T. (2019). Position adjustment control of a nursing-care robot holding a patient in its arms. In *2019 IEEE International Conference on Mechatronics and Automation (ICMA)* (p. 976-981).

Li, Z., Li, C., Li, S., & Cao, X. (2019). A fault-tolerant method for motion planning of industrial redundant manipulator. *IEEE Transactions on Industrial Informatics*, *16*(12), 7469-7478.

Li, Z., & Li, S. (2020). A sparse optimization based control method for manipulator with simultaneous potential energy minimization. *IEEE Transactions on Circuits and Systems II: Express Briefs*.

Liang, J., Xu, Z., Zhou, X., Li, S., & Ye, G. (2020). Recurrent neural networks-based collision-free motion planning for dual manipulators under multiple constraints. *IEEE Access*, *8*, 54225-54236.

Liu, Y., Su, W., Li, Z., Shi, G., Chu, X., Kang, Y., & Shang, W. (2019). Motor-imagery-based teleoperation of a dual-arm robot performing manipulation tasks. *IEEE Transactions on Cognitive and Developmental Systems*, *11*(3), 414-424.

Lopez, E., Zollo, L., & Guglielmelli, E. (2013). Teleoperated control based on virtual fixtures for a redundant surgical system. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems* (p. 450-455).

Lu, Z., Huang, P., & Liu, Z. (2018). Predictive approach for sensorless bimanual teleoperation under random time delays with adaptive fuzzy control. *IEEE Transactions on Industrial Electronics*, *65*(3), 2439-2448.

Lv, H., Yang, G., Zhou, H., Huang, X., Yang, H., & Pang, Z. (2020). Teleoperation of collaborative robot for remote dementia care in home environments. *IEEE Journal of Translational Engineering in Health and Medicine*, *8*, 1-10.

Makris, S., Tsarouchi, P., Matthaiakis, A.-S., Athanasatos, A., Chatzigeorgiou, X., Stefos, M., ... Aivaliotis, S. (2017). Dual arm robot in cooperation with humans for flexible assembly. *CIRP Annals*, *66*(1), 13-16.

McConachie, D., Dobson, A., Ruan, M., & Berenson, D. (2020). Manipulating deformable objects by interleaving prediction, planning, and control. *The International Journal of Robotics Research*, *39*(8), 957-982.

Mitash, C., Shome, R., Wen, B., Boularias, A., & Bekris, K. (2020). Task-driven perception and manipulation for constrained placement of unknown objects. *IEEE Robotics and Automation Letters*, *5*(4), 5605-5612.

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

Nicolis, D., Palumbo, M., Zanchettin, A. M., & Rocco, P. (2018). Occlusion-free visual servoing for the shared autonomy teleoperation of dual-arm robots. *IEEE Robotics and Automation Letters*, *3*(2), 796-803.

Niemeyer, G., Preusche, C., Stramigioli, S., & Lee, D. (2016). Telerobotics. In B. Siciliano & O. Khatib (Eds.), *Springer handbook of robotics* (pp. 1085–1108). Cham: Springer International Publishing.

Parigi Polverini, M., Zanchettin, A. M., & Rocco, P. (2019). A constraint-based programming approach for robotic assembly skills implementation. *Robotics and Computer-Integrated Manufacturing*, *59*, 69-81.

Park, S.-O., Lee, M. C., & Kim, J. (2020). Trajectory planning with collision avoidance for redundant robots using jacobian and artificial potential field-based real-time inverse kinematics. *International Journal of Control, Automation and Systems*, *18*(8), 2095-2107.

Qu, J., Zhang, F., Wang, Y., & Fu, Y. (2019). Human-like coordination motion learning for a redundant dual-arm robot. *Robotics and Computer-Integrated Manufacturing*, *57*, 379-390.

Rakita, D., Mutlu, B., Gleicher, M., & Hiatt, L. M. (2019). Shared control–based bimanual robot manipulation. *Science Robotics*, *4*(30).

Salehian, S. S. M., Figueroa, N., & Billard, A. (2018). A unified framework for coordinated multi-arm motion planning. *The International Journal of Robotics Research*, *37*(10), 1205-1232.

Saracino, A., Oude-Vrielink, T. J. C., Menciassi, A., Sinibaldi, E., & Mylonas, G. P. (2020). Haptic intracorporeal palpation using a cable-driven parallel robot: A user study. *IEEE Transactions on Biomedical Engineering*, *67*(12), 3452-3463.

Selvaggio, M., Abi-Farraj, F., Pacchierotti, C., Giordano, P. R., & Siciliano, B. (2018). Haptic-based shared-control methods for a dual-arm system. *IEEE Robotics and Automation Letters*, *3*(4), 4249-4256.

Selvaggio, M., Ghalamzan, A., Moccia, R., Ficuciello, F., & Siciliano, B. (2019, 07). Haptic-guided shared control for needle grasping optimization in minimally invasive robotic surgery. In *IEEE/RSJ International Conference on Intelligent Robots and Systems.*

Sepúlveda, D., Fernández, R., Navas, E., Armada, M., & González-De-Santos, P. (2020). Robotic aubergine harvesting using dual-arm manipulation. *IEEE Access*, *8*, 121889-121904.

Siciliano, B., & Khatib, O. E. (2008). *Handbook of robotics.* London, UK: Springer-Verlag.

Siciliano, B., Sciavicco, L., Villani, L., & Oriolo, G. (2009). *Robotics: Modelling, planning and control.* London, UK: Springer-Verlag.

Sintov, A., Macenski, S., Borum, A., & Bretl, T. (2020). Motion planning for dual-arm manipulation of elastic rods. *IEEE Robotics and Automation Letters*, *5*(4), 6065-6072.

Sivev, S., Coleman, J., Omerdi, E., Dooly, G., & Toal, D. (2018). Underwater manipulators: A review. *Ocean Engineering*, *163*, 431-450.

Smith, C., Karayiannidis, Y., Nalpantidis, L., Gratal, X., Qi, P., Dimarogonas, D. V., & Kragic, D. (2012). Dual arm manipulationa survey. *Robotics and Autonomous Systems*, *60*(10), 1340-1353.

Solanes, J., Muñoz, A., Gracia, L., Martí, A., Girbés-Juan, V., & Tornero, J. (2020). Teleoperation of industrial robot manipulators based on augmented reality. *The International Journal of Advanced Manufacturing Technology*, *111*, 1077-1097.

Su, Y., & Zheng, C. (2020). A new nonsingular integral terminal sliding mode control for robot manipulators. *International Journal of Systems Science*, *51*(8), 1418-1428.

Suarez, A., Real, F., Vega, V. M., Heredia, G., Rodriguez-Castaño, A., & Ollero, A. (2020). Compliant bimanual aerial manipulation: Standard and long reach configurations. *IEEE Access*, *8*, 88844-88865.

Utkin, V., Guldner, J., & Shi, J. (2009). *Sliding mode control in electro-mechanical systems* (2nd ed.). London: Taylor & Francis.

Video of Experiment 1. (2021). *https://media.upv.es/player/?id=036cf340-a731-11eb-a0b0-2fbcb59aaef7.*

Video of Experiment 2. (2021). *https://media.upv.es/player/?id=cec9b0e0-a732-11eb-a0b0-2fbcb59aaef7.*

Video of Experiment 3. (2021). *https://media.upv.es/player/?id=28a91da0-a731-11eb-a0b0-2fbcb59aaef7.*

Video of Experiment 4. (2021). *https://media.upv.es/player/?id=42de19a0-a731-11eb-a0b0-2fbcb59aaef7.*

Video of Experiment 5. (2021). *https://media.upv.es/player/?id=e3904b60-a732-11eb-a0b0-2fbcb59aaef7.*

Video of Experiment 6. (2021). *https://media.upv.es/player/?id=fa89cb70-a732-11eb-a0b0-2fbcb59aaef7.*

Video of Experiment 7. (2021). *https://media.upv.es/player/?id=15ffabe0-a733-11eb-a0b0-2fbcb59aaef7.*

Wu, Q., Li, M., Qi, X., Hu, Y., Li, B., & Zhang, J. (2019). Coordinated control of a dual-arm robot for surgical instrument sorting tasks. *Robotics and Autonomous Systems*, *112*, 1-12.

Yoon, H., Jeong, J. H., & Yi, B. (2018). Image-guided dual masterslave robotic system for maxillary sinus surgery. *IEEE Transactions on Robotics*, *34*(4), 1098-1111.

Zhang, X., Ma, H., Luo, M., & Liu, X. (2020). Adaptive sliding mode control with information

concentration estimator for a robot arm. *International Journal of Systems Science*, *51*(2), 217-228.

Zhong, F., Wang, Y., Wang, Z., & Liu, Y. (2019). Dual-arm robotic needle insertion with active tissue deformation for autonomous suturing. *IEEE Robotics and Automation Letters*, *4*(3), 2669-2676.

Zimmermann, S., Hakimifard, G., Zamora, M., Poranne, R., & Coros, S. (2020). A multi-level optimization framework for simultaneous grasping and motion planning. *IEEE Robotics and Automation Letters*, *5*(2), 2966-2972.

