Local Autonomy-Based Haptic-Robot Interaction With Dual-Proxy Model

Mikael Jorda, Margot Vulliez, Oussama Khatib

Local Autonomy-Based Haptic-Robot Interaction with Dual-Proxy Model

Mikael Jorda[1], Margot Vulliez[2], and Oussama Khatib[1],

Abstract

We present a new paradigm for performing remote haptic-robot interactive operations. The new paradigm is anchored on an architecture that combines local autonomy with a high-level exchange strategy of reference input. This represents a departure from the conventional reliance on direct exchange of low-level control signals in a global feedback control system. The new approach establishes two local autonomous controllers acting on the robot and the haptic device, interfaced at a higher level via a dual-proxy model. The dual proxy is a passive bridge between the local autonomous controllers. It generates appropriate motion and force reference inputs that are consistent with the task physical interactions and the levels of autonomy. Its model is adjusted online with respect to exchanged position, contact, and environment geometry information. A key component in this methodology is the perception algorithm on the robot side, the Force-Space Particle Filter, designed to reliably estimate in real time the environment contact geometry. The series of simulations and physical experimental validations of the approach demonstrate the transparency and high fidelity in haptic-robot interaction and its inherent robustness to communication delays.

Index Terms

Teleoperation, Haptic-robot control, dual proxy, local autonomy, contact geometry estimation

[1] Stanford Robotics Lab, Computer Science Department, Stanford University, 353 Jane Stanford Way, Stanford, CA, 94305, USA
{*mjorda, khatib*}*@stanford.edu*
[2] Pprime Institute, CNRS, University of Poitiers, 11 Bd Marie et Pierre Curie, 86962 Futuroscope Chasseneuil, France
margot.vulliez@univ-poitiers.fr

Local Autonomy-Based Haptic-Robot Interaction with Dual-Proxy Model

I. Introduction

ROBOTICS, well established in industrial production, is rapidly expanding into other areas that will have great impact on our society. The next stages will see new robotic systems that are able to work with and among humans in diverse environments, well outside of their usual, structured, factory domains. As robots become more and more a part of our lives, they will require greater and greater autonomy both at task execution levels (functional autonomy) and the situational-understanding and task-planning levels (cognitive autonomy). While much recent research increases robotic manipulation capabilities, the lack of general frameworks for high-level task understanding largely prevents modern systems from achieving full autonomy. One way of building highly capable robotic systems with today's technologies is to combine the cognitive autonomy and task expertise of humans with the functional autonomy of the robot executing the task. This improves both the human capabilities (becoming stronger and faster with the robot support) and the robot's capabilities (unable to complete the task without human guidance). A promising method to enable such human-robot collaborative systems is haptic teleoperation. This technology enables the human operator remote control of a robot while receiving feedback on its interactions. Haptic teleoperation is particularly beneficial in confined, unsafe, or sensitive environments such as hazardous sites, underwater, in space or surgery rooms. It naturally combines human high-level intelligence and robot physical capabilities while maintaining the safety and comfort required for the human's involvement.

Conventional teleoperation methods do not sufficiently exploit the robot's functional autonomy. These methods, based on a direct exchange of position, velocity and/or force commands between the human operator and the remote robot, aim at transferring the human intent to the robot and returning relevant task feedback to attain haptic transparency. They are often subject to additional transformations to maintain system stability. Such a direct low-level design introduces many challenges due to inaccurate dynamic modeling, sensor noise, task uncertainties, human-in-the-loop disturbances, communication time delays and non collocation as detailed in Section II. In particular, precise robot contact control is very challenging without local autonomy on the robot side. These challenges have been at the heart of haptic research as they impose a trade off between haptic transparency and stability.

Our proposal, in contrast, is the design of a new paradigm for haptic-robot control. Our approach leverages the functional autonomy on the robot and haptic sides by adding reasoning between the two via a dual-proxy model. The new concept of local autonomy and proxy-based haptic-robot interaction is depicted in Figure 1. Instead of exchanging direct commands between the robot and haptic device in a global feedback loop, we implement three different components. First, a local, autonomous controller on the robot side ensures a safe and accurate behavior toward the task objective. Second, a local controller on the haptic device side delivers effective task-specific haptic feedback to the operator. Third, a smart dual-proxy model exchanges high-level information through the communication channel and generates safe and consistent reference inputs for both local controllers. Conceptually, this approach is similar to having the haptic device and robot inform a task planner (the dual-proxy model) that decides the next steps to take on both sides, rather than connecting them directly at the task level. This enables more functional autonomy on both sides since the input can come from either the dual-proxy model or a local planner for different parts of the task and at under different circumstances. In addition, this approach does not introduce non-collocation in the task control which provides the advantageous side effect of removing stability issues that arise when using conventional teleoperation systems with time-delayed communication. The system's stability now depends on the stability of the two local controllers, and the interactions between the dual-proxy model and the controllers that occur at a slower rate. Finally, this approach also seeks to address the wide differences in remote tasks and in the physical interactions they involve within a simple and generic framework. As such, this paper brings the following contributions:

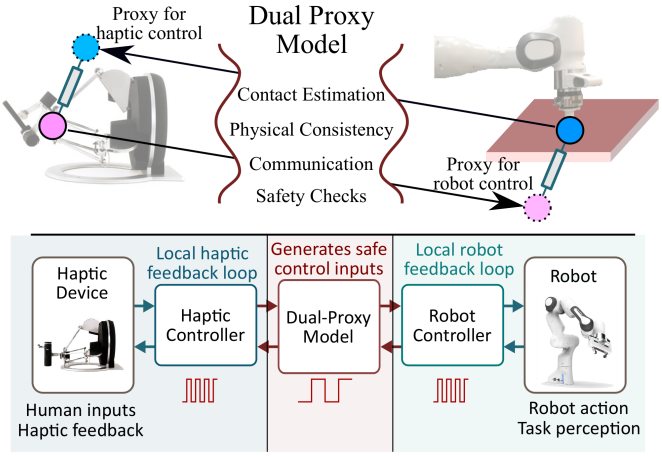

Fig. 1. Dual-proxy model for local autonomy-based haptic interaction. The dual-proxy model computes proxys for the robot and haptic device, generates safe and consistent control inputs, and parametrizes local controllers which regulate the desired behaviors on both sides. There is no fast global feedback loop required.

(i) The development of a novel, modular and adaptable

approach to haptic-robot control that leverages the robot functional autonomy and implements two local, autonomous controllers that interact at a higher level, rather than relying on a low-level global feedback loop.

(ii) The design of a dual-proxy model that acts as a bridge between the local control loops by handling the communication and generating safe and physically consistent reference inputs on both sides.

(iii) The design of a contact perception method, the Force-Space Particle Filter, that builds a local model of the environment to increase the robot functional autonomy and enable a fully autonomous parameterization of the controllers by the dual-proxy model.

(iv) The experimental validation of the new haptic teleoperation system on diverse contact interaction scenarios with diverse communication delays.

This paper is organized as follows: Section II presents an overview of related works and challenges in haptic control. Section III introduces the local autonomy-based approach and details its main components. Section IV discusses the Force-Space Particle Filter developed for adjusting the system behavior to the contact situation. Section V describes our implementation of the framework and details the selected local controllers, presenting simulation results showing the performances of the contact space estimator and local controllers. Section VI presents experimental results for different contact tasks under different time delays. Section VII presents our conclusions and proposes future considerations for this work.

II. Related works

Several decades of progress in haptics has advanced the originating mechanically-coupled teleoperation manipulators into the space of haptic-robot systems where a bilateral control framework ensures force and position pairing between the haptic device (user side) and the robot (the effector).

Most common bilateral teleoperation schemes are based on the classic force-position flow. It consists in sending the position of the haptic device as the input to an impedance controller on the robot [1], while sensing the interaction forces at the robot tool via a force sensor, then directly transferring this back to the haptic device for the human operator to experience. Unfortunately, the apparent inertia and friction of the haptic device quickly interferes with the perceived impedance, and this impacts feedback realism when using impedance-type controller [2]. High-inertia master devices either require closing the haptic control loop with force sensory feedback at the device level [3] or reversing the controlled flow by sensing the force applied by the operator on the haptic device and sending it as the input to an admittance controller on the robot [4]. With this admittance-type architecture, the velocity or displacement of the robot is sent back to control the haptic device and, therefore, create an implicit haptic feedback. Once again, issues arise as both the closed-loop admittance-type and impedance-type controllers are sensitive to environmental factors – such as its stiffness – and can exhibit position tracking errors or oscillatory behaviors [5].

Hybrid solutions have been designed to provide a better transparency balance under environmental change, such as continuously switching and interpolating between the impedance and admittance causality – at a fixed switching period [6] or with respect to the human input frequency [7]. However, only a four-channel control framework where both position and force of the device and robot are exchanged could theoretically achieve perfect transparency [8]. The four-channel bilateral controller has been widely studied and extended to include rate-controlled teleoperation systems [9] and both impedance and admittance-type robots into a generic framework [10]. Experimental evaluations of impedance, position, and force tracking [11] during free motions and hard contacts show that the four-channel architecture is clearly superior to the force-position controller, in any respect, by making a good use of available sensory information.

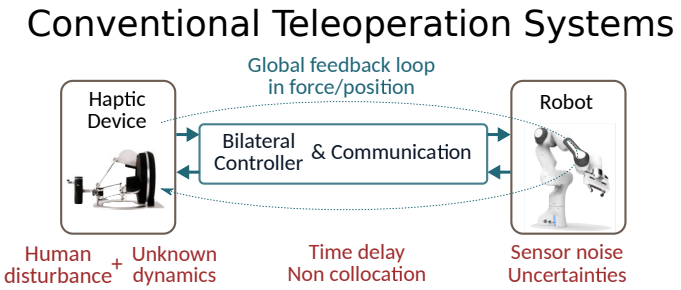

Fig. 2. Conventional haptic bilateral control implements a global feedback loop, which suffers from uncertainties and limits of the teleoperator, non collocation and time delays. Such drawbacks result in a trade-off between haptic transparency and stability in real-world applications.

While four-channel bilateral architectures allow perfect transparency under ideal conditions, they face several issues in real world, as depicted in Figure 2. Unpredictable dynamics, such as variable friction terms and effects of the human grasp dynamics [12], can cause errors and complicates the application of model-based methods – e.g. integrating a dynamic feed-forward term on the haptic device [13] or compensating for the robot inertial, Coriolis, and friction forces within a model-based natural admittance approach [14]. Sensor noise, time-varying delays, and communication data losses – among other effects – add up to model uncertainties and induce critical stability issues [15], [10]. They also reduce the dynamic range of achievable impedance [16] and impact transparency in contact, particularly during interaction with stiff objects [17].

Numerous methods have been proposed in the literature aimed at improving performance of haptic-robot systems. Transparency can be increased through adding local closed loops such as a force control loop [18] or two simultaneous force and position weighted loops [19] at the robot level, by modifying the force-position mapping within the bilateral architecture using non-linear and linear time-invariant filters [20], or by scaling the exchanged position/velocity and force variables [21]. Adaptive control laws are also commonly used to increase transparency of the teleoperation system, since they can estimate uncertainties in the environment, such as the environment stiffness [22], [23], compensate for unknown behavior of the human operator [24], adjust the models of the haptic device and robot [25], [26], and address hardware limitations such as the actuator saturation [27].

On the other hand, increasing stability of the bilateral controller in the presence of time delays and uncertainties has been the focus of many efforts (see [28] and [29] for surveys). Most are based on the passivity concept which builds upon the fact that a passive zero-state detectable system is stable in the Lyapunov-Krasovskii sense. Instability is mainly due to power variables (force, velocity) being transmitted through the communication channel, to be input at both sides of the global feedback loop. These power variables inject extra energy into the overall system and, therefore, jeopardize system stability. The first keystone work of [30] proposed a scattering method to maintain passivity of the communication channel. [31] proved that system energy can be preserved through the communication channel by transforming the exchanged data into specific wave variables. Such a wave-variable transformation can stabilize the four-channel bilateral controller [32] but, it can quickly skew position tracking of teleoperators, suffers from wave reflection which can result in unpredictable disturbances, and cannot guarantee system passivity in the presence of time-varying delays, jitter, or data loss [33]. This remains a foundation of numerous wave-based controllers, improving robustness to time-varying delays or packet losses [34] and force/velocity tracking [35].

Scattering and wave-based methods can be over conservative since they guarantee system passivity in the frequency domain for the worst-case scenario. To adjust the energy dissipation to the exactly-needed amount at the current time, the Time Domain Passivity Approach (TDPA) [36] extends the passivity-based approach to the time domain. This method does not require an exact estimation of the system model as long as one can observe the energy flows of its equivalent network. TDPA can, therefore, ensure the overall stability of the bilateral teleoperator, in force-position [37], as well as in position-position [38] or within the four-channel controller [39]. TDPA has also been extended to ensure the system passivity under time-varying communication delays [40], but at the expense of highly conservative passivity conditions, and resulting tracking errors. TDPA conservatism has been addressed by methods which consider the energy reflection from a storage element [41]. Overall, wave-variable controllers are more transparent in free motion or for higher delays while TDPA provides better performances in contact [42].