Table 1. Control algorithm for the WR

Algorithm executed at sampling time of T_w seconds
$[\mathbf{q}_w, \dot{\mathbf{q}}_w]$ =`GetWRState();`
$\mathbf{p}_{w,ref}$ =`GetUserReferenceWR();`
$\mathbf{p}_w = \mathbf{l}_w(\mathbf{q}_w)$; `// Eq. (1)`
$\dot{\mathbf{p}}_w = \mathbf{J}_w \dot{\mathbf{q}}_w$; `// Eq. (2)`
$\dot{\mathbf{p}}_{w,ref} = (\mathbf{p}_{w,ref} - \mathbf{p}_{w,ref,prev})/T_w$; `// Derivative`
$\ddot{\mathbf{p}}_{w,ref} = (\dot{\mathbf{p}}_{w,ref} - \dot{\mathbf{p}}_{w,ref,prev})/T_w$; `// Derivative`
$\overline{\mathbf{p}}_{wb} = {}^b\mathbf{R}_w \left(\begin{bmatrix} \mathbf{I}_3 & \mathbf{O}_3 \end{bmatrix} \mathbf{p}_w - \overline{\mathbf{p}}_{wc} \right)$; `// Eq. (31)`
$\phi_{w1} = -1 + \left\lvert \frac{x_{wb}}{W} \right\rvert^m + \left\lvert \frac{y_{wb}}{H} \right\rvert^m + \left\lvert \frac{z_{wb}}{M} \right\rvert^m + K_{w1}\, \mathbf{H}_{w1}\, \mathbf{J}_w\, \dot{\mathbf{q}}_w$; `// Eq. (32)`
$\boldsymbol{\phi}_{w2} = \begin{bmatrix} \lvert \alpha_w - \alpha_{w,mid} \rvert - \alpha_{w,max} \\ \lvert \beta_w - \beta_{w,mid} \rvert - \beta_{w,max} \\ \lvert \gamma_w - \gamma_{w,mid} \rvert - \gamma_{w,max} \end{bmatrix} + \mathbf{K}_{w2}\, \mathbf{H}_{w2}\, \mathbf{J}_w\, \dot{\mathbf{q}}_w$; `// Eq. (39)`
$\mathbf{A}_{w1} = \mathrm{pos}(\phi_{w1})\, K_{w1}\, \mathbf{H}_{w1}\, \mathbf{J}_w$; `// Eq. (35)`
$b_{w1} = -\mathrm{pos}(\phi_{w1})\, u^+_{w1}$; `// Eq. (35)`
$\mathbf{A}_{w2} = \mathrm{v2dm}(\mathrm{pos}(\boldsymbol{\phi}_{w2}))\, \mathbf{K}_{w2}\, \mathbf{H}_{w2}\, \mathbf{J}_w$; `// Eq. (42)`
$\mathbf{b}_{w2} = -\mathbf{W}_{w2}\, \mathrm{pos}(\boldsymbol{\phi}_{w2})\, u^+_{w2}$; `// Eq. (42)`
$\mathbf{A}_{w3} = \mathbf{J}_w$; `// Eq. (43)`
$\mathbf{b}_{w3} = \ddot{\mathbf{p}}_{w,ref} + K_{w3,v}\, \dot{\mathbf{e}}_w + K_{w3,p}\, \mathbf{e}_w + \mathrm{sign}\left(\dot{\mathbf{e}}_w + \frac{K_{w3,p}}{K_{w3,v}} \mathbf{e}_w \right) \mathbf{u}^+_{w3}$; `// Eq. (43)`
$\mathbf{A}_{w4} = \mathbf{I}$; `// Eq. (44)`
$\mathbf{b}_{w4} = -K_{w4,v}\, \dot{\mathbf{q}}_w + K_{w4,p}(\mathbf{q}_{w0} - \mathbf{q}_w)$; `// Eq. (44)`
$\ddot{\mathbf{q}}_{wc,1} = \mathbf{A}^\dagger_{w1} b_{w1}$; `// Eq. (5), i = 1`
$\mathbf{N}_{w1} = \mathbf{I} - \mathbf{A}^\dagger_{w1} \mathbf{A}_{w1}$; `// Eq. (6), i = 1`
$\ddot{\mathbf{q}}_{wc,2} = \ddot{\mathbf{q}}_{wc,1} + (\mathbf{A}_{w2}\mathbf{N}_{w1})^\dagger (\mathbf{b}_{w2} - \mathbf{A}_{w2}\ddot{\mathbf{q}}_{wc,1})$; `// Eq. (5), i = 2`
$\mathbf{N}_{w2} = \mathbf{N}_{w1}(\mathbf{I} - (\mathbf{A}_{w2}\mathbf{N}_{w1})^\dagger (\mathbf{A}_{w2}\mathbf{N}_{w1}))$; `// Eq. (6), i = 2`
$\ddot{\mathbf{q}}_{wc,3} = \ddot{\mathbf{q}}_{wc,2} + (\mathbf{A}_{w3}\mathbf{N}_{w2})^\dagger (\mathbf{b}_{w3} - \mathbf{A}_{w3}\ddot{\mathbf{q}}_{wc,2})$; `// Eq. (5), i = 3`
$\mathbf{N}_{w3} = \mathbf{N}_{w2}(\mathbf{I} - (\mathbf{A}_{w3}\mathbf{N}_{w2})^\dagger (\mathbf{A}_{w3}\mathbf{N}_{w2}))$; `// Eq. (6), i = 3`
$\ddot{\mathbf{q}}_{wc,4} = \ddot{\mathbf{q}}_{wc,3} + (\mathbf{A}_{w4}\mathbf{N}_{w3})^\dagger (\mathbf{b}_{w4} - \mathbf{A}_{w4}\ddot{\mathbf{q}}_{wc,3})$; `// Eq. (5), i = 4`
$\dot{\mathbf{q}}_{wc} = \ddot{\mathbf{q}}_{wc,4}\, T_w + \dot{\mathbf{q}}_{wc,prev}$; `// Integration`
$\mathbf{q}_{wc} = \dot{\mathbf{q}}_{wc}\, T_w + \mathbf{q}_{wc,prev}$; `// Integration`
`SendToWRJointControllers(`$\mathbf{q}_{wc}$`);`
$\mathbf{q}_{wc,prev} = \mathbf{q}_{wc}$; `// For next iteration`
$\dot{\mathbf{q}}_{wc,prev} = \dot{\mathbf{q}}_{wc}$; `// For next iteration`
$\mathbf{p}_{w,ref,prev} = \mathbf{p}_{w,ref}$; `// For next iteration`
$\dot{\mathbf{p}}_{w,ref,prev} = \dot{\mathbf{p}}_{w,ref}$; `// For next iteration`

Table 2. Control algorithm for the STR

Algorithm executed at sampling time of T_s seconds	
$[\mathbf{q}_s, \dot{\mathbf{q}}_s, \mathbf{F}]$ =GetSTRStateAndForces();	
$\overline{\mathbf{p}}_{sb,ref}$ =GetUserReferenceSTR();	
$\mathbf{p}_w$ =GetWRPose();	
$[\overline{\mathbf{p}}_{sc},\ {}^b\mathbf{R}_s]$ =PoseToPositionAndRot($\mathbf{p}_w$);	
$\mathbf{p}_s = \mathbf{l}_s(\mathbf{q}_s)$;	// Eq. (1)
$\dot{\mathbf{p}}_s = \mathbf{J}_s \dot{\mathbf{q}}_s$;	// Eq. (2)
$\overline{\mathbf{p}}_{sb} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} {}^b\mathbf{R}_s \left(\begin{bmatrix} \mathbf{I}_3 & \mathbf{O}_3 \end{bmatrix} \mathbf{p}_s - \overline{\mathbf{p}}_{sc} \right)$;	// Eq. (46)
$\dot{\mathbf{F}} = (\mathbf{F} - \mathbf{F}_{prev})/T_s$;	// Derivative
$\dot{\overline{\mathbf{p}}}_{sb} = (\overline{\mathbf{p}}_{sb} - \overline{\mathbf{p}}_{sb,prev})/T_s$;	// Derivative
$\dot{\overline{\mathbf{p}}}_{sb,ref} = (\overline{\mathbf{p}}_{sb,ref} - \overline{\mathbf{p}}_{sb,ref,prev})/T_s$;	// Derivative
$\ddot{\overline{\mathbf{p}}}_{sb,ref} = (\dot{\overline{\mathbf{p}}}_{sb,ref} - \dot{\overline{\mathbf{p}}}_{sb,ref,prev})/T_s$;	// Derivative
$\sigma_{s1} = -1 + \left\lvert \frac{x_{sb}}{W} \right\rvert^m + \left(\frac{\max(\lvert y_{sb} \rvert - (H - W), 0)}{W} \right)^m$;	// Eq. (45)
$\phi_{s1} = \sigma_{s1} + K_{s1} \left(\sigma_{s1} - \sigma_{s1,prev} \right)/T_s$;	// Eq. (47)
$\boldsymbol{\phi}_{s2} = \begin{bmatrix} F_z - F_{z,ref} + K_{s2,z}\, \dot{F}_z \\ F_\alpha + K_{s2,\alpha}\, \dot{F}_\alpha \\ F_\beta + K_{s2,\beta}\, \dot{F}_\beta \end{bmatrix}$;	// Eq. (54)
$\mathbf{A}_{s1} = \mathrm{pos}(\phi_{s1})\, K_{s1}\, \mathbf{H}_{s1}\, \mathbf{J}_s$;	// Eq. (50)
$b_{s1} = -\mathrm{pos}(\phi_{s1})\, u^+_{s1}$;	// Eq. (50)
$\mathbf{A}_{s2} = \mathbf{K}_{s2}\, \mathbf{H}_{s2}\, \mathbf{J}_{sn}$;	// Eq. (56)
$\mathbf{b}_{s2} = -\overline{\mathbf{W}}_{s2}\, \mathrm{sign}(\boldsymbol{\phi}_{s2})\, u^+_{s2}$;	// Eq. (56)
$\mathbf{A}_{s3} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} {}^b\mathbf{R}_s & \mathbf{O}_3 \end{bmatrix} \mathbf{J}_s$;	// Eqs. (58) and (59)
$\mathbf{b}_{s3} = \ddot{\overline{\mathbf{p}}}_{sb,ref} + K_{s3,v}\, \dot{\overline{\mathbf{e}}}_{sb} + K_{s3,p}\, \overline{\mathbf{e}}_{sb} + \mathrm{sign}\left(\dot{\overline{\mathbf{e}}}_{sb} + \frac{K_{s3,p}}{K_{s3,v}} \overline{\mathbf{e}}_{sb} \right) u^+_{s3}$;	// Eq. (58)
$\ddot{\mathbf{q}}_{sc,1} = \mathbf{A}^\dagger_{s1} b_{s1}$;	// Eq. (5), $i = 1$
$\mathbf{N}_{s1} = \mathbf{I} - \mathbf{A}^\dagger_{s1} \mathbf{A}_{s1}$;	// Eq. (6), $i = 1$
$\ddot{\mathbf{q}}_{sc,2} = \ddot{\mathbf{q}}_{sc,1} + (\mathbf{A}_{s2}\mathbf{N}_{s1})^\dagger (\mathbf{b}_{s2} - \mathbf{A}_{s2} \ddot{\mathbf{q}}_{sc,1})$;	// Eq. (5), $i = 2$
$\mathbf{N}_{s2} = \mathbf{N}_{s1} (\mathbf{I} - (\mathbf{A}_{s2}\mathbf{N}_{s1})^\dagger (\mathbf{A}_{s2}\mathbf{N}_{s1}))$;	// Eq. (6), $i = 2$
$\ddot{\mathbf{q}}_{sc,3} = \ddot{\mathbf{q}}_{sc,2} + (\mathbf{A}_{s3}\mathbf{N}_{s2})^\dagger (\mathbf{b}_{s3} - \mathbf{A}_{s3} \ddot{\mathbf{q}}_{sc,2})$;	// Eq. (5), $i = 3$
$\dot{\mathbf{q}}_{sc} = \ddot{\mathbf{q}}_{sc,3}\, T_s + \dot{\mathbf{q}}_{sc,prev}$;	// Integration
$\mathbf{q}_{sc} = \dot{\mathbf{q}}_{sc}\, T_s + \mathbf{q}_{sc,prev}$;	// Integration
SendToSTRJointControllers($\mathbf{q}_{sc}$);	
$\mathbf{q}_{sc,prev} = \mathbf{q}_{sc}$;	// For next iteration
$\dot{\mathbf{q}}_{sc,prev} = \dot{\mathbf{q}}_{sc}$;	// For next iteration
$\sigma_{s1,prev} = \sigma_{s1}$;	// For next iteration
$\mathbf{F}_{prev} = \mathbf{F}$;	// For next iteration
$\overline{\mathbf{p}}_{sb,prev} = \overline{\mathbf{p}}_{sb}$;	// For next iteration
$\overline{\mathbf{p}}_{sb,ref,prev} = \overline{\mathbf{p}}_{sb,ref}$;	// For next iteration
$\dot{\overline{\mathbf{p}}}_{sb,ref,prev} = \dot{\overline{\mathbf{p}}}_{sb,ref}$;	// For next iteration

Table 3. Denavit-Hartenberg parameters for the STR ($d_{tool} = 0.06$)

Link i	θ_i(rad)	d_i(m)	a_i(m)	α_i(rad)
1	q_1	-0.4	0.025	$\pi/2$
2	q_2	0	0.455	0
3	$q_3 - \pi/2$	0	0.035	$\pi/2$
4	q_4	-0.42	0	$-\pi/2$
5	q_5	0	0	$\pi/2$
6	q_6	$-0.08 - d_{tool}$	0	π

Table 4. Denavit-Hartenberg parameters for the WR ($d_{tool} = 0.06$)

Link i	θ_i(rad)	d_i(m)	a_i(m)	α_i(rad)
1	$q_1 + \pi$	0.36	0	$\pi/2$
2	$q_2 + \pi$	0	0	$\pi/2$
3	q_3	0.42	0	$\pi/2$
4	$q_4 + \pi$	0	0	$\pi/2$
5	q_5	0.4	0	$\pi/2$
6	$q_6 + \pi$	0	0	$\pi/2$
7	q_7	0.152	0	$\pi/2$

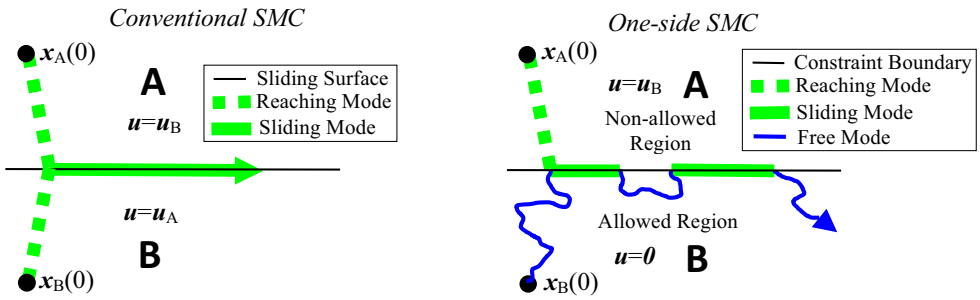


Figure 1. Graphical comparison between conventional SMC (left) and one-side SMC (right).

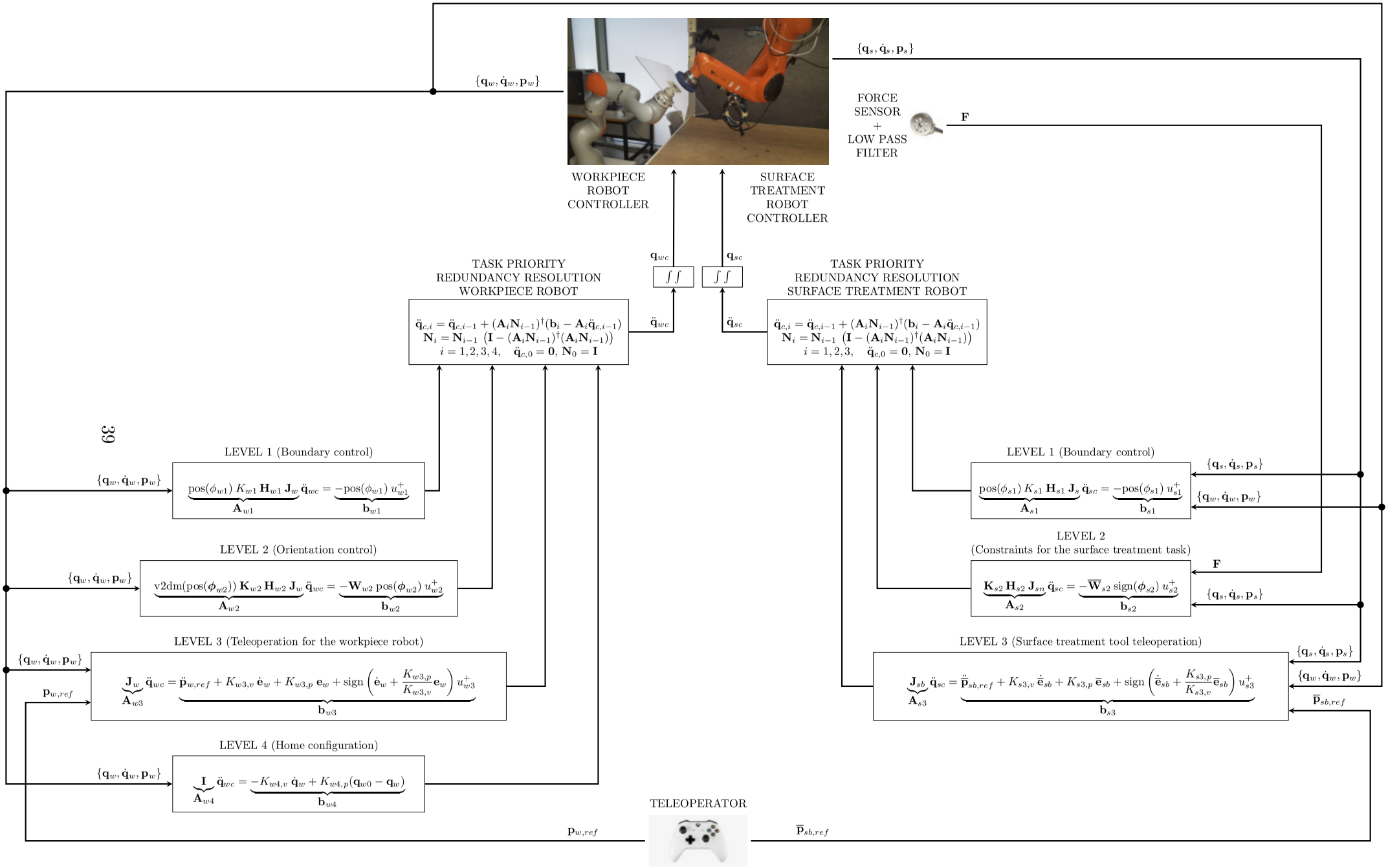


Figure 2. Block diagram of the proposed control for the WR and STR.

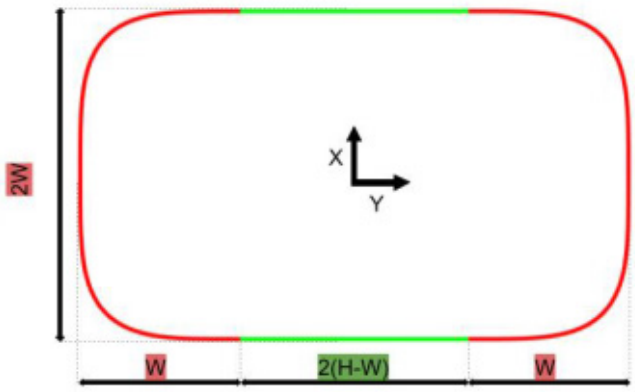


Figure 3. Modified superellipse proposed in this work, which is composed of a $2W \times 2(H-W)$ rectangle and two offseted halfs of an even-sided $2W \times 2W$ superellipse.

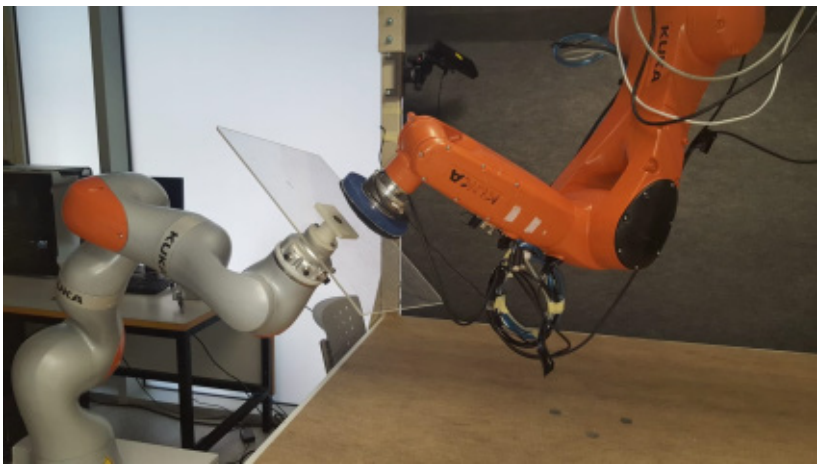


Figure 4. Experimental setup. STR: a 6R serial manipulator with an F/T sensor, a tool consisting of a cylinder (blue) and a piece of cloth attached to it (black). WR: a 7R cobot serial manipulator with a methacrylate flat workpiece attached to its end-effector by means of a self developed adaptor (white).