Even though passivity-based strategies can guarantee stability under time-varying delays and uncertainties, they narrow the range of achievable stiffness to maintain the passive behavior and, therefore, decrease the system fidelity. The passivity criterion is impacted by filtering and sampling delay variations, sensor resolution, and the system dynamics [43]. Methods are therefore needed to compensate for the degraded transparency while maintaining a sufficient level of force feedback fidelity [44], [45] to ensure haptic-robot position tracking [46], [47], or to reduce the inherent conflict between the transparency and stability objectives of the control framework [48].

This short literature review shows that, currently, bilateral haptic-robot coupling is provided through global feedback loops, which ensure force-position tracking between the two robots. While providing good performances at the expense of increasing complexity, the most advanced architectures are still balancing the trade-off between stability and transparency of the haptic teleoperator. This balance is ensured within a given bandwidth, under strictly limited conditions (such as a bounded stiffness range) and with finely tuned control gains. Aside from this complex tuning of the controller, such global-loop control architectures do not adapt to significant changes in the teleoperator design and can be strongly affected by hardware parameters, such as stiffness and inertia of the haptic device and the robot [49] and force sensing and filtering quality [50], [51]. Real-world uncertainties, variability, and constraints of teleoperators complicate the spread of haptic applications outside of a limited circle of expertise. This observation motivates our work – we aim to develop a generic and modular approach to haptic-robot control that can easily adapt to different tasks, haptic devices, and robots.

In his precursor works, Hannaford [52] shows that duplicating the impedance control modality to the device side, to form a bilateral impedance controller also known as position-position controller, increases stability and robustness to time delays, as two local impedance loops enforce the device and robot commands. Another way to break down the global feedback loop has been later proposed with the Model-Mediated Telemanipulation (MMT) framework [53], [54]. The MMT method abstracts sensory data, from the robot and other potential sensors, to build and update an environment model. Assuming that the environment is only slowly changing, its model is transmitted at the master side to locally replicate the interaction. The force feedback is, then, computed from the interaction between a first-order dynamic proxy, connected to the haptic device, and the local environment model. Then, the slave robot either tracks the master motion command, in the environment-model free space, or the master force command, when a contact is detected between the proxy and the model. Such a local proxy-based interaction allows stable behavior of the teleoperation system under large communication delays.

Our framework further develops this idea of local controllers and proxy-based interaction. We propose a novel paradigm for haptic-robot control where both the haptic device and remote robot are autonomous agents, enabling as much autonomy as possible at the task level, and interfacing them at a higher level, through a dual proxy, to ensure synchronized force-position behaviors. The main improvements from the MMT approach are : 1) Our framework leverages local autonomy in the haptic device and the robot controllers. The robot, and the haptic device, are fully autonomous and can locally adjust their behavior with respect to their interaction with the task environment, or with the human hand. For example, the robot can create its own local model of the environment to parameterize its local controller and improve its contact behavior. 2) In MMT, a first-order dynamic proxy follows the master kinematics, within the constrained environment model, to embody a representation of the slave behavior. Our proxy concept is different: a virtual object dynamically complies with the state of each robot. The master proxy is directly computed from the slave-robot configuration and is, therefore, physically consistent with the robot behavior (most likely to have a second-order dynamics). 3) The proxy-based interaction is extended on both sides, to remove any direct coupling between

the haptic device and the robot. Two proxys are updated from the haptic device and the robot configurations. Inputs for the haptic and robot local controllers are then respectively computed from the interaction between the haptic device and the haptic proxy, and, between the robot and the robot proxy. 4) No global environment model needs to be estimated, since the desired haptic feedback is computed from a virtual spring which links the haptic device to the haptic proxy. It makes our approach more robust to estimation errors as it only takes advantages of local environment estimations. 5) The bilateral dual-proxy model enables the integration of safety conditions and physical constraints within the generation of the controller commands. 6) Our generic framework is validated on a real-world multi-DOF teleoperation system for several case applications.

III. A NEW FRAMEWORK FOR HAPTIC-ROBOT INTERACTION

In this discussion we reconsider haptic teleoperation through a local autonomy-based control strategy. It implements local controllers at the robot and haptic device sides, to respectively monitor the task actions and provide a proper perception to the user, while only exchanging high-level reference inputs between the two robots. This approach is based on the introduction of the dual-proxy model: a smart bridge between the two local loops which encodes haptic-robot interactions.

A. Local autonomy-based approach

The concept of the new haptic control framework is depicted in Figure 1. It comprises three main components: the device local control loop, the robot local control loop, and the dual-proxy model. Dividing the teleoperation scheme in these two local loops – linked by the dual-proxy model – endows the haptic device and the robot with a certain autonomy. It prevents the exchange of direct force/velocity data through the communication channel, as the local loops handle the motion and force control of the robot and device. This local autonomy-based approach separates functions and objectives of the two robots. The haptic device must reflect the perception of the task to the human operator through proper force feedback. To meet this perception objective, the haptic local loop must generate a high-fidelity remote sensing of the environment, which means that free space must feel free while contact with objects must feel real. On the other hand, the robot must perform the remote actions to do the task while tracking the human inputs. The robot controller monitors the end-effector dynamic behavior and the task physical interactions, involving both precise motion and contact force. Each local controller is designed independently of the other and tuned with respect to its targeted performance.

Using autonomous local controllers ensures the modularity and generality of the framework since they can be easily changed and tuned for adapting to different tasks or hardware without having to redesign the whole system. In addition, local controllers allow many possibilities for shared autonomy where different parts of the task can be supervised via different agents – autonomous or human – using dual-proxy models, and provide easy interfacing with local sensing modalities to improve the functional autonomy of the task. An example of this latter point will be the use of the Force Space Particle Filter in detecting the contact geometry of robot interactions during operation and helping to parametrize the system.

B. Dual-proxy model

The notion of proxy was first developed for increased robustness in haptic rendering and haptic shading of virtual environments [55], [56], [57]. Similarly to the god point concept [58], the idea is to create a point, the proxy, in the virtual environment that kinematically follows the user input but is subject to the constraints of the virtual environment. The deviation between the proxy and the haptic device positions is then used to generate a haptic spring force. It pulls the haptic device toward the proxy, and therefore, replicates the environmental constraints. It has been shown to be an extremely effective model to render a virtual environment haptically. The idea of the dual-proxy model is to define two proxys, one for the robot and one for the haptic device that will be used to generate commands for the local controllers. The proxys are, therefore, virtual objects which represent the behavior of the two robots in the workspace of their counterparts. With this proxy-based approach, only position information has to be transmitted over the network, which cannot jeopardize stability of the teleoperation framework. The dual-proxy model uses knowledge of the robot environment that can be known *a priori* or computed with the online perception algorithm described in section IV, and the knowledge of the robot and haptic device states in order to generate inputs to the local controllers. The dual-proxy model handles the communication between the robot and haptic device and prevents non-collocation issues. It is described in Figure 3.

1) Computing haptic and robot proxys:

The first role of the dual-proxy model is to handle the communication between the two robots and to generate a proxy for each. The proxy captures the behavior of the robot and of the haptic device in the frame of the other. To this end, the dual-proxy model copies each robot position data and transforms it through a proper workspace mapping. The most commonly used mapping relates the two workspaces via a translation, rotation, and constant scaling factor.

The haptic device current position x_h and orientation R_h, in pink, are measured in the device frame. They encode the human desired pose for the robot. Positions and orientations are sent to the dual-proxy model, where they are mapped onto the robot frame – given the rotation matrix from the device frame to the robot frame R_{h-r}, the translation between the robot frame and the desired center of the robot workspace in position p_{r-cw} and orientation R_{r-cw}, and the scaling factor s – in order to obtain the robot proxy position $x_{p,r}$ and orientation $R_{p,r}$:

$$\begin{aligned} x_{p,r} &= p_{r-cw} + sR_{h-r}^T\ x_h \\ R_{p,r} &= R_{h-r}^T\ R_h\ R_{h-r}\ R_{r-cw} \end{aligned} \tag{1}$$

The reverse transformation computes the haptic proxy $x_{p,h}$, $R_{p,h}$ from the robot position x_r and orientation R_r:

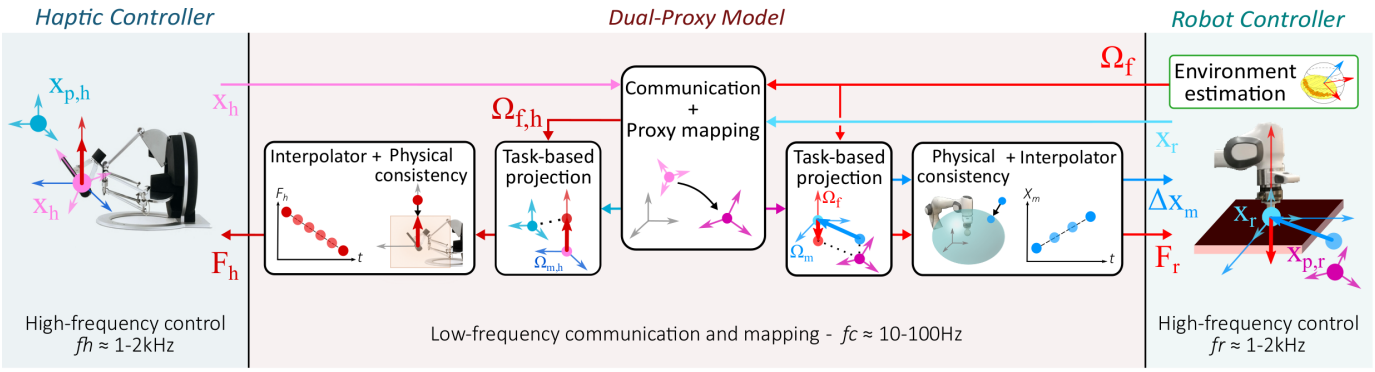

Fig. 3. A smart dual-proxy model replicates haptic-robot interactions within the local autonomy-based framework. It receives robot and task-related information through the network and computes respective proxys for the two robots. Safe and consistent control inputs are generated on both sides from the robot-proxy and haptic-proxy interaction.

$$\begin{aligned} x_{p,h} &= R_{h-r}\ (x_r - p_{r-cw})\ /\ s \\ R_{p,h} &= R_{h-r}\ R_r\ R_{r-cw}^T\ R_{h-r}^T \end{aligned} \tag{2}$$

It is worth noting that the same workspace mapping transformation is applied to any data passed through the network from one robot to the other, in order to maintain the physical consistency between all remote information. In particular, the model of the robot environment geometry needs to be mapped to the haptic device to maintain the correct force feedback.

In general purpose teleoperation applications, the communication network is often slower than the desired control frequency and rarely ensures the hard-realtime constraints of haptic-robot interaction. In addition, communication delays and packet losses generate many issues in conventional teleoperation systems that use one global feedback loop. This is particularly true when there is a significant distance between the haptic device and the robot, for example when teleoperating a robot in space.

The dual-proxy model removes these communication issues. Indeed, since the dual-proxy model and the two local controllers are independently implemented, they can run at different frequencies. Typically, the local control loops will run at $1kHz$ or more in order to get the best possible performance. However, the communication channel can run at a lower frequency, which will mostly depend on the chosen communication technology, and will typically be between $10-100Hz$. In addition, this frequency difference enables the computation of smooth proxys by averaging position data between two consecutive communication messages. Finally, communication delay and latency spikes can be monitored by the dual proxy in many ways without compromising the performance of the local control loops since it runs independently.

2) Task-related proxy projection:

The second role of the dual proxy is to use the proxys, robot configurations, and robot environment information to generate motion and force control inputs. A broad spectrum of physical interactions are involved when performing tasks in haptic teleoperation. Remote minimally-invasive surgical operations require a high accuracy in motion while assembly tasks or polishing works need precise perception and regulation of the contact forces with the environment. This range of interaction implies different objectives for the local control loops. To achieve high fidelity in both motion and contact tasks, the dual-proxy model smartly generates the control inputs such that they properly reflect the real-world physical interaction. The strategy consists in either computing a motion command (if the robot is moving in some direction) or computing a force command (when the robot is in contact with the environment).

The control inputs are computed based on a projection of the robot proxy and haptic proxy onto motion and force spaces, defined by the task physical interaction, whether the robot is in contact with the environment or not. The environment geometry is described locally by the directions of constraints and directions of free motion. These are encoded by selection matrices for the robot force space Ω_f and the robot motion space Ω_m. These matrices are 3D projection matrices, and their rank corresponds to the number of degrees of freedom in the force and motion spaces respectively. Additionally, the force space and motion space must be orthogonal so we have $\Omega_f + \Omega_m = I_{3\times3}$. A similar selection process projects the end-effector rotational motion and applied torques onto rotation and moment spaces, respectively, through Ω_{rot} and Ω_τ, with $\Omega_{rot} + \Omega_\tau = I_{3\times3}$. Note that because of these relationships, only the force and moment selection matrices require transfer from the robot to the haptic device. However, they need to be mapped to the haptic device workspace. The force and moment selection matrices for the haptic device are therefore:

$$\begin{aligned} \Omega_{f,h} &= R_{h-r}\ \Omega_f\ R_{h-r}^T \\ \Omega_{\tau,h} &= R_{h-r}\ \Omega_\tau\ R_{h-r}^T \end{aligned} \tag{3}$$

Figure 4 illustrates the generation of force and motion commands for the robot and haptic device. On each side, the proxy is projected onto the force/moment spaces on the one hand, and onto the motion (linear and angular) spaces on the other. In motion space, the projected proxys represent the goal points. In force/moment spaces, the position/orientation difference is transformed to a reference force using a virtual spring. We detail the generation of the linear inputs here, as the generation of rotational inputs is very similar. On the robot side, the set points for the motion and force controllers are:

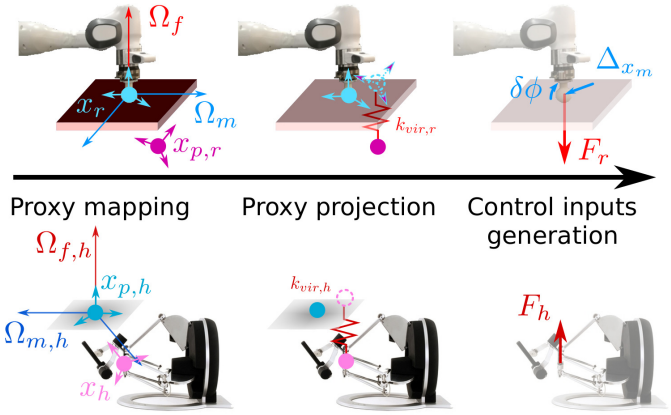

Fig. 4. Force/motion-space projection of the robot-proxy and haptic-proxy interactions to generate consistent local control inputs. Left: Ω_f and Ω_m represent the force space and motion space projections respectively. They allow construction of a local model of the environment geometry on the haptic side (gray plane). x_r (cyan) and x_h (pink) are the positions of the robot and haptic device, $x_{p,r}$ (magenta) is the robot-proxy position, and $x_{p,h}$ (blue) is the haptic-proxy position. Middle: The proxys are projected into the force space and motion space, a virtual spring is used in the force space to generate a force from the difference in positions. Right: Control inputs are generated using interpolation and safety checks. On the robot side, a position error Δ_{x_m} is generated in the motion space and an orientation error $\delta\phi$ to perform orientation control. In the force space, a reference force F_r is generated. On the haptic side, only a reference force F_h is generated in this example.

$$\begin{aligned} \Delta x_m &= \Omega_m \ (x_r - x_{p,r}) \\ F_r &= -k_{vir,r} \ \Omega_f \ (x_r - x_{p,r}) \end{aligned} \tag{4}$$

where $k_{vir,r}$ is the stiffness of the virtual spring on the robot side. Monitoring these task-based control modalities at the robot level gives an intuitive and explicit control of the task interaction, whether it involves precise motions or contact forces. The robot local controller regulates the end-effector dynamic behavior to reach the desired control objectives and properly perform the remote task. On the haptic side, the controller must reflect the task interaction to the human operator. The haptic control objective is, therefore, to produce a high-fidelity remote sensing of the contact forces without any constraint in free space. The haptic control inputs reflect the contact force onto the force space while ensuring free motion of the operator onto the motion space. The projection-based approach ensures a good free space transparency as the desired control force is set to zero in motion space. In the force space, the feedback force is generated as:

$$F_h = -k_{vir,h} \ \Omega_{f,h} \ (x_h - x_{p,h}) \tag{5}$$

In addition to avoiding the exchange of force signals, computing the desired haptic and robot force through a virtual-spring model is a convenient tool for adjusting the haptic feedback with respect to the device maximum stiffness and to adapt the robot behavior in contact. For example, we could reduce the spring stiffness to bring compliance in the force control input and to increase safety of the interaction. If the same virtual-spring model is implemented on both sides, the same (opposite) desired forces, up to the scaling factor, are input to the robot and haptic local loops.

3) Adjusting the local control inputs:

The last component of the dual-proxy model aims at maintaining safety and adequacy of the control inputs for the local loops. The desired force and motion, computed from the proxy projection, are adjusted according to high-level exchanged information, such as the control and communication frequencies, the robot and haptic device limits, and task-related constraints. Therefore, each proxy-based input not only reflects the interaction, but adapts to the controlled robot and can be modified to increase task performance.

The first step is to perform safety checks on the control inputs with respect to hardware limitations. This ensures physical consistency of the desired force and motion for the two robots. The desired haptic feedback is maintained under the haptic device maximum force, and damping can be added in the force space to prevent undesired oscillations. On the robot side, if the motion input goes outside the range of motion, it is projected back onto the robot workspace. The robot control force can also be saturated to a maximum value to preserve safety of the robot/environment contact. Finally, the monitoring of communication delays and possible latency spikes or communication loss enables the dual proxy model to reject potentially dangerous and outdated commands.

An additional safety stage is implemented on the robot side to prevent the motion command going outside the robot kinematic and dynamic operational range. The command is interpolated according to the robot maximum velocity and acceleration in the operational space [59]. The desired robot motion is, therefore, safely reachable, even if the human motion input is too fast or involves high acceleration change. In addition, the frequency difference between the communication loop and the control loops could result in undesired step commands in the force space, in particular when contact transition occur. In order to mitigate this and maintain the system's stability, the desired forces on both sides are interpolated by the controller to prevent force commands that would be uncomfortable for the human on the haptic device and potentially dangerous on the robot side.

These safety checks enable the generation of safe commands for the robot and haptic device, regardless of what is happening on the other side of the communication network or the time it took for the data to be transmitted. In particular, it ensures the safety of the system in case of a communication loss.

IV. Online Perception of Robot Environment for Task-Level Autonomy

With the new framework, proper knowledge of the robot environment geometry – the force space and motion space – is crucial to the haptic force fidelity. In certain tasks (manufacturing, for example), it is possible to know in advance the location and direction of the constraints. However, in general use cases, we want the robot to autonomously detect the directions of constraints, in real time, to infer the force and motion spaces and properly parameterize the proxy projection. In this section, we design the Force-Space Particle Filter (FSPF). The FSPF states will represent the likely contact normals at the robot

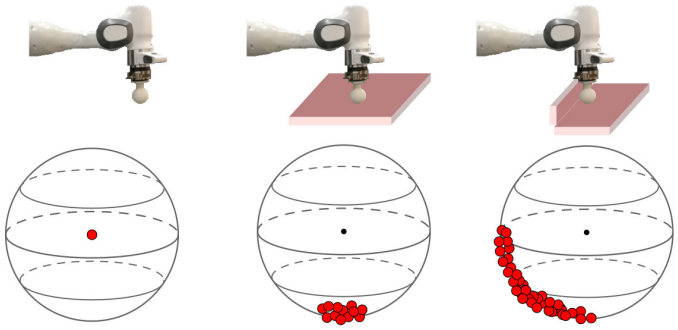

Fig. 5. Illustration of the Force-Space Particle Filter. The figures on the top show the robot and potential contact scenarios, the figures at the bottom show a corresponding distribution of the particles. Left: no contact. Middle: one direction of contact. Right: two directions of contact.

end effector, in order to find the force and motion spaces. The FSPF is implemented for linear force and motion directions. Its extension to angular force-motion spaces is immediate.

A. The Force-Space Particle Filter

The idea behind the FSPF is the following: given a robot in contact that is moving, we should be able to find the contact normal by combining contact force and velocity data. In particular, the contact normal should be close to the direction of the sensed force, and orthogonal to the robot velocity. Methods such as Kalman Filters were studied to provide environment estimations such as in [60]. However, the problem we are trying to solve here has huge discontinuities when transitioning from free space to contact, and when transitioning to different contact dimensions (contact with a plane vs contact in a corner). The advantage of using a particle filter over Kalman filters or other methods is its ability to represent and estimate a mixed continuous (contact directions) and discrete (number of contact constraints) state space.

The FSPF state space is the 3D unit sphere plus its center. The sphere represents all the possible contact normals at the robot end effector, while its center encodes the belief that there is no contact on the robot end effector. Figure 5 illustrates the behavior of the FSPF. When the robot is in free space, we expect most particles to be at the center. When the robot is in contact in a single direction, we expect most particles to be grouped around the normal to the contact. When there are several directions of contact, the particles will be scattered along all the probable directions and will form a portion of circle (two directions of contact) or a portion of sphere (three directions of contact). Every iteration of the particle filter consists of two steps – the update step and the resampling step. Let $\mathcal{P}_k$ be the list of particles at the k^{th} iteration of the filter and n_p the number of particles. We will use the symbol ρ to refer to individual particles. Let us also note F_s, v_s, F_m, F_f, respectively, the end effector sensed force, the sensed velocity, the control force in motion space, and the control force in force space. These four quantities are required in the update and resampling steps.

The update step consists of creating a list of augmented particles $\overline{\mathcal{P}}_k$ knowing the current list of particles and the control and sensor inputs. $\overline{\mathcal{P}}_k$ contains $\overline{n_p} > n_p$ particles. The resampling step first assigns a weight $w(\rho, F_s, v_s) \in [0, 1]$ to each particle $\rho \in \overline{\mathcal{P}}_k$ that measures the likeliness of that particle given the sensor measurements, and, then, resamples n_p particles from $\overline{\mathcal{P}}_k$ by prioritizing the particles with high weights in order to obtain the next list of particles $\mathcal{P}_{k+1}$

1) Step 1: Update:

At any point during robot operation, 3 things can happen:

- The contact directions can change
- A contact direction can appear
- A contact direction can disappear

The update step needs to accommodate for these three possibilities. We will create the augmented particle list $\overline{\mathcal{P}}_k$, starting with an empty list $\overline{\mathcal{P}}_k = \emptyset$ and using the procedure described in Algorithm 1. We use a parameter $0 < \epsilon \ll 1$ to numerically determine if the vectors are at the origin. The algorithm is explained next:

Algorithm 1: FSPF Update Step

Result: A list of augmented particles $\overline{\mathcal{P}}_k$

```
/* start with an empty list                          */
P̄_k = [];
/* move the existing particles                       */
for ρ ∈ P_k do
    if ‖ρ‖ > ε then
        d ~ N(α, Σ);
    else
        d = 0;
    end
    P̄_k.append(ρ + d)
end
/* add particles in the direction of motion
   control                                           */
if ‖F_m‖ > ε then
    ρ_tent = F_m/‖F_m‖;
    if ‖F_f‖ > ε then
        u_f = F_f/‖F_f‖;
    else
        u_f = 0;
    end
    n_add = w(ρ_tent, F_s, v_s) * n_p;
    for i ∈ [1, n_add] do
        s = (i − 0.5)/n_add;
        P̄_k.append(s * u_f + (1 − s) * ρ_tent);
    end
end
/* add a particle at the center                      */
P̄_k.append(0);
/* renormalize all particles                         */
for ρ̄ ∈ P̄_k do
    if ‖ρ̄‖ > ε then
        ρ̄ = ρ̄/‖ρ̄‖;
    else
        ρ̄ = 0;
    end
end
```

Algorithm 2: FSPF Resampling Step

Result: A list of resampled particles $\mathcal{P}_{k+1}$

/* compute the list of cumulative weights */
$w_{list} = []$;
$w_{cumul} = 0$;
for $\overline{\rho} \in \overline{\mathcal{P}_k}$ **do**
  $w_{cumul} = w_{cumul} + w(\overline{\rho}, F_s, v_s)$;
  $w_{list}.append(w_{cumul})$;
end
/* rescale the cumulative weight list */
for $w \in w_{list}$ **do**
  $w = w/w_{cumul}$;
end
/* perform low variance resampling */
$\mathcal{P}_{k+1} = []$;
$r \sim \mathcal{U}(0, \frac{1}{n_p})$;
$l = 0$;
for $i \in [1, n_p]$ **do**
  while $w_{list}[l] > r$ **do**
    $l = l + 1$;
  end
  $\mathcal{P}_{k+1}.append(\overline{\mathcal{P}_k}[l])$;
  $r = r + 1/n_p$;
end

1) First, we take each particle in $\mathcal{P}_k$ and move it randomly following a normal distribution $\mathcal{N}(\alpha, \Sigma)$ (if it is not at the center). α is the mean of the normal distribution and Σ its standard deviation. We add the obtained particle to $\overline{\mathcal{P}_k}$. This allows to take into account the possible rotation of the force space.
2) Second, we look at the direction of the motion control force F_m and create a tentative particle ρ_{tent}. We assume that if a contact appears, it will be in the direction towards which the robot is moving. In order to know if this tentative particle should be added, we look at its likeliness using the same weight function as the resampling step (that will be later described) $w(\rho_{tent}, F_s, v_s)$. We then create new particles between the force control F_f direction and the tentative particle. The higher the likeliness of ρ_{tent}, the more particles we add. This step allows the filter to take into account a new direction of contact.
3) Third, we add one particle at the center in order to take into account a possible contact loss.
4) Finally, we normalize all the particles that are not in the center in order to project them back onto the sphere.