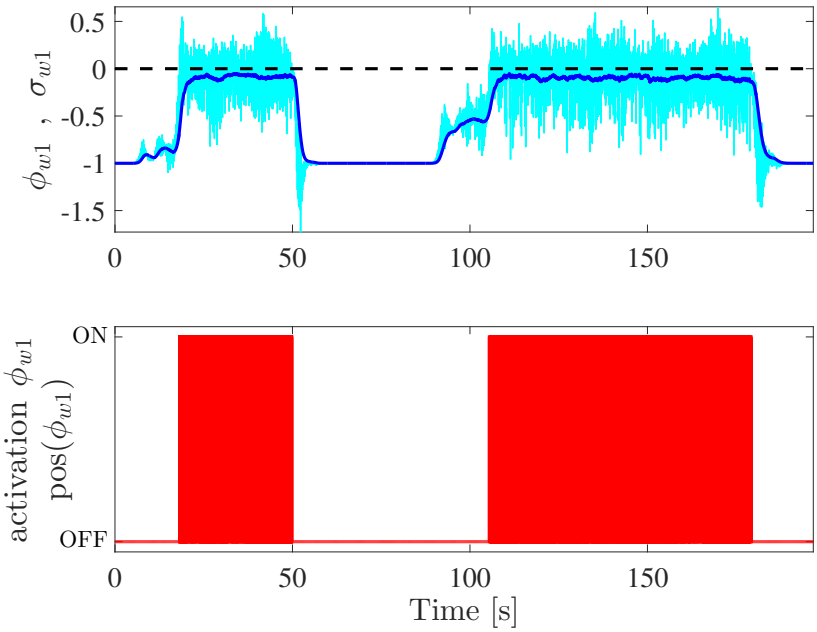


Figure 5. Experiment 1. WR Level 1: Top, constraint functions σ_{w1} (thick dark-blue) and ϕ_{w1} (thin light-cyan) and constraint limit (dashed); and bottom, constraint activation.

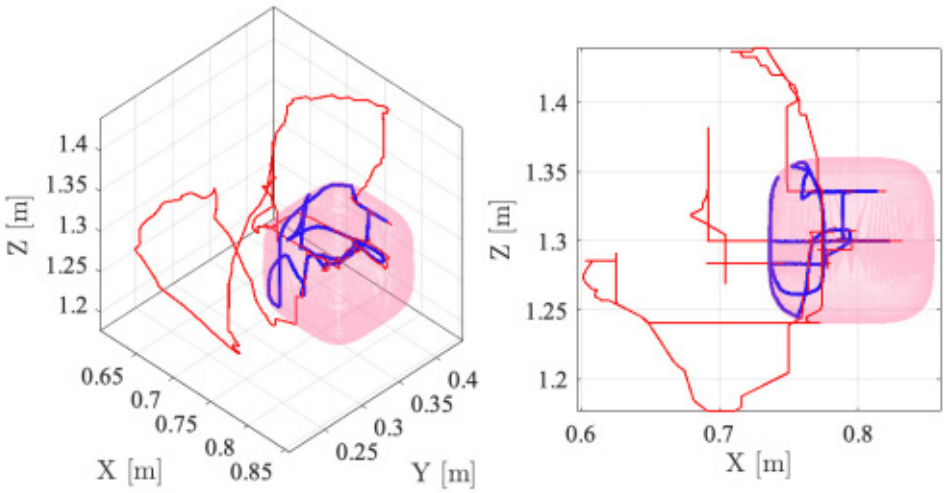


Figure 6. Experiment 1. 3D view (left) and top view (right) of the boundary constraint of WR Level 1: allowed region (pink mesh); actual position of the workpiece center (thick-blue line); and reference position for the workpiece center (thin-red line).

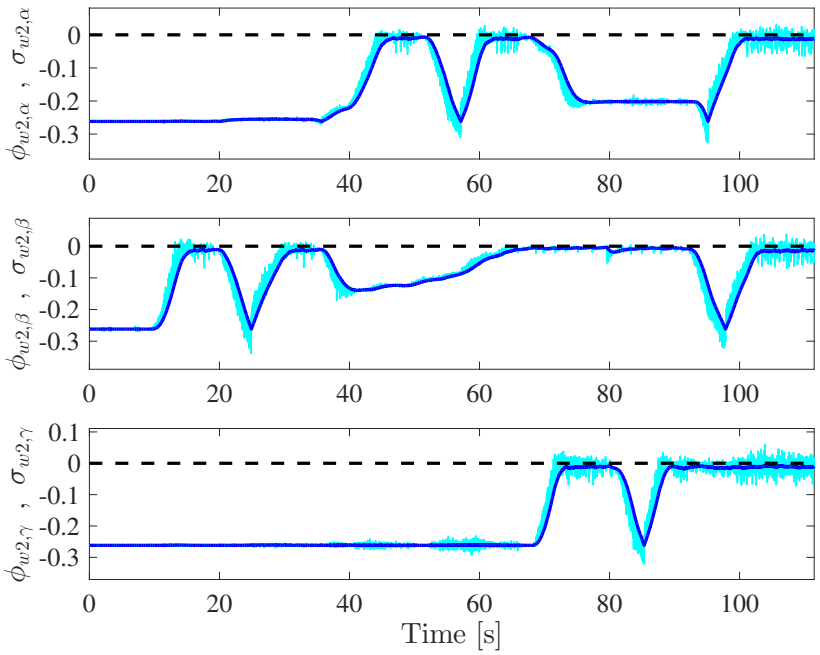


Figure 7. Experiment 2. WR Level 2: Constraint functions $\sigma_{w2,i}$ (thick dark-blue) and $\phi_{w2,i}$ (thin light-cyan) of the roll (α), pitch (β) and yaw (γ) angles of the workpiece and constraint limit (dashed).

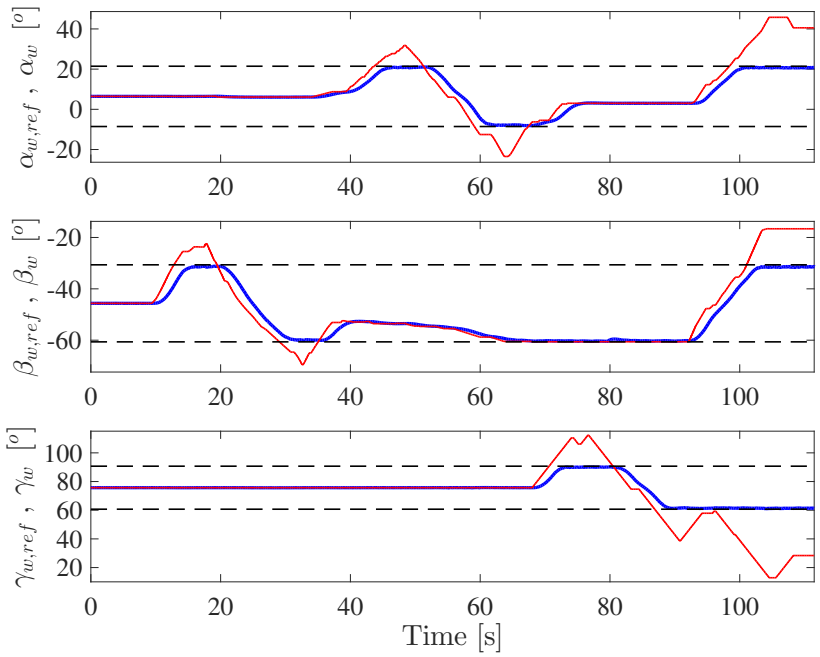


Figure 8. Experiment 2. Behavior of the restrictions of WR Level 2: angular reference (thin-red), actual angular position (thick-blue) of the workpiece and angular limits (dashed).

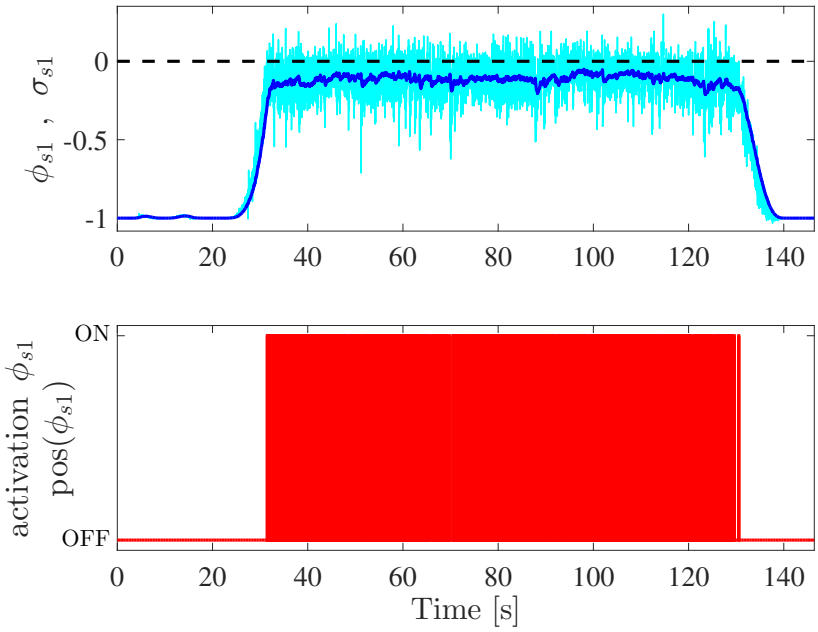


Figure 9. Experiment 3. STR Level 1: Top, constraint functions σ_{s1} (thick dark-blue) and ϕ_{s1} (thin light-cyan) and constraint limit (dashed); and bottom, constraint activation.

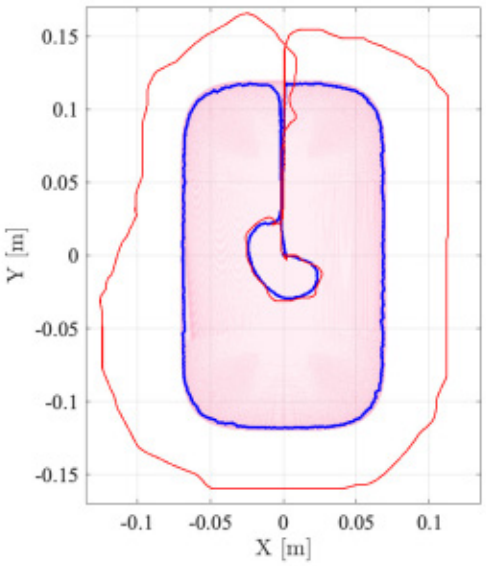
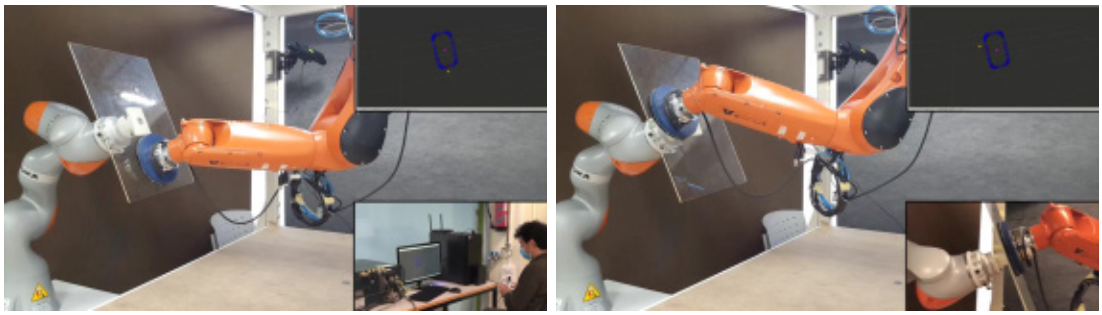
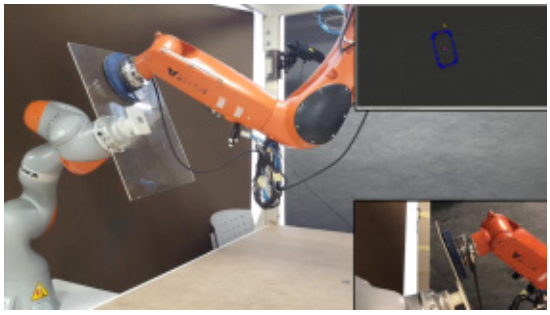


Figure 10. Experiment 3. Representation of the boundary constraint of STR Level 1: allowed region (pink mesh); actual position of the STR tool (thick-blue line); and reference position for the STR tool (thin-red line). Coordinates relative to the workpiece center.

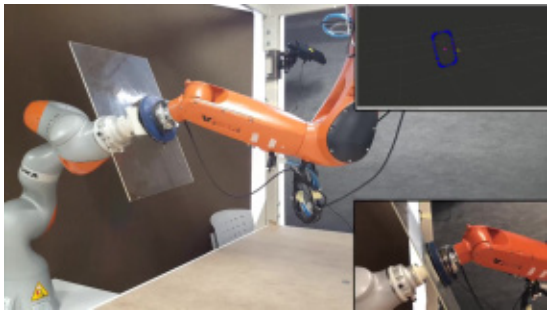


(a) video: 0m 40s

(b) video: 1m 08s



(c) video: 1m 31s



(d) video: 1m 55s

Figure 11. Frames of the video of Experiment 3.

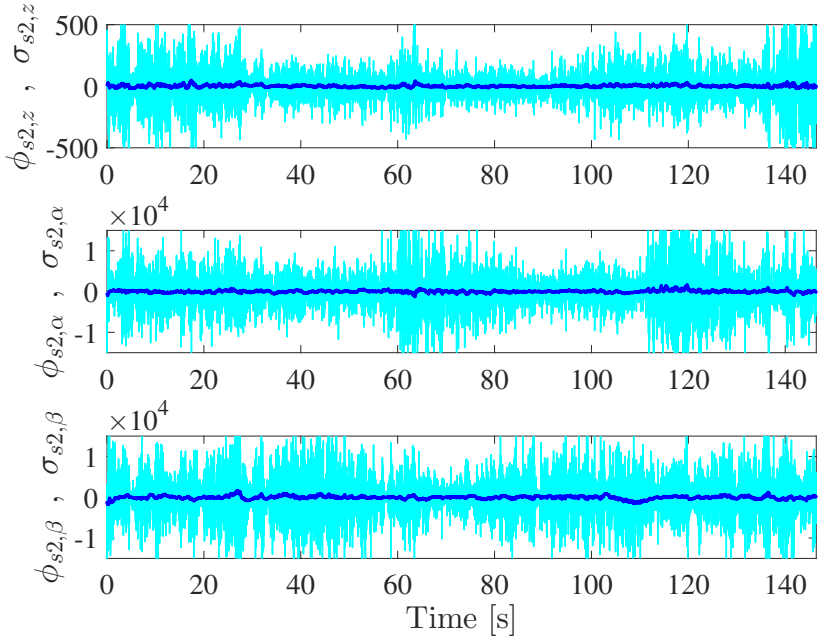
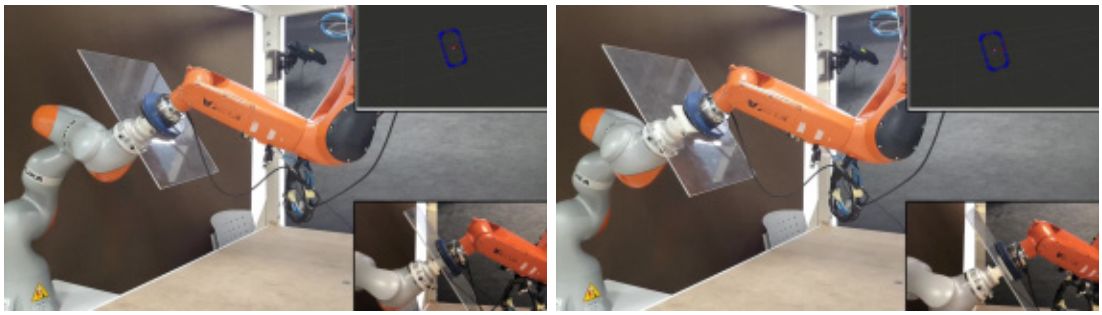
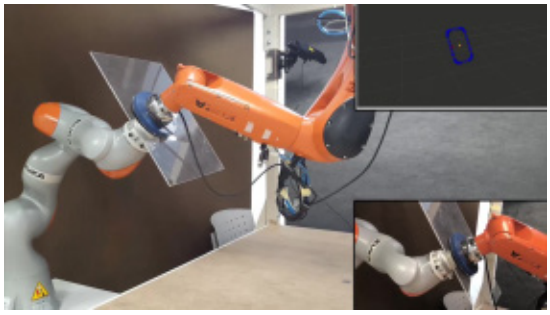


Figure 12. Experiment 3. STR Level 2: constraint functions $\sigma_{s2,i}$ (thick dark-blue) and $\phi_{s2,i}$ (light-cyan).

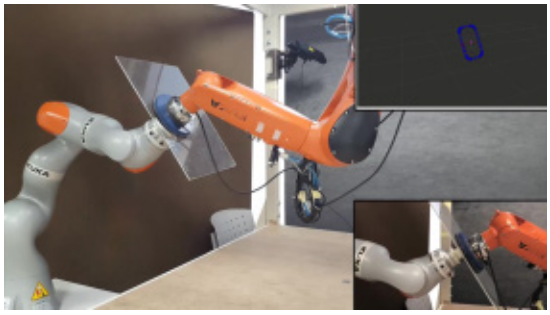


(a) video: 0m 11s

(b) video: 0m 15s



(c) video: 0m 26s



(d) video: 0m 32s

Figure 13. Frames of the video of Experiment 4.

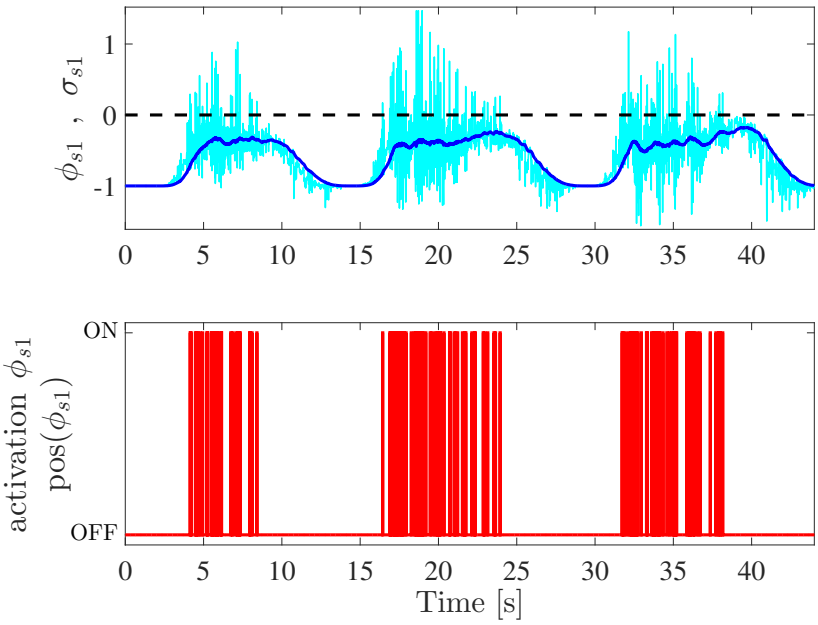


Figure 14. Experiment 4. STR Level 1: Top, constraint functions σ_{s1} (thick dark-blue) and ϕ_{s1} (thin light-cyan) and constraint limit (dashed); and bottom, constraint activation.