2) Step 2: Resampling:

We define a force weight function w_f and a velocity weight function w_v for a given particle ρ and the value of the measured force F_s and velocity v_s. These functions represent a credence value for the particle based on the force sensor measurement and the velocity measurement. For the force weight function and particles on the sphere, we will consider that if the sensed force is higher than a threshold f_h in the direction of the particle, then it is highly likely, and if it is lower than a threshold f_l, it is not likely at all. For particles at the center of the sphere, we consider them likely if there is no sensed force, and not likely if the sensed force is high. We define:

$$w_f(\rho, F_s) = \begin{cases} \dfrac{< \rho, F_s > - f_l}{f_h - f_l} & \text{if } \rho \neq 0 \\ 1 - \dfrac{\|F_s\| - f_l}{f_h - f_l} & \text{if } \rho = 0 \end{cases} \tag{6}$$

where $< ., . >$ represents the euclidean scalar product. For the velocity weight function and particles on the sphere, we will consider that a particle is not very likely if there is a significant velocity in its direction or its opposite direction, and it is likely otherwise. For particles at the center, the velocity gives no information (as the robot could be moving in free space or in contact) so we will assign a velocity weight of 0.5. We define:

$$w_v(\rho, v_s) = \begin{cases} 1 - \dfrac{|< \rho, v_s >| - v_l}{v_h - v_l} & \text{if } \rho \neq 0 \\ 0.5 & \text{if } \rho = 0 \end{cases} \tag{7}$$

where v_h and v_l are high and low velocity thresholds. Figure 6 illustrates these weight functions. The thresholds f_l, f_h, v_l, v_h are used to provide some robustness to sensor noise and prevent the detection of false contacts. They need to be tuned depending on the system and sensors.

Using the previously defined functions, we can perform the resampling step. The weight of each particle $w(\rho, F_s, v_s)$ is defined as the product of the force and velocity weights for that particle:

$$w(\rho, F_s, v_s) = w_f(\rho, F_s) w_v(\rho, v_s) \tag{8}$$

We see that $w(\rho, F_s, v_s)$ is always between 0 and 1. Besides, the product means that as soon as one of our sensor modalities deems the particle unlikely, then it will be excluded. For the resampling itself, we use low variance resampling as defined in [61] in order to prevent the loss of particles that are isolated but have a high probability. The resampling step is implemented using Algorithm 2.

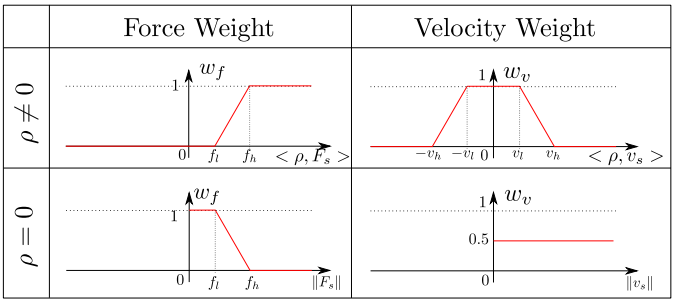

Fig. 6. Force and velocity weight functions

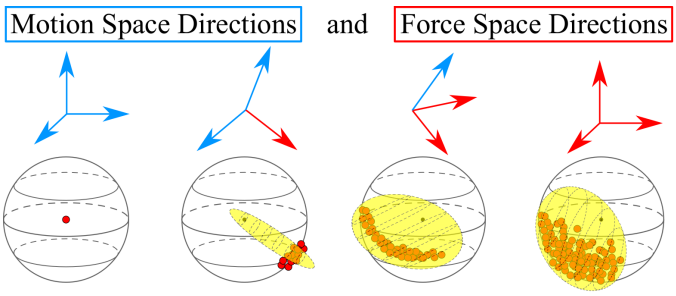

Fig. 7. Motion and force spaces for a given set of particles are found by performing an SVD of the particles. The high singular values correspond to force directions and the low singular values to motion directions.

B. Finding the Force and Motion Spaces

The output of the FSPF, described, is a list of likely particles at the current time. We need to extract a likely force space and motion space from this particle list. Intuitively, the distribution of the particles will give us enough information to estimate the force and motion directions. In particular:

- When most particles are at the center, we expect the robot to be in free space,
- When most particles are grouped somewhere on the sphere, we expect the force space to be of dimension one, and the location of the particles gives us its expected direction.
- When most particles are along an arc of circle, we expect the force space to be of dimension two, and its direction is given by the plane that contains the arc of circle.
- When the particles are scattered in a 3D pattern on the sphere, we expect the force space to be of dimension three.

In order to find the dimension and direction of the force space, we perform singular value decomposition (SVD) of the particle set. We define X_p, a matrix of dimension $3 \times 1.5 * n_p$ such that the first n_p columns of X_p are the coordinates of the n_p particles of the current state of the particle filter $\mathcal{P}_k$, and the last $0.5n_p$ columns are zero (which represent particles at the center). The left SVD of X_p will give a 3D ellipsoid that best explains the points, as depicted in yellow in Figure 7. The magnitude of the principal axes of the ellipsoid are the singular values and their orientation are the singular vectors. The directions associated with high singular values are force directions and the ones associated with low singular values are motion directions.

Let us note d_{fs} the dimension of the force space that can be between 0 and 3. Let us also note U the 3x3 matrix of the left singular vectors of X_p, and $U = [U_1\ U_2\ U_3]$ its columns. Let us finally note $S = [\sigma_0, \sigma_1, \sigma_2]$ the vector whose elements are the singular values of X_p ordered in decreasing order ($\sigma_0 \geq \sigma_1 \geq \sigma_2$). We apply Algorithm 3 in order to find the selection matrices Ω_f and Ω_m. The algorithm is described here:

- The norm of S gives an indication on the number of particles that are not at the center. Indeed, since all particles have norm 1 or 0, let m be the number of particles not at the center, we have $\|S\| = \sqrt{m}$. In general, we will consider that the robot is in free space unless at least half of the particles are not at the center. Therefore, we have the following relationship:

$$\text{if} \quad \|S\| < \sqrt{n_p/2}, \qquad d_{fs} = 0 \tag{9}$$

- When $d_{fs} > 0$, we define two thresholds $0 < \alpha_{add} < 1$ and $0 < \alpha_{remove} \leq \alpha_{add}$ that will be used to add or remove dimensions in the force space. When $\sigma_1 > \alpha_{add}\sigma_0$, and $\sigma_2 > \alpha_{add}\sigma_0$, we add their directions to the force space. When $\sigma_1 < \alpha_{remove}\sigma_0$, and $\sigma_2 < \alpha_{remove}\sigma_0$, we remove their directions from the force space. Choosing $\alpha_{add} > \alpha_{remove}$ generates some hysteresis in the value of d_{fs} which helps stabilizing the transitions.
- With the value of d_{fs}, we can compute the projection matrices Ω_f and Ω_m from the columns of U.

The three algorithms presented in this section show the full implementation of the FSPF. The particle filter is used to provide an added layer of local autonomy. The controllers on the robot and haptic side can self parameterize autonomously in real time using sensory observations. The FSPF and the dual-proxy model are the key innovations in our local autonomy-based approach. For the robot and haptic local controllers, many choices can be made.

C. Extensions of the FSPF

In this section, The FSPF was presented in its simplest form. It provides a baseline algorithm that works even for changing environments and where no additional sensing is available. If vision sensing is available on the system, this can be used to improve the performance of the estimation algorithm (as has been shown in [62] for model mediated teleoperation). It would be integrated easily with the particle filter by providing a prior to the particles states before the update step. More precisely, instead of starting with an empty list of particles at the start of the update state, the vision system would provide an expected list of particles. In a similar way, position history can be used to provide a prior based on the particle filter output last time a certain position was visited, which would increase the performance for static environments.

In addition, the extension of the FSPF to a full 6DoF position plus orientation estimation of the constraints is immediate. A second FSPF can be run, replacing linear quantities with angular quantities in the previous algorithms (forces to moments, linear velocity to angular velocity, linear control input to angular control input). This second FSPF will provide the directions of constraints and free space motions for the robot orientation. Combining the two FSPF gives the full 6DoF geometric constraints. For example, when the robot end effector is in surface contact with the environment, the linear FSPF will detect one constrained direction (in the normal direction to the surface contact) and the angular FSPF will detect one direction of free motion (the rotation around that same normal axis).

V. Implementation of the Local Controllers

The local autonomy-based dual-proxy method allows us to choose controllers for the robot and haptic device independently of everything else. The dual-proxy model and FSPF

Algorithm 3: FSPF Force and Motion space Matrices Computation

Result: d_{fs}, Ω_f and Ω_m
/* compute the SVD of the particles */
$X_p = Matrix.Zeros(3, 1.5n_p)$;
$i = 0$ **for** $\rho \in \mathcal{P}_k$ **do**
  $X_p[i,:] = \rho$;
  $i = i + 1$;
end
$U, S, V = SVD(X_p)$; // U is 3x3, S is 3x1 and the coefficients are in decreasing order
/* compute the dimension of the force space and the projection matrices */
if $\|S\| < \sqrt{n_p/2}$ **then**
  $d_{fs} = 0$;
  $\Omega_f = 0$;
  $\Omega_m = I_3$;
else
  /* compute the dimension using the lower and higher thresholds */
  $d_{lb} = 3 - (\sigma_1 < \alpha_{remove}\sigma_0) - (\sigma_2 < \alpha_{remove}\sigma_0)$;
  $d_{hb} = 1 + (\sigma_1 > \alpha_{add}\sigma_0) + (\sigma_2 > \alpha_{add}\sigma_0)$;
  if $d_{fs} < d_{hb}$ **then**
    $d_{fs} = d_{lb}$;
  end
  if $d_{fs} > d_{lb}$ **then**
    $d_{fs} = d_{hb}$;
  end
  /* compute the projection matrices */
  if $d_{fs} == 1$ **then**
    $\Omega_f = U_1 U_1^T$;
    $\Omega_m = I_3 - \Omega_f$;
  else
    if $d_{fs} == 2$ **then**
      $\Omega_m = U_3 U_3^T$;
      $\Omega_f = I_3 - \Omega_m$;
    else
      $\Omega_f = I_3$;
      $\Omega_m = 0$;
    end
  end
end

together ensure the safety and physical consistency of the local control inputs, as well as the correct parameterization of force and motion spaces. The robot local controller needs to be able to regulate both free space motions and physical interactions in a stable way. Two good choices are to implement an impedance controller [63] or a unified force/motion control [64]. The latter requires a re-parameterization of the controller when the force space changes, but it enables a completely decoupled dynamic response in force space and in motion space. In haptic applications, the choice of stiffness in force space will mostly depend on the maximum stiffness that the haptic device can render, and the compliance needed for the task. We want to be able to select it independently of the

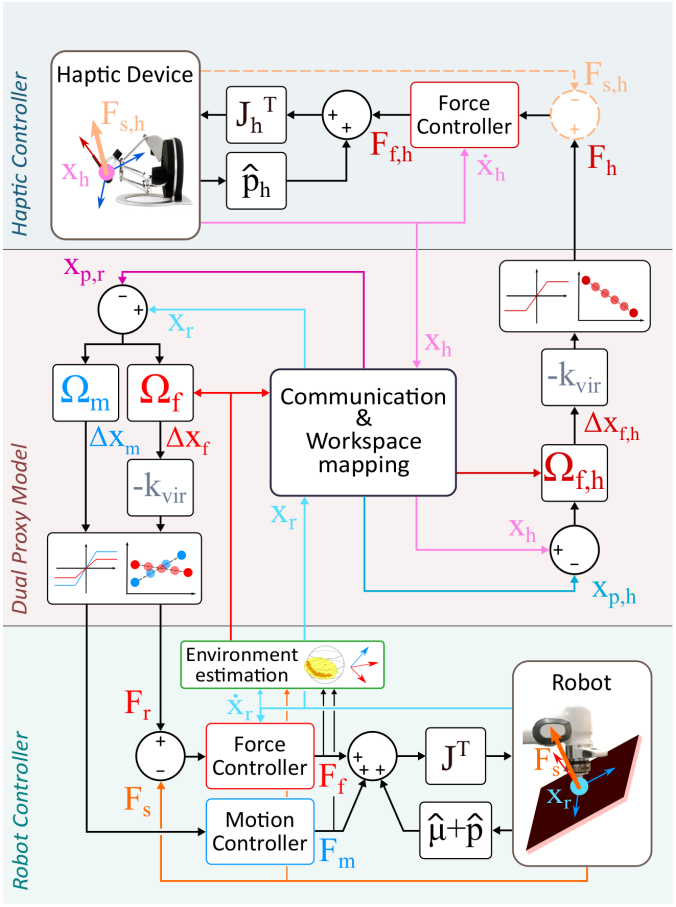

Fig. 8. Block diagram of the local autonomy-based dual-proxy framework with our selection of local controllers.

free-space motion tracking performance. For this reason, we advocate for the usage of operational space control on the robot side. On the haptic side, the controller will mostly depend on which haptic device is used. When using a low-inertia haptic device with isotropic force capabilities, an open-loop force controller can be implemented to monitor the force feedback. Note that even if we used impedance control on the robot side, the force/motion-space projection would still be required to compute consistent haptic control inputs.