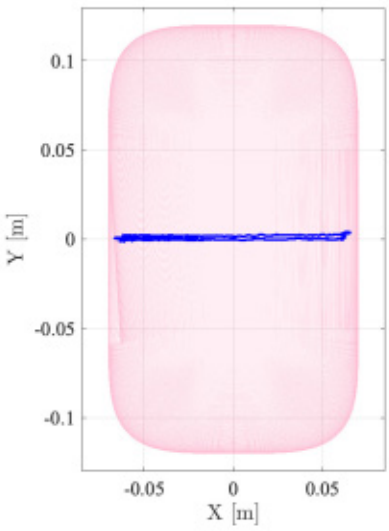
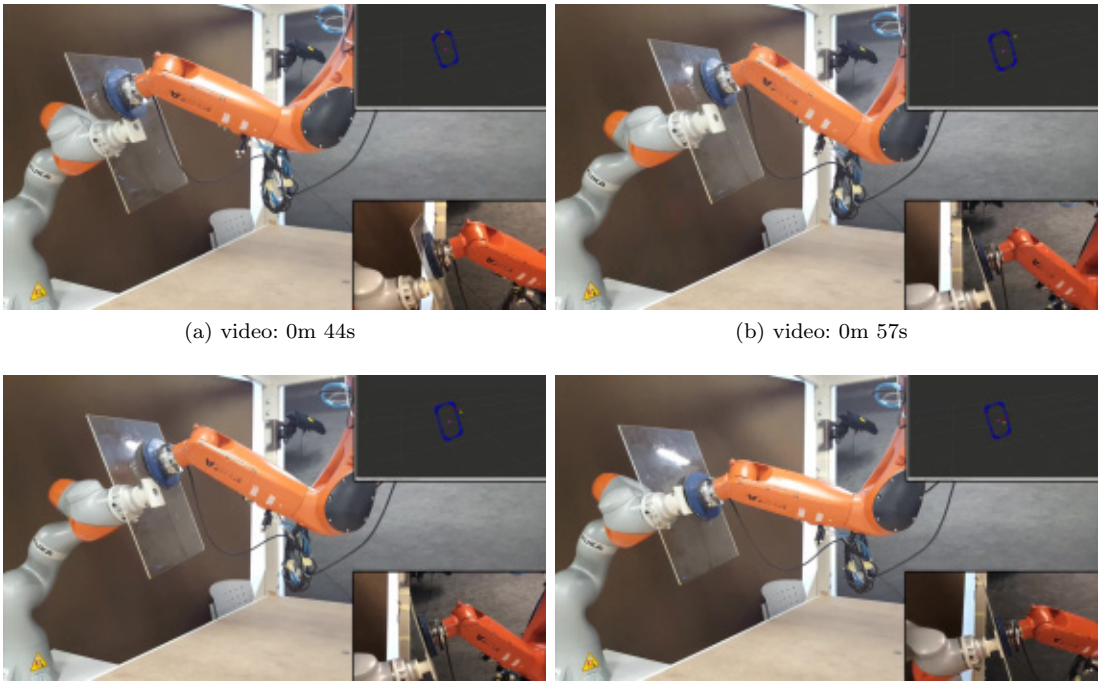


Figure 15. Experiment 4. Representation of the boundary constraint of STR Level 1: allowed region (pink mesh); and actual position of the STR tool (thick-blue line). Note that there is no reference position for the STR tool in this experiment, i.e., the STR tries to keep the treatment tool still.



(a) video: 0m 44s

(b) video: 0m 57s

(c) video: 1m 00s

(d) video: 1m 09s

Figure 16. Frames of the video of Experiment 5.

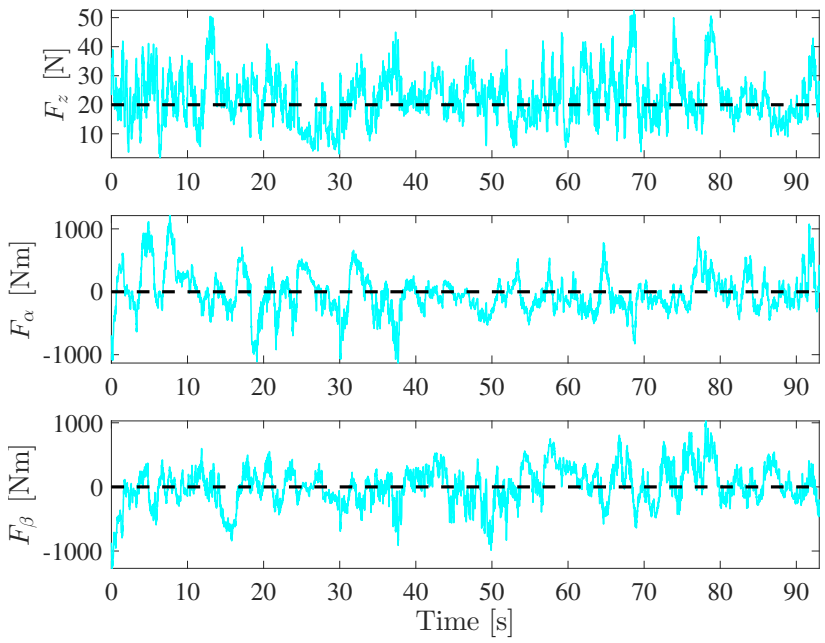


Figure 17. Experiment 5. Behavior of STR Level 2: measurements of the F/T sensor in the linear Z-axis (top), angular X-axis (middle) and angular Y-axis (bottom) of the STR end-effector frame. The reference value for each signal is represented with a dashed line.

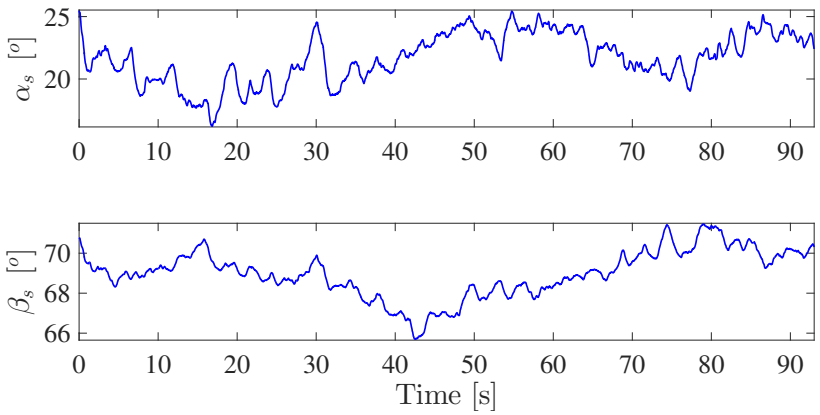


Figure 18. Experiment 5. Behavior of the restrictions of STR Level 2: roll (top) and pitch (bottom) angles of the STR tool.

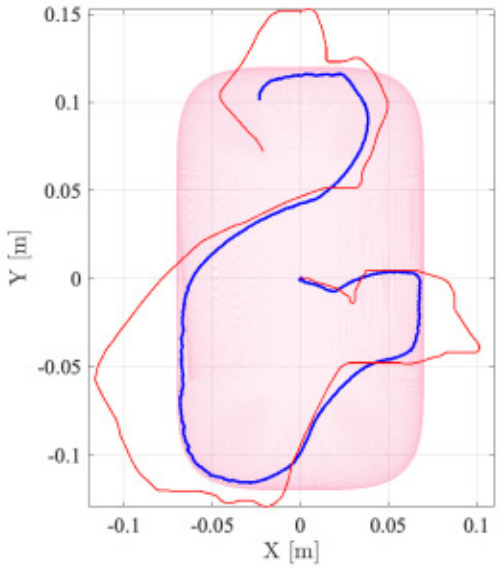
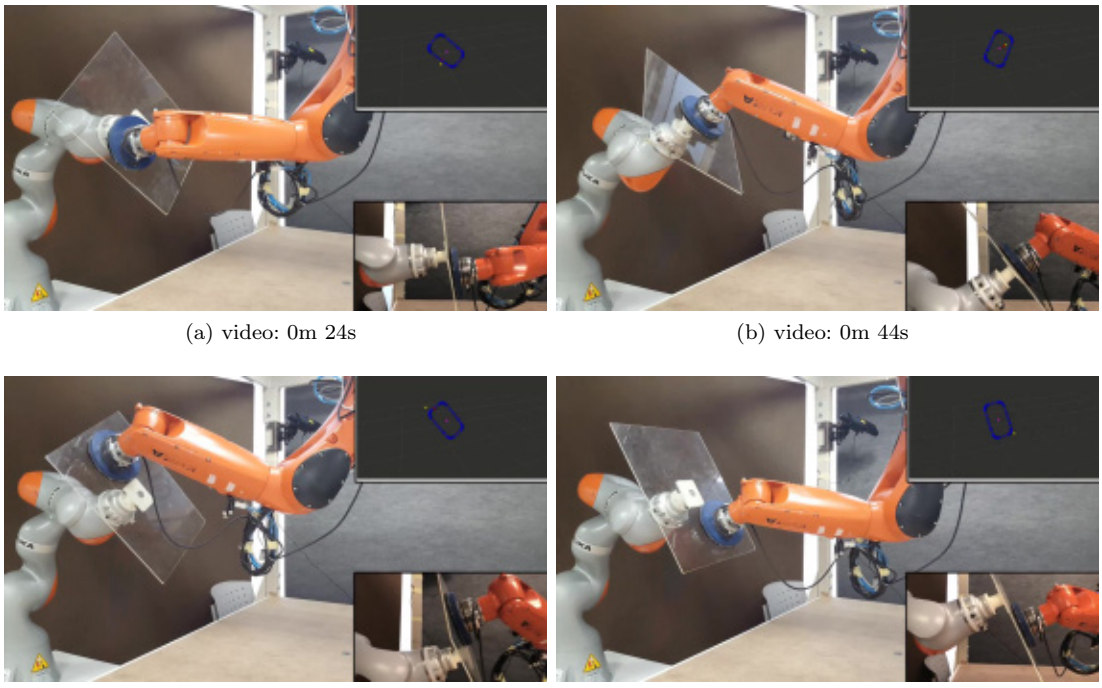


Figure 19. Experiment 5. Representation of the boundary constraint of STR Level 1: allowed region (pink mesh); actual position of the STR tool (thick-blue line); and reference position for the STR tool (thin-red line). Coordinates relative to the workpiece center.