A. Local robot and haptic device controllers

We use the unified force/motion controller in operational space, as described in [64], to locally control both the motion and force modality on the robot. With this decoupling technique and assuming perfect estimates of the dynamic components, the motion/force loops regulate the desired behavior of the decoupled end effector equivalent to a unit mass system in motion space and a force source in force space. This unified control strategy provides good performances in both motion and contact tasks. For completeness of this paper, we describe the robot controller here. Let us take a robot arm in contact with the environment at its end effector. The dynamic equations of the robot, expressed in joint space, are:

$$A(q)\ddot{q} + b(q,\dot{q}) + g(q) + J^T F_c = \Gamma \qquad (10)$$

where A is the robot mass matrix, b is the vector of centrifugal and coriolis forces, g is the joint gravity vector, J is the task jacobian at the end effector, F_c is the vector of contact forces resolved at the task point and Γ is the joint torque vector. The operational space dynamics (11) of the robot at the end effector is obtained by multiplying the previous equation by the transpose of the dynamically consistent inverse of the Jacobian: $\bar{J}^T = \Lambda J A^{-1}$.

$$\Lambda\ddot{x} + \mu + p + F_c = F \tag{11}$$

where

$$\Lambda = (JA^{-1}J^T)^{-1}$$
$$\mu = \bar{J}^T b - \Lambda\dot{J}\dot{q}$$
$$p = \bar{J}^T g$$

are respectively the task space inertia matrix, the task space centrifugal and coriolis forces and the task space gravity. F is the task force and x is the task space coordinates.

With the robot in contact, its workspace is separated into the force space (directions of constraints) and the motion space (directions of free motion). The force-space and motion-space specification matrices, respectively Ω_f and Ω_m, are obtained via the FSPF. By projecting the robot-proxy interaction, the dual-proxy model computes two setpoints for the robot local controller: the displacement Δx_m on motion space and the desired contact force F_r on force space. With the chosen unified force/motion control approach, two distinct controllers are, then, designed to monitor the robot behavior in the force and motion directions.

The control force in motion space F_m is generated by an operational space proportional derivative (PD) controller:

$$F_m = \hat{\Lambda}(-K_{pm}\Delta x_m - K_{vm}\Omega_m\dot{x}_r) \tag{12}$$

where the estimate of the task space inertia $\hat{\Lambda}$ is used to dynamically decouple the system. K_{pm} is the proportional gain and K_{vm} is the derivative gain.

The control force in force space F_f is computed to follow the desired robot force F_r computed by the dual-proxy model thanks to a proportional integral (PI) controller with feedforward force compensation and a velocity-based damping term. The proportional and integral gains are respectively K_{pf} and K_{if} and the damping gain is K_{vf}. The sensed task force F_s is used as feedback to the PI loop.

$$F_f = F_r - K_{pf}(F_s - F_r) - K_{if}\int(F_s - F_r) - K_{vf}\Omega_f\dot{x}_r \tag{13}$$

Finally, the two controllers are composed together, with two additional gravity and Coriolis/centrifugal compensation terms, to produce the dynamically decoupled and unified force/motion control force F.

$$F = F_f + F_m + \hat{\mu} + \hat{p} \tag{14}$$

$\hat{\mu}$ and $\hat{p}$ respectively represent estimates of the task space Coriolis and centrifugal forces, and of the gravity. The control torques in joint space are, then, simply computed as:

$$\Gamma = J^T F \tag{15}$$

If the robot is redundant with respect to the task, the redundancy can be controlled thanks to a control torque vector Γ_0 projected into the dynamically consistent null-space of the task N such that:

$$\Gamma = J^T F + N^T\Gamma_0 \tag{16}$$

where

$$N = I_{n\times n} - \bar{J}J \tag{17}$$

Such a unified force/motion closed loop at the robot level yields the expected end-effector dynamic behavior to perform the remote task. It monitors the task-consistent control inputs, generated by the dual-proxy model to produce the haptic-robot interactions.

On the haptic side, the haptic local closed loop must reflect the task interaction to the human operator. The desired haptic feedback F_h that was computed by the dual-proxy model, is sent as an input to the local controller. This force command is exclusively within the force space, such that the motion feels fully unconstrained in the motion space. The haptic controller also must compensate for parasitic dynamic components of the haptic device, which can be decently assimilated to the operational space estimated gravity $\hat{p}_h$ when using a low inertia, low friction and parallel haptic device. For such devices, which rarely have a force sensor, open-loop force control is usually satisfactory. We also neglect Coriolis and centrifugal effects as they are small and depend on the human arm dynamics. The haptic control force is computed, with a velocity-based damping term of gain $K_{vf,h}$, by:

$$F_{f,h} = F_h + \hat{p}_h - K_{vf,h}\ \Omega_f\ \dot{x}_h \tag{18}$$

Overall, our full implementation of the local controllers is depicted in Figure 8. It is worth noting that the proxy-based projection method can be used as an input to any local controller, on both sides, in order to achieve optimal performances for different applications. For example, a PI controller could be implemented at the device level to close the force loop, if we have access to a force sensor there, to increase the accuracy in force tracking. In the case of a high-inertia haptic device without force sensor, the haptic control loop could also be directly closed on the haptic-proxy position error. The framework modularity enables adjusting the controllers to the hardware specificities, whatever devices or strategies are used in the teleoperation system.

B. *Stability of the system*

The stability of the global system depends on two factors:

1) The stability of the local controllers
2) The stability of the interaction between the local controllers and the dual-proxy model plus FSPF

We ensure the first by implementing stable controllers. On the haptic side, the open-loop force controller with velocity-based damping is stable by construction. On the robot side, the

PD motion control loop is stable by construction. We need to ensure the stability of the closed-loop force controller. In our implementation, we use a passivity-based explicit force control as described in [65], [66], which provides Lyapunov stability based on a TDPA. In addition, contact transitions could potentially cause force jumps and be a source of instability. Several methods have been shown to effectively prevent vibrations in contact transitions, such as active damping, or force preshaping [67]. Here we use a simple interpolation of the force commands (on both haptic and robot controllers as explained in section III) to prevent the step responses and vibrations. The force interpolation parameters are selected in function of the hardware and virtual stiffness, and independently of the dual-proxy communication rate, which prevent force steps and make contact transitions smoother.

The second is ensured by the timescale difference between the local controllers and their parameterization by the dual-proxy model. Typically, the control loops run at $1kHz$ while the dual-proxy communication and the FSPF run at least ten times slower (typically between $10Hz$ and $100Hz$). This frequency difference ensures a separation of the time scales of the servo loop itself, on the one hand, and the controller parameterization on the other, such that the controller parameters are considered quasi-static compared to the controller frequency. This prevents the introduction of instabilities at the controller parameterization level.

Note that this reasoning stays valid independently of the amount of communication delay. In the presence of large delays, the proxys will be late compared to the robot and haptic positions, but the dual-proxy model will still generate inputs that are consistent with the robot and haptic device capabilities, and the controller parameterization will still be quasi static compared to the frequency of the servo loops. Since the local control loops are not subject to communication delays, there is no reason for the system to go unstable in the presence of delays. This will be shown in the experimental validation.

C. Simulation results

In order to demonstrate the efficiency of the new haptic control method, we perform various simulations and experiments. We use Stanford Robotics Lab's in-house simulation engine, SAI2.0. For the simulations, the robot end effector is modeled as a unit mass sphere. The haptic device is a Force Dimension Omega.7 device. We perform three simulations using the following parameters:

- **dual-proxy model:** the communication runs at $50Hz$, and the amount of delay is variable and will be precised for each experiment. A scaling factor of $s = 2.5$ is used such that $1cm$ in the haptic workspace corresponds to $2.5cm$ in the robot workspace. The proxys are interpolated using an online trajectory generator [59] with a velocity limit of $0.2m.s^{-1}$ and an acceleration limit of $10.0m.s^{-2}$ on the robot side. The virtual spring on the robot side is $k_{vir,r} = 500N.m^{-1}$ while it is $k_{vir,h} = 200N.m^{-1}$ on the haptic side (so that the stiffness ratio is the same as the scaling factor and the forces felt on the haptic side are equal to the applied forces on the robot side). The force interpolation imposes a limit of $50N.s^{-1}$ on the robot side and $20N.s^{-1}$ on the haptic side for the reference force variation.
- **FSPF :** The particle filter runs at a frequency of $15Hz$ with $n_p = 1000$ particles. The parameters of the weight functions are $f_l = 3N, f_h = 10N, v_l = 0.001m.s^{-1}, v_h = 0.005m.s^{-1}$ in the update phase (for the potentially added particles) and $f_l = 0N, f_h = 3N, v_l = 0.001m.s^{-1}, v_h = 0.01m.s^{-1}$ in the resampling phase. The thresholds to add or remove force-space directions are $\alpha_{add} = 0.5$ and $\alpha_{remove} = 0.1$.
- **Robot Controller :** We use P and D gains of $K_{pm} = 100N.m^{-1}$ and $K_{vm} = 20N.m^{-1}.s$ on the motion controller, and P, I and D gains of $K_{pf} = 1.3$, $K_{if} = 1.7s^{-1}$ and $K_{vf} = 20N.m^{-1}.s$ on the force controller.
- **Haptic Controller :** we use a velocity D gain of $K_{vf,h} = 27N.m^{-1}s$ (which corresponds to 90% of the maximum damping of our haptic device) for the velocity-based damping in force space.

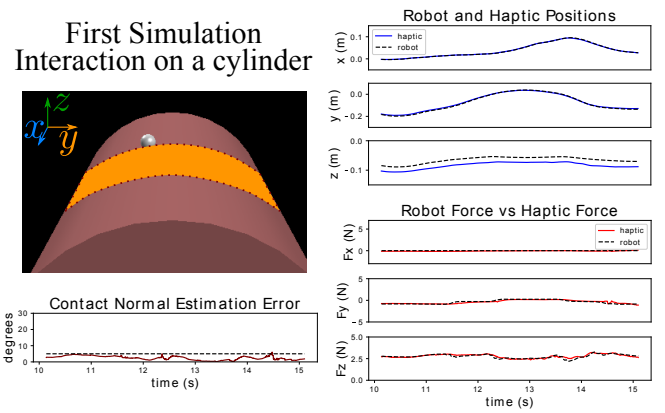

Fig. 9. Top left: screenshot of the simulation rendering. The orange zone is the interaction zone. Bottom left: error in degrees between the force-space direction estimated by the FSPF and the real contact normal. Top right: Positions of the haptic device and robot. Bottom right: haptic and robot forces

1) Simulation 1: Interaction on a cylinder.:
In the first simulation, the robot interacts with a cylinder aligned with the x axis. There is no friction at the contact and no communication delay. The setup and results are shown in Figure 9. The top right plot shows the positions of the haptic device and simulated robot. We see that the position tracking is very good, and there is some error in the z direction, as expected with the proxy method. This error generates a force of approximately $2.5N$ in the z direction. There is a small force that appears and dissapears in the y direction due to the rotation of the force space during the interaction. We see that the force is rendered with very high fidelity. Besides, the force stays zero in the motion direction which enables a very good transparency in the motion space. We also compare the actual force direction (normal to the cylinder) with the one estimated by the FSPF and see that the error stays below a value of $5°$ throughout the interaction. The estimated force space dimension stays equal to 1 during the whole interaction and is not shown here. This simulation shows the validity of the dual-proxy approach in a simple case and the performance

of the particle filter to correctly detect the contact normal and parameterize the controllers.

2) Simulation 2: Interaction on a cylinder and a wall.: In the second simulation, the robot interacts with a cylinder aligned with the x axis and a wall perpendicular to the x axis. There is still no friction at the contact and no communication delay. The setup and results are shown in Figure 10. Once again, the position tracking is good in motion space and the controller generates a high-fidelity feedback force. We now see an error in the x position tracking that results in a force in that direction due to the interaction with the blue wall. This time, we compare the real motion-space direction with the FSPF estimation and it stays below $5°$ once again. The estimated force-space dimension stays equal to 2 throughout the interaction and is not shown in the plots.

3) Simulation 3: Interaction on multiple surfaces with communication delay: In the third scenario, the robot interacts with multiple surfaces. There is no friction, but a communication delay of $50ms$ is added. The setup and results are shown in Figure 11. The position curves show a good tracking. We can observe a horizontal shifting between the two curves due to the delay. The force curves show a high-fidelity force feedback once again, even in the presence of delays. We see a few artifacts in the curve that happen when the robot transitions between the two surfaces and the force-space dimension changes, but these are quickly eliminated by the controller. Finally, we show the dimension of the force space and the state of the particle filter on the bottom left. We see that the robot is in free space at the start, then it interacts on the floor, the obstacle, the floor again, and finally with both at the same time on the edge. The particle filter is able to detect each contact interaction properly and to parameterize the controllers accordingly, even in the presence of delays. The first part of the video shows the rendering of this simulation.

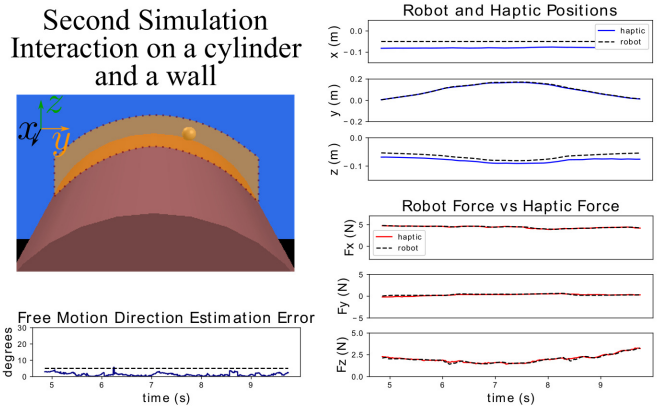

Fig. 10. Top left: screenshot of the simulation rendering. The orange zone is the interaction zone. Bottom left: error in degrees between the motion space direction estimated by the FSPF and the real motion axis (tangent to the cylinder in the plane of the blue wall). Top right: Positions of the haptic device and robot. Bottom right: haptic and robot forces

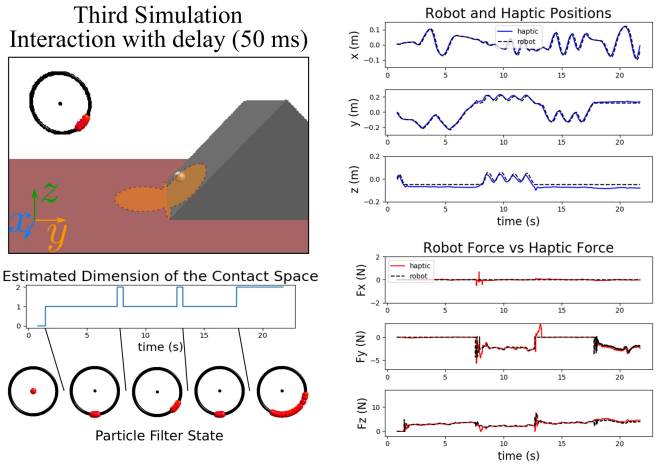

Fig. 11. Top left: screenshot of the simulation rendering. The orange zone is the interaction zone. Bottom left: Dimension of the force space as detected by the FSPF and state of the particle filter throughout the different phases of the interaction. Top right: Positions of the haptic device and robot. Bottom right: haptic and robot forces

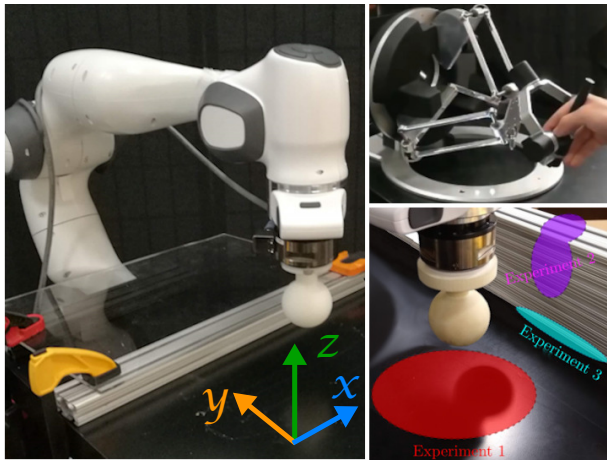

Fig. 12. Experimental setup

VI. Experimental Validation

Three experiments were realized to test the framework on a real teleoperation system. We remotely control a 7-DoF Franka Panda robot, equipped with an ATI Gamma force sensor at its end effector, through the Omega.7 haptic device. The software uses the custom control library developed at Stanford Robotics Lab as part of SAI2.0 that communicates with the torque control interface of the Franka Panda robot through the libfranka library provided by the manufacturer, and with the haptic device through the Force Dimension Chai3D drivers. The experimental setup, common to the three experiments is depicted on Figure 12. In the first experiment, the robot interacts with the horizontal surface (red area). In the second experiment, the robot interacts on the upper corner of the wall (purple area). In the third experiment, the robot interacts on the corner between the wall and the horizontal surface (cyan area). The parameters for the dual-proxy model, particle filter, and haptic controller are the same as in the simulations. The robot controller now uses a 6-DoF position

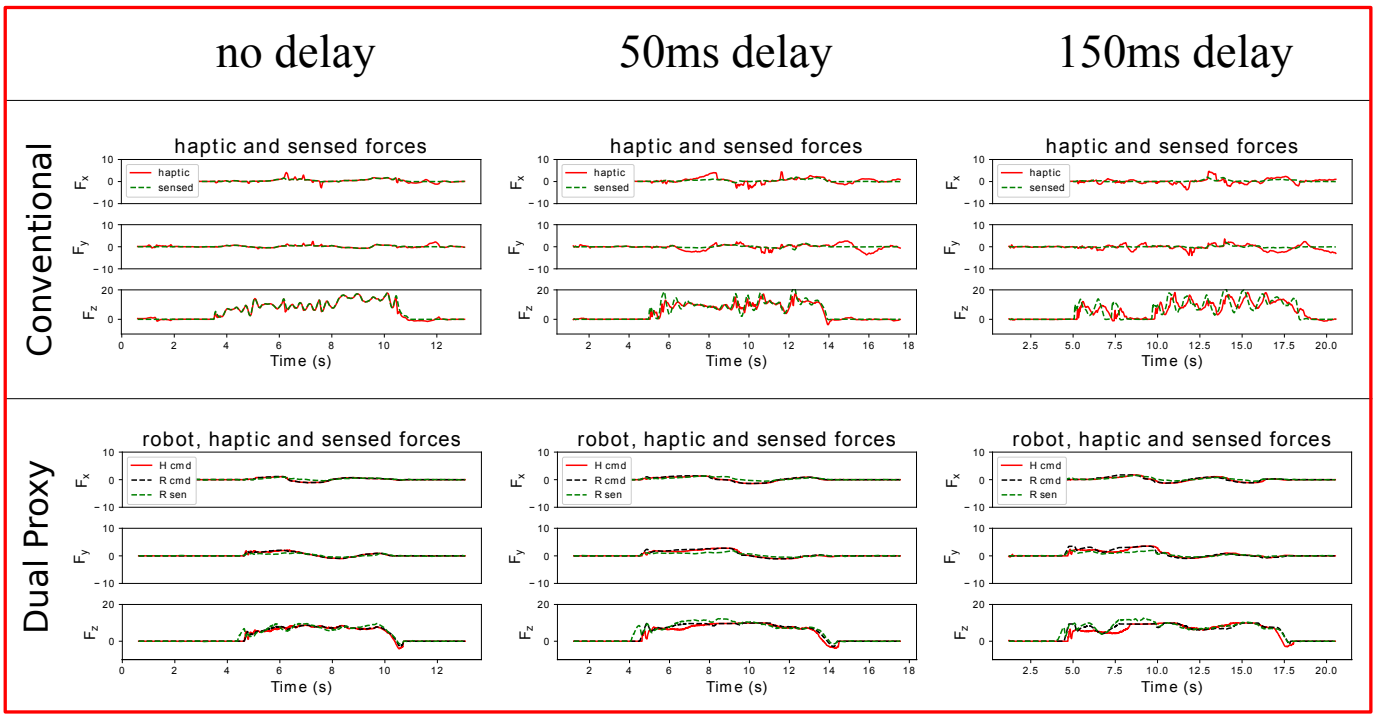

Fig. 13. First experiment, comparison between the conventional and the dual-proxy approaches for force tracking

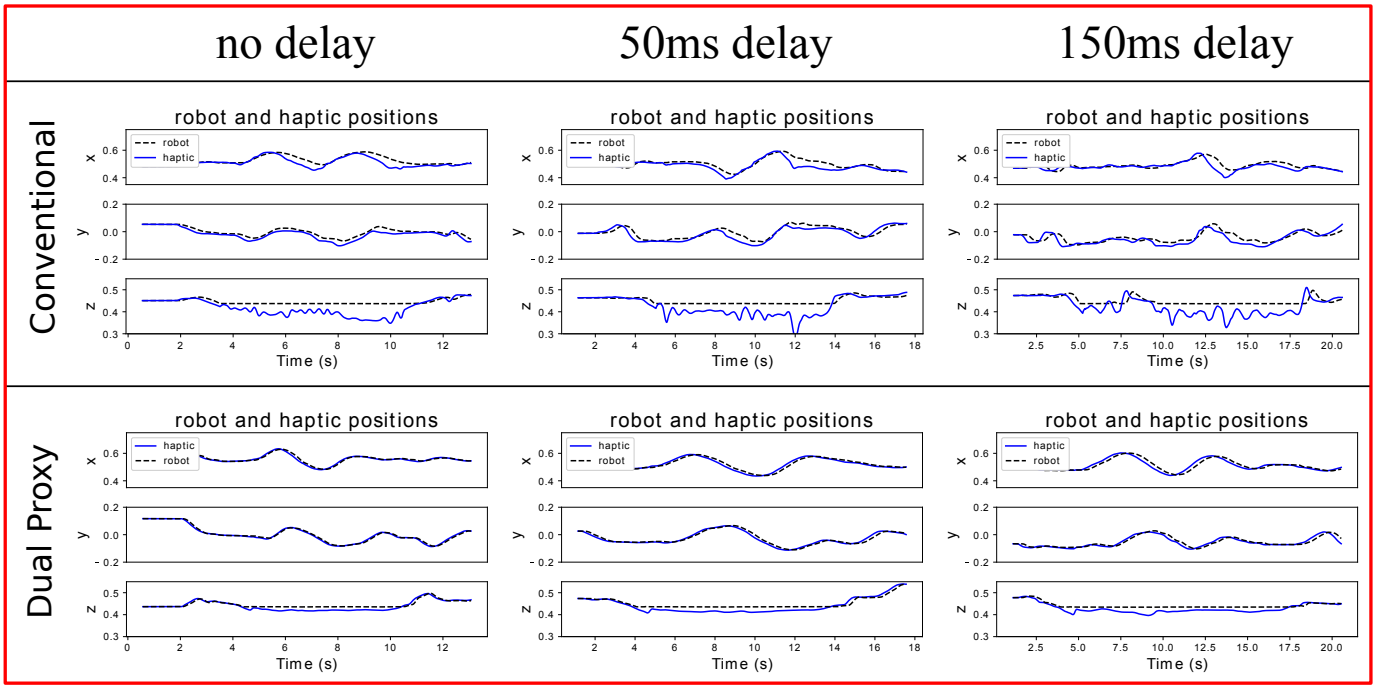

Fig. 14. First experiment, comparison between the conventional and the dual-proxy approaches for position tracking

plus orientation controller. The orientation is controlled to stay constant using a PD controller in operational space with a P gain of $600N.rad^{-1}$ and a D gain of $38N.rad^{-1}.s$. For the position part, the motion controller uses the gains $K_{pm} = 500N.m^{-1}, K_{vm} = 40N.m^{-1}.s$ and the force part uses the gains $K_{pf} = 2.5, K_{if} = 5.5s^{-1}, K_{vf} = 20N.m^{-1}.s$.

A. First experiment: Interaction with a plane

In this experiment, the robot starts in free space. It is haptically controlled to go into contact with a table (horizontal surface), to move on the table, and to go back to free space. We perform this experiment with increasing amounts of communication delays between the robot and the haptic device. In order to provide a qualitative comparison with a conventional approach, we also implement a two-channel force-position bilateral haptic controller in which the haptic position is used as the input to the robot operational-space motion controller, and the sensed force on the robot side is directly sent as the force command to the haptic device (up to the workspace mapping). We also implement a basic bilateral TDPA [37], which is a commonly used method in the haptic field to stabilize teleoperation systems. The gains of the conventional bilateral controller are tuned to get approximately the same robot stiffness in contact as the one implemented with the dual-proxy approach, resulting in gains of $K_{pm} = 100N.m^{-1}$ and $K_{vm} = 20N.m^{-1}.s$.

The forces throughout the first experiment are plotted in Figure 13 for different communication delays varying from $0ms$ to $150ms$. The conventional approach is plotted on top and the dual-proxy method on the bottom. The z axis is the constraint direction (normal to the plane of interaction) and the x and y directions are free. With the conventional approach, there are oscillations in the contact force that are due to the non collocation between the human hand and the robot end effector, and to force filtering and communication delays. As the time delay increases, an error starts appearing between the robot sensed force and the haptic force, which worsen the oscillations. With the dual-proxy approach the contact forces are much smoother. We plot the haptic command forces, the robot command forces, and the robot sensed forces. The three plots are almost superimposed, with the only mismatch happening during the contact transition. In addition, there are almost no forces felt in the x and y directions, as expected for the directions of free motion. The FPSF correctly detects the z axis as the contact direction, with an average error of 10 degrees throughout all the experiments. This error is higher than the one in simulation because the friction between the robot and the surface is not taken into account by the FSPF in this implementation, but it does not prevent the correct execution of the task and sensing of the contact directions versus free space directions.

Figure 14 repeats the comparison for the position tracking with the same delays. The conventional approach in on top and the dual proxy on the bottom row. With the conventional approach, there are small oscillations in the position of the haptic device in the constraint direction (that are related to the force oscillations on the human hand). These oscillations become worse as the delay increase. This phenomenon does not happen at all with the dual-proxy method. In addition, because the conventional approach was tuned to perform well in contact, its position tracking in the free-motion directions is less accurate as it is not able to compensate well enough the friction at the robot contact point. In contrast, the dual proxy enables an independent tuning of the behavior in motion space and force space, so it can achieve a very good position tracking (up to the communication delay) independently of the selected compliance in force space.

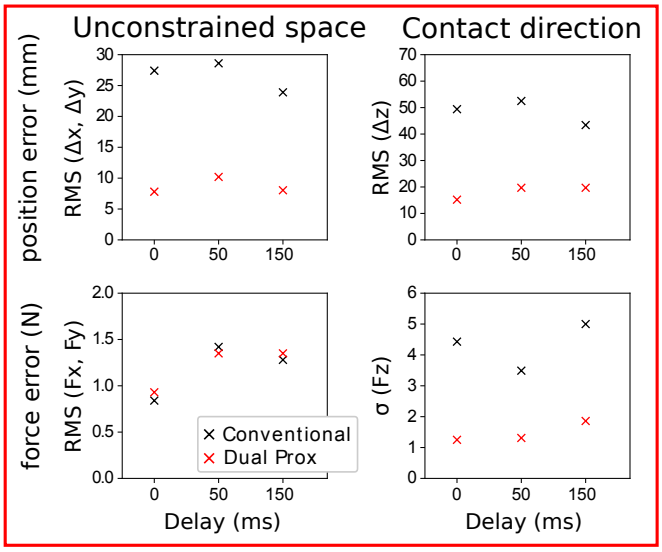

Fig. 15. First experiment, comparison between the force and position errors during the contact phase for the conventional and the dual-proxy approaches

It is worth noting that using extended TDPA [40], [41] would exhibit less oscillations, by better estimating the delayed passivity conditions. But it would result in large force-position errors as the delay increases, since the passivity controllers dissipate active energy through a time-varying damping element. This damping element results in slowing down the slave robot, inducing a position drift, and either sudden haptic force changes or force distortions [40]. In addition, TDPA assumes that the exact needed amount of energy can be dissipated at each time step. However, passivizing complex multi-DOF teleoperation systems with a high-effective-inertia robot, as in our experimental setup, is not trivial and could require too much damping which would rather destabilize the system.

For numerical comparison purpose, we compute the Root Mean Square (RMS) values of the haptic-robot motion-tracking errors, during the interaction phase, both in free-space (mean of x and y position errors) and in contact (z direction). The force fidelity is evaluated through the RMS value of the haptic force in free-space, which must stay close to zero (unconstrained motion), and through the standard deviation σ of the z-force feedback, which is equal to zero if the contact force stays constant on the plane (smooth interaction). Values of these numerical criteria are given in Figure (15) for the two control methods, with respect to the communication delay. The force error in the unconstrained space is similar for both the conventional and the dual-proxy approaches, and stays

relatively small (close to $1N$). In contrast, motion-tracking errors are notably smaller with the dual-proxy method, both in unconstrained space and in contact. The dual-proxy mean position error in the contact direction stays around $15-20mm$ during the interaction and is directly related to the virtual spring used to generate the haptic force. Position errors in the unconstrained space stay below $10mm$ which shows the robust motion tracking behavior achieved with our approach. The low standard deviation of the haptic force in the contact space, smaller than $2N$ over the different experiments, demonstrates the stable force behavior of the teleoperation system with the dual-proxy approach.

We repeat the experiment for delays up to $1500ms$. The force and position tracking plots for the dual-proxy approach are plotted on Figure 16. Even for high delay values, the task is achievable and the oscillations are limited to the contact transition phases. With the conventional approach, such high delay made the task impossible to realize and the robot was constantly losing contact.

This first experiment shows the performances of the dual-proxy model both in free space and in contact space, and the quality of the haptic interaction. As the communication delay increases, the tracking stays effective and the force feedback stays relevant.

B. Second experiment: Interaction with an outer corner

In this experiment, the robot starts in free space, touches the wall (normal to the y axis) and interacts on the outer corner of the experimental setup to end up on the top of the wall (contact aligned with the z axis), while staying in contact the whole time. Figure 17 shows the plots for three values of communication delay (up to $350ms$). In the experiment with no delay, the robot comes back on the side of the wall after reaching the top, while on the other two experiments, it finishes on top of the wall. The other values of delay are not shown for compactness because the plots are all very similar. Once again, the position and force tracking is very good, independently of the amount of communication delay. The forces vary continuously between the y axis and the z axis as the contact normal changes and is correctly detected by the FSPF. The force transition between these two directions is done smoothly and without contact losses. The position tracking shows a well regulated motion along the z and y axes as the robot is commanded to move on the corner. This experiment displays the capability of the dual-proxy approach to allow precise interaction with non-flat surfaces, irrespective of the amount of communication delay.

C. Third experiment : Interaction with an inner corner

In this experiment, the robot interacts within the inner corner between the wall and the table. More precisely, it starts in single contact, goes to the corner, slides along the corner, and return to single contact with the wall or the table. Once again, we only show three values of communication delay. The results are plotted in Figure 18. The normal to the wall is the y axis, the normal to the table is the z axis, and the x axis is the free-motion direction during the double-contact phase. The contact-space dimension is correctly detected, and the orientation error average of the estimated contact direction stays around 10 degrees throughout all the experiments. This results in a good motion tracking in the motion space (x direction) and a force feedback in the force space along the y and z axes that renders the geometry of the contact accurately. This experiment showcases the ability of the dual-proxy approach to allow robot interactions with multiple surfaces, and to correctly render contact transitions.

D. Controlling a robot in China from the US

A qualitative experiment was also performed using the dual proxy model where a Panda robot in Chengdu (China) was controlled from Stanford (USA) via the internet using an Omega7 haptic device. The controllers used are the same as the other three experiments. The dual proxy model and communication runs at $5\ Hz$, the delay is around $300\ ms$ and the system will render no haptic forces in case of communication loss as a safety. The robot interacts with a replica of a human torso and the experiment consists in simulating an ultrasound imaging performed by the robot controlled remotely. The video shows the stability and performance of the system. The user was able to feel the torso, and the corner between the chin and neck, even with the presence of delays in the visual feedback (done using zoom) and communication.

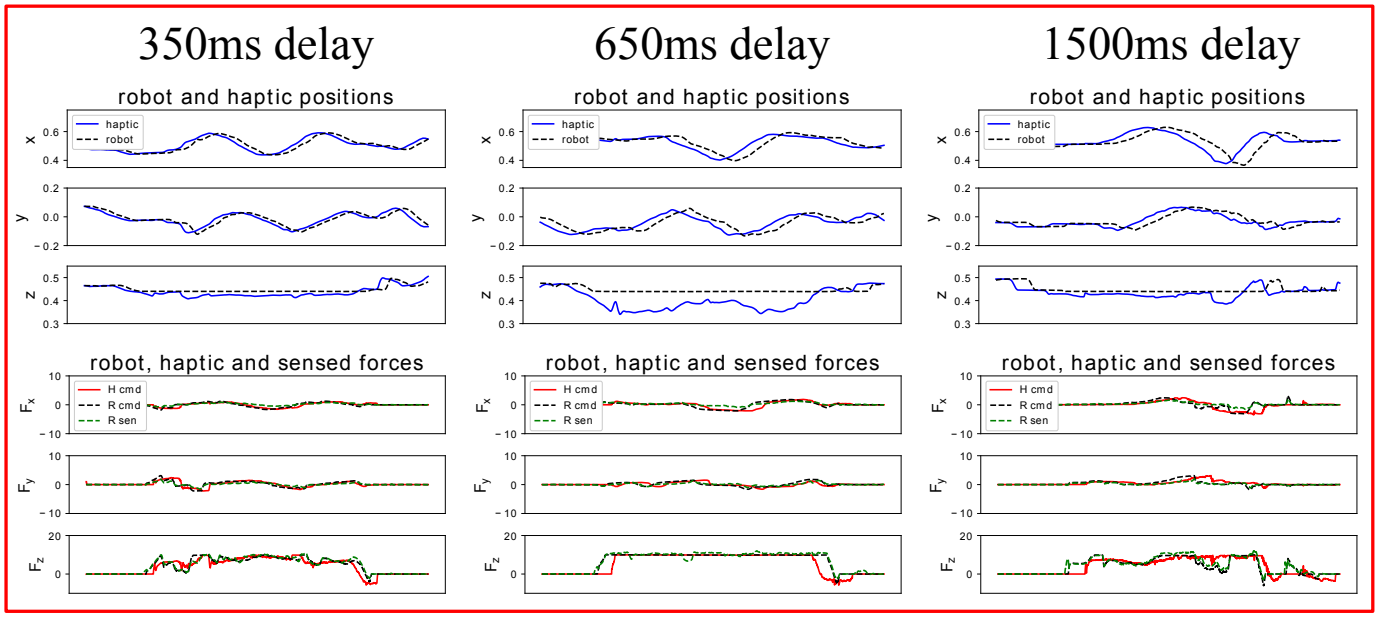

Fig. 16. First experiment with high delay

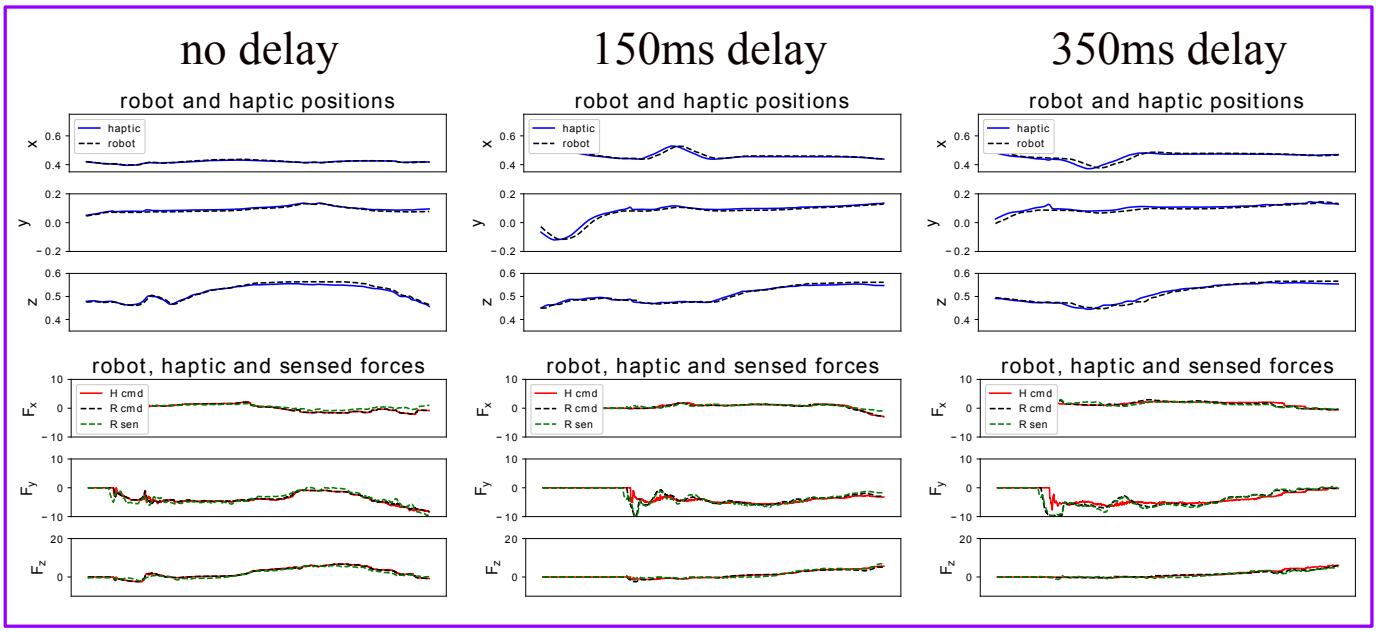

Fig. 17. Second experiment, interaction with an outer corner: Positions and forces

VII. Conclusions

We have presented a novel approach to haptic-robot control based on local autonomy and a dual-proxy model. The dual proxy guarantees the generation of safe and consistent commands for the two local controllers, and the local controllers ensure the compliance and stability of the systems on both sides. A Force-Space Particle Filter was developed to enable autonomous modeling and rendering of the task contact geometry from the robot state and sensory data. The method leverages robot functional autonomy and automatically suppresses the instability issues caused by the transfer of power variables for low-level control through a network with communication delays in conventional haptic-robot controllers. We have validated the method experimentally in multiple contact situations. The results demonstrated the transparency and high fidelity of the dual-proxy method and its robustness to communication delays. The dual-proxy method maintained high performance for delays of up to one and a half seconds. The local autonomy-based haptic control of robots with the dual-proxy model enables many applications in areas such as medical procedures, underwater interactions, and space robotics.