(a) video: 0m 24s

(b) video: 0m 44s

(c) video: 1m 10s

(d) video: 1m 53s

Figure 20. Frames of the video of Experiment 6.

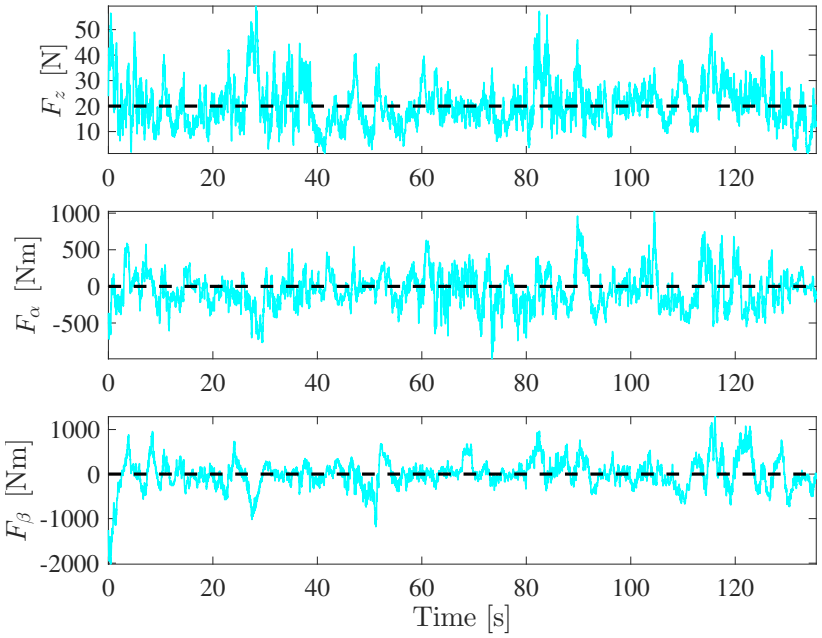


Figure 21. Experiment 6. Behavior of STR Level 2: measurements of the F/T sensor in the linear Z-axis (top), angular X-axis (middle) and angular Y-axis (bottom) of the STR end-effector frame. The reference value for each signal is represented with a dashed line.

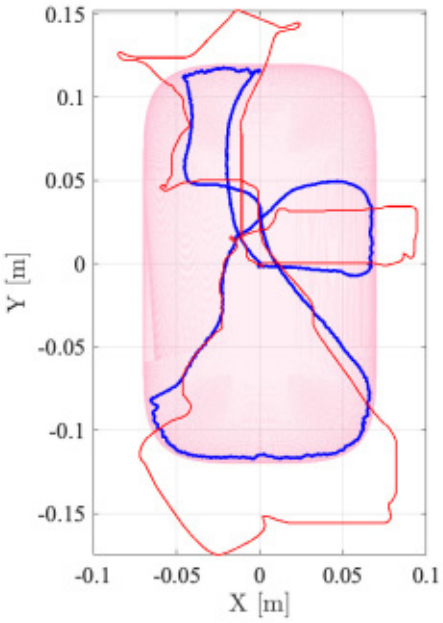
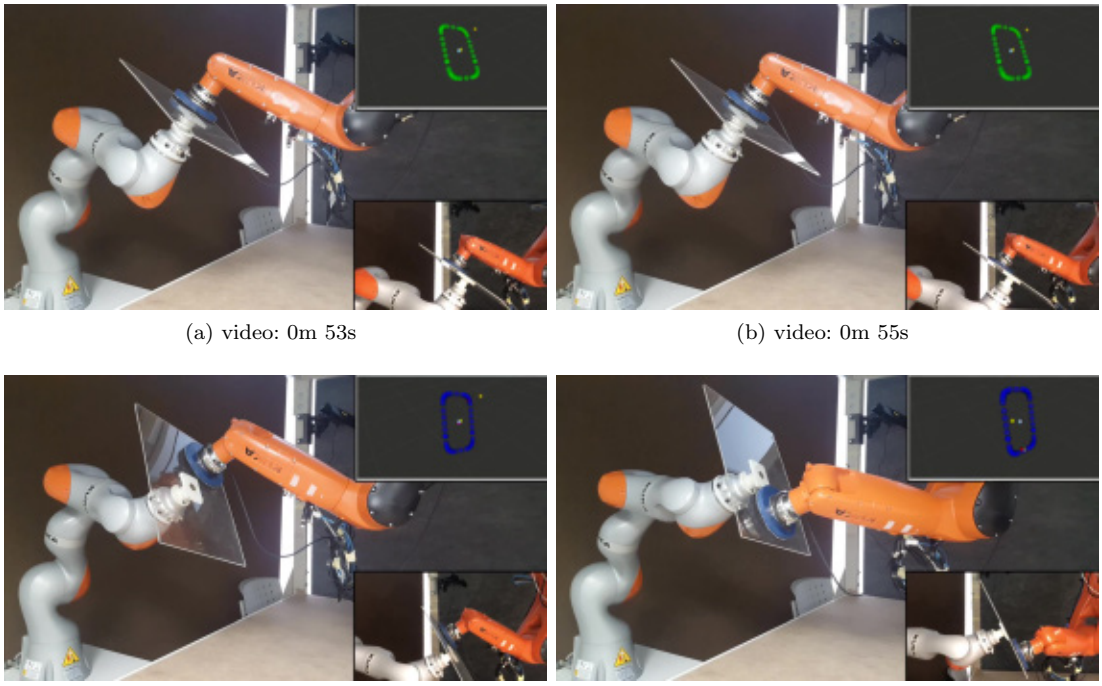


Figure 22. Experiment 6. Representation of the boundary constraint of STR Level 1: allowed region (pink mesh); actual position of the STR tool (thick-blue line); and reference position for the STR tool (thin-red line). Coordinates relative to the workpiece center.



(a) video: 0m 53s (b) video: 0m 55s

(c) video: 1m 03s (d) video: 1m 52s

Figure 23. Frames of the video of Experiment 7.

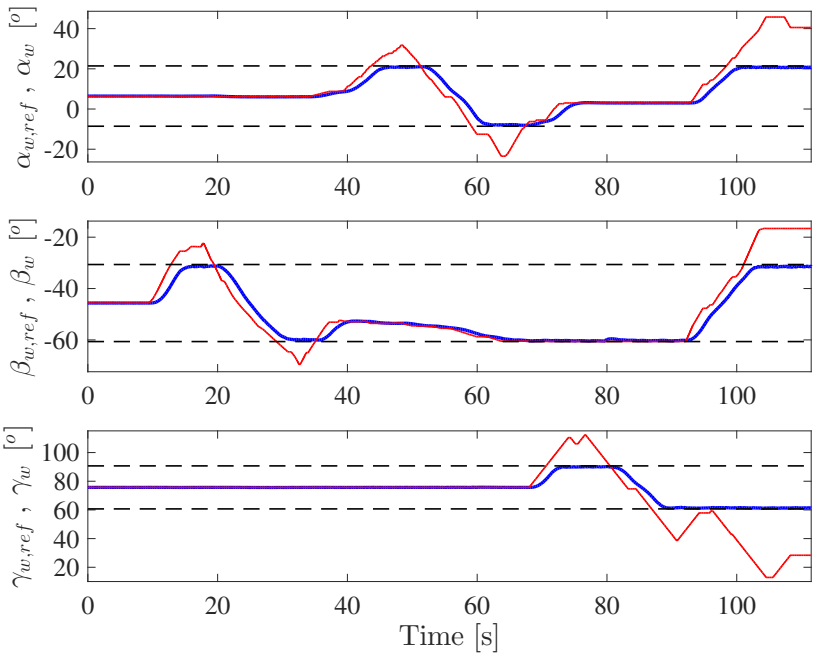


Figure 24. Experiment 7. Behavior of the restrictions of WR Level 2: angular reference (thin-red), actual angular position (thick-blue) of the workpiece and angular limits (dashed).

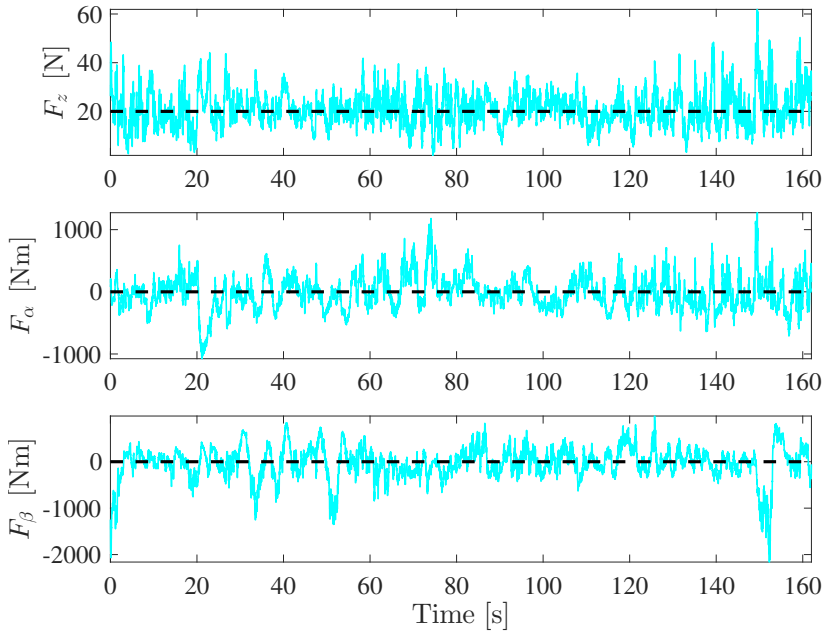


Figure 25. Experiment 7. Behavior of STR Level 2: measurements of the F/T sensor in the linear Z-axis (top), angular X-axis (middle) and angular Y-axis (bottom) of the STR end-effector frame. The reference value for each signal is represented with a dashed line.