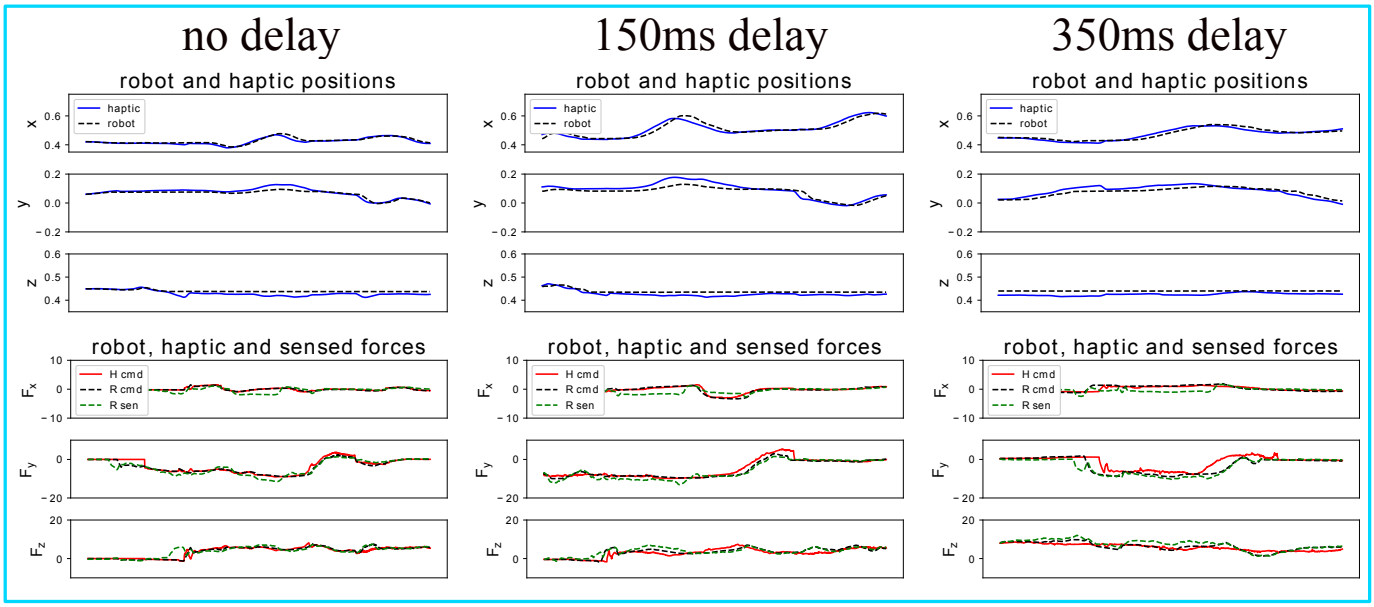

Fig. 18. Third experiment, interaction with an inner corner: Positions and forces

REFERENCES

[1] N. Hogan, "Impedance control: An approach to manipulation," in *1984 American control conference*. IEEE, 1984, pp. 304–313.

[2] B. Willaert, B. Corteville, D. Reynaerts, H. Van Brussel, and E. B. Vander Poorten, "A mechatronic analysis of the classical position-force controller based on bounded environment passivity," *The International Journal of Robotics Research*, vol. 30, no. 4, pp. 444–462, 2011.

[3] M. Bergamasco, B. Allotta, L. Bosio, L. Ferretti, G. Parrini, G. Prisco, F. Salsedo, and G. Sartini, "An arm exoskeleton system for teleoperation and virtual environments applications," in *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*. IEEE, 1994, pp. 1449–1454.

[4] J. Maples and J. Becker, "Experiments in force control of robotic manipulators," in *Proceedings. 1986 IEEE International Conference on Robotics and Automation*, vol. 3. IEEE, 1986, pp. 695–702.

[5] C. Carignan and K. Cleary, "Closed-loop force control for haptic simulation of virtual environments," *Haptics-e, The electronic journal of haptics research*, 2000.

[6] C. Ott, R. Mukherjee, and Y. Nakamura, "A hybrid system framework for unified impedance and admittance control," *Journal of Intelligent & Robotic Systems*, vol. 78, no. 3-4, pp. 359–375, 2015.

[7] H. Kim, J. Kwon, Y. Oh, B. J. You, and W. Yang, "Weighted hybrid admittance-impedance control with human intention based stiffness estimation for human-robot interaction," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1–6.

[8] D. A. Lawrence, "Stability and transparency in bilateral teleoperation," *IEEE transactions on robotics and automation*, vol. 9, no. 5, pp. 624–637, 1993.

[9] S. E. Salcudean, M. Zhu, W.-H. Zhu, and K. Hashtrudi-Zaad, "Transparent bilateral teleoperation under position and rate control," *The International Journal of Robotics Research*, vol. 19, no. 12, pp. 1185–1202, 2000.

[10] K. Hashtrudi-Zaad and S. E. Salcudean, "Analysis of control architectures for teleoperation systems with impedance/admittance master and slave manipulators," *The International Journal of Robotics Research*, vol. 20, no. 6, pp. 419–445, 2001.

[11] I. Aliaga, A. Rubio, and E. Sanchez, "Experimental quantitative comparison of different control architectures for master-slave teleoperation," *IEEE transactions on control systems technology*, vol. 12, no. 1, pp. 2–11, 2004.

[12] S. Hirche and M. Buss, "Human-oriented control for haptic teleoperation," *Proceedings of the IEEE*, vol. 100, no. 3, pp. 623–647, 2012.

[13] J. J. Gil, A. Rubio, and J. Savall, "Decreasing the apparent inertia of an impedance haptic device by using force feedforward," *IEEE Transactions on Control Systems Technology*, vol. 17, no. 4, pp. 833–838, 2009.

[14] J. S. Hart and G. Niemeyer, "Multi-dof model-based force control for telerobotic applications," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2010, pp. 5282–5287.

[15] B. Hannaford, "Stability and performance tradeoffs in bi-lateral telemanipulation." in *IEEE International Conference on Robotics and Automation (ICRA)*, 1989, pp. 1764–1767.

[16] J. E. Colgate and J. M. Brown, "Factors affecting the z-width of a haptic display," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 1994, pp. 3205–3210.

[17] R. Daniel and P. R. McAree, "Fundamental limits of performance for force reflecting teleoperation," *The International Journal of Robotics Research*, vol. 17, no. 8, pp. 811–830, 1998.

[18] K. Hashtrudi-Zaad and S. E. Salcudean, "Transparency in time-delayed systems and the effect of local force feedback for transparent teleoperation," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 1, pp. 108–114, 2002.

[19] K. Hashtrudi Zaad and S. E. Salcudean, "Bilateral parallel force/position teleoperation control," *Journal of Robotic Systems*, vol. 19, no. 4, pp. 155–167, 2002.

[20] P. Malysz and S. Sirouspour, "Nonlinear and filtered force/position mappings in bilateral teleoperation with application to enhanced stiffness discrimination," *IEEE Transactions on Robotics*, vol. 25, no. 5, pp. 1134–1149, 2009.

[21] J. Guo, C. Liu, and P. Poignet, "A scaled bilateral teleoperation system for robotic-assisted surgery with time delay," *Journal of Intelligent & Robotic Systems*, vol. 95, no. 1, pp. 165–192, 2019.

[22] R. Cortesão, J. Park, and O. Khatib, "Real-time adaptive control for haptic telemanipulation with kalman active observers," *IEEE Transactions on Robotics*, vol. 22, no. 5, pp. 987–999, 2006.

[23] J. Park and O. Khatib, "A haptic teleoperation approach based on contact force control," *The International Journal of Robotics Research*, vol. 25, no. 5-6, pp. 575–591, 2006.

[24] T. Tsumugiwa, R. Yokogawa, and K. Hara, "Variable impedance control based on estimation of human arm stiffness for human-robot cooperative calligraphic task," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, vol. 1. IEEE, 2002, pp. 644–650.

[25] A. Shahdi and S. Sirouspour, "Adaptive/robust control for time-delay teleoperation," *IEEE Transactions on Robotics*, vol. 25, no. 1, pp. 196–205, 2009.

[26] Y.-C. Liu and M.-H. Khong, "Adaptive control for nonlinear teleoperators with uncertain kinematics and dynamics," *IEEE/ASME Transactions on Mechatronics*, vol. 20, no. 5, pp. 2550–2562, 2015.

[27] D.-H. Zhai and Y. Xia, "Adaptive control for teleoperation system with

varying time delays and input saturation constraints," *IEEE Transactions on industrial electronics*, vol. 63, no. 11, pp. 6921–6929, 2016.
[28] E. Nuño, L. Basañez, and R. Ortega, "Passivity-based control for bilateral teleoperation: A tutorial," *Automatica*, vol. 47, no. 3, pp. 485–495, 2011.
[29] P. M. Kebria, H. Abdi, M. M. Dalvand, A. Khosravi, and S. Nahavandi, "Control methods for internet-based teleoperation systems: A review," *IEEE Transactions on Human-Machine Systems*, vol. 49, no. 1, pp. 32–46, 2018.
[30] R. J. Anderson and M. W. Spong, "Asymptotic stability for force reflecting teleoperators with time delay," *The International Journal of Robotics Research*, vol. 11, no. 2, pp. 135–149, 1992.
[31] G. Niemeyer and J.-J. Slotine, "Towards force-reflecting teleoperation over the internet," in *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 3. IEEE, 1998, pp. 1909–1915.
[32] D. Sun, F. Naghdy, and H. Du, "Wave-variable-based passivity control of four-channel nonlinear bilateral teleoperation system under time delays," *IEEE/ASME transactions on mechatronics*, vol. 21, no. 1, pp. 238–253, 2015.
[33] D.-W. Sun, F. Naghdy, and H. Du, "Application of wave-variable control to bilateral teleoperation systems: A survey," *Annual Reviews in Control*, vol. 38, no. 1, pp. 12–31, 2014.
[34] E. J. Rodriguez-Seda, D. Lee, and M. W. Spong, "Experimental comparison study of control architectures for bilateral teleoperators," *IEEE Transactions on robotics*, vol. 25, no. 6, pp. 1304–1318, 2009.
[35] F. Ferraguti, M. Bonfe, C. Fantuzzi, and C. Secchi, "Optimized power modulation in wave based bilateral teleoperation," *IEEE/ASME Transactions on Mechatronics*, 2020.
[36] B. Hannaford and J.-H. Ryu, "Time-domain passivity control of haptic interfaces," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 1, pp. 1–10, 2002.
[37] J.-H. Ryu, D.-S. Kwon, and B. Hannaford, "Stable teleoperation with time-domain passivity control," *IEEE Transactions on robotics and automation*, vol. 20, no. 2, pp. 365–373, 2004.
[38] J. Artigas, J.-H. Ryu, and C. Preusche, "Time domain passivity control for position-position teleoperation architectures," *Presence: Teleoperators and Virtual Environments*, vol. 19, no. 5, pp. 482–497, 2010.
[39] J. Rebelo and A. Schiele, "Time domain passivity controller for 4-channel time-delay bilateral teleoperation," *IEEE transactions on haptics*, vol. 8, no. 1, pp. 79–89, 2014.
[40] J.-H. Ryu, J. Artigas, and C. Preusche, "A passive bilateral control scheme for a teleoperator with time-varying communication delay," *Mechatronics*, vol. 20, no. 7, pp. 812–823, 2010.
[41] M. Panzirsch, J.-H. Ryu, and M. Ferre, "Reducing the conservatism of the time domain passivity approach through consideration of energy reflection in delayed coupled network systems," *Mechatronics*, vol. 58, pp. 58–69, 2019.
[42] R. Balachandran, J. Artigas, U. Mehmood, and J.-H. Ryu, "Performance comparison of wave variable transformation and time domain passivity approaches for time-delayed teleoperation: Preliminary results," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 410–417.
[43] N. Diolaiti, G. Niemeyer, F. Barbagli, and J. K. Salisbury, "A criterion for the passivity of haptic devices," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2005, pp. 2452–2457.
[44] K. Hosseini-Suny, H. Momeni, and F. Janabi-Sharifi, "A modified adaptive controller design for teleoperation systems," *Robotics and autonomous systems*, vol. 58, no. 5, pp. 676–683, 2010.
[45] H. Singh, M. Rothammer, C.-I. Lee, T. Hulin, J.-H. Ryu, and C. Ott, "Ensuring stable and transparent high stiffness haptic interaction using successive force augmention with time domain passivity approach," in *International Symposium on Experimental Robotics*. Springer, 2020, pp. 263–273.
[46] V. Chawda, H. Van Quang, M. K. O'Malley, and J.-H. Ryu, "Compensating position drift in time domain passivity approach based teleoperation," in *2014 IEEE Haptics Symposium (HAPTICS)*. IEEE, 2014, pp. 195–202.
[47] V. Chawda and M. K. O'Malley, "Position synchronization