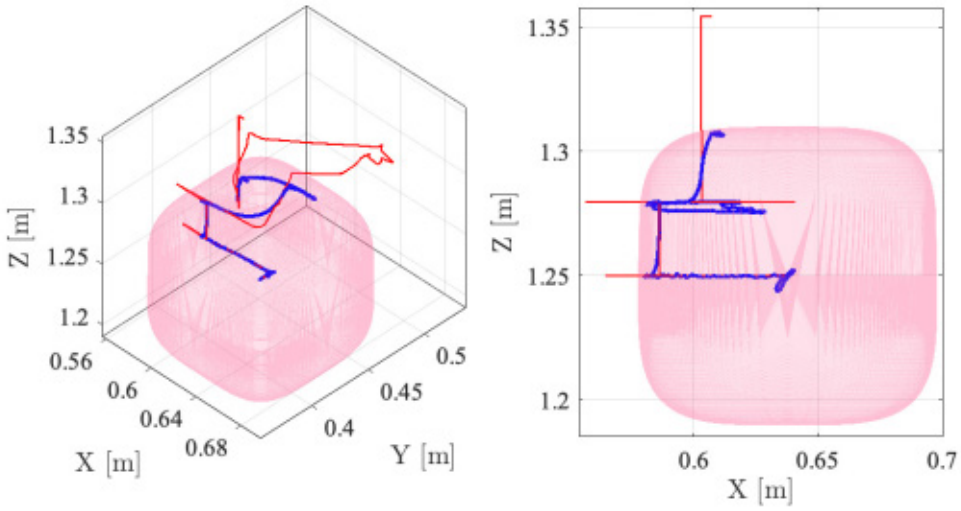


Figure 26. Experiment 7. 3D view (left) and top view (right) of the boundary constraint of WR Level 1: allowed region (pink mesh); actual position of the workpiece center (thick-blue line); and reference position for the workpiece center (thin-red line).

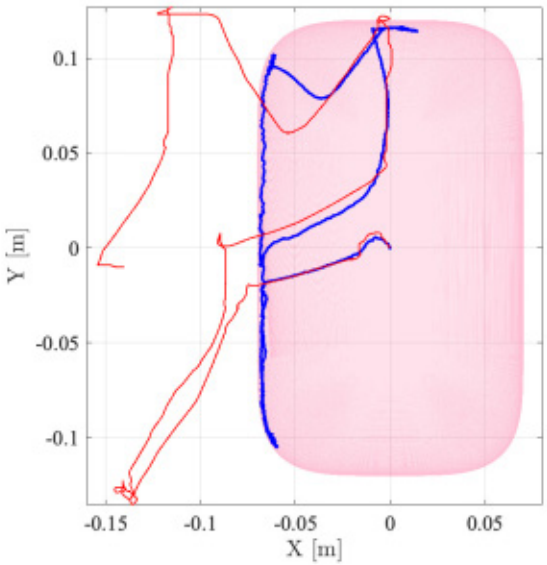


Figure 27. Experiment 7. Representation of the boundary constraint of STR Level 1: allowed region (pink mesh); actual position of the STR tool (thick-blue line); and reference position for the STR tool (thin-red line). Coordinates relative to the workpiece center.

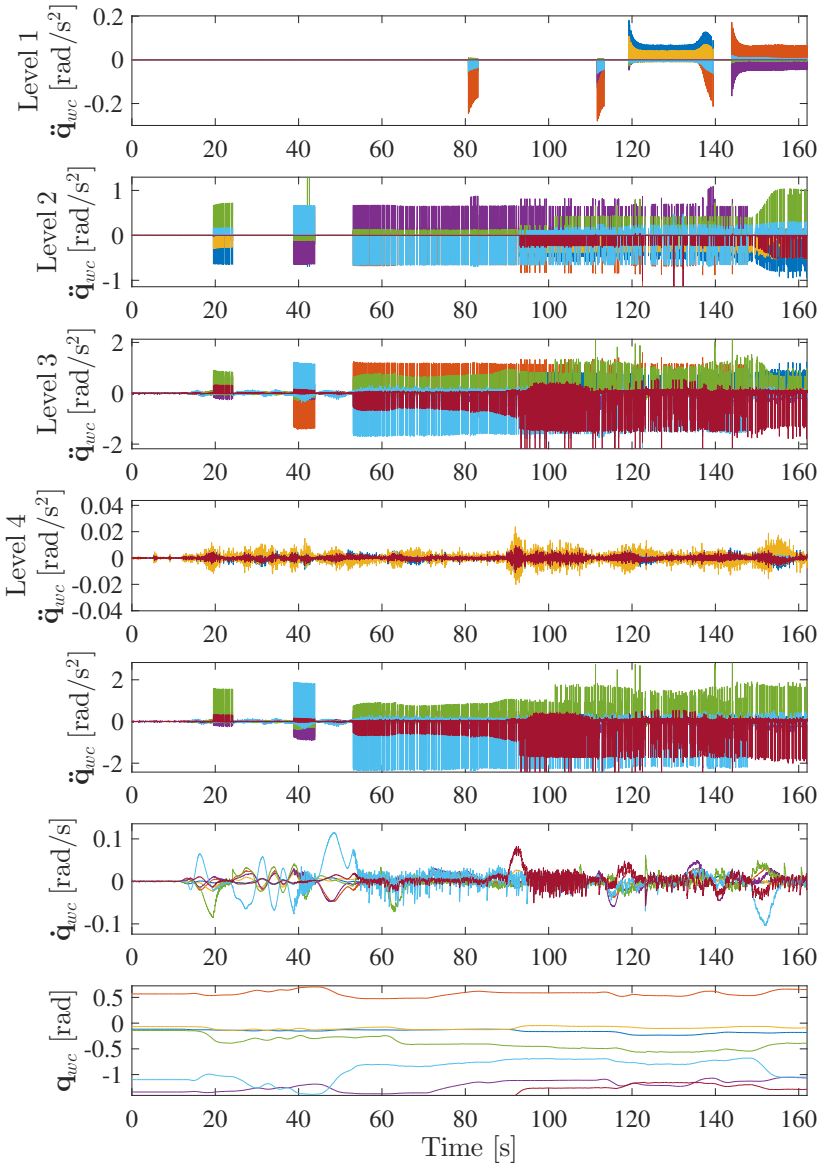


Figure 28. Experiment 7. Commanded joint actions for the WR: contribution of each priority level to the commanded joint accelerations in the first four plots, fifth plot represents commanded joint accelerations, sixth plot represents commanded joint velocities and seventh plot represents commanded joint positions.

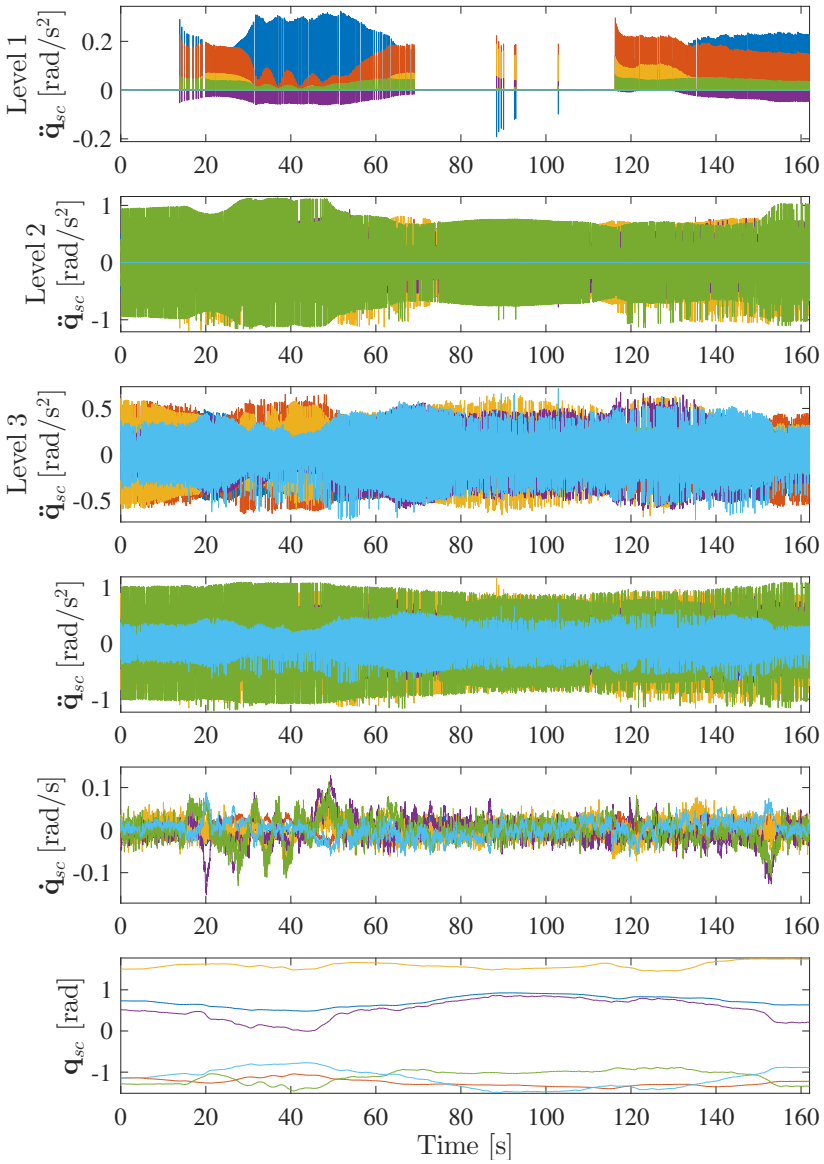


Figure 29. Experiment 7. Commanded joint actions for the STR: contribution of each priority level to the commanded joint accelerations in the first three plots, fourth plot represents commanded joint accelerations, fifth plot represents commanded joint velocities and seventh plot represents commanded joint positions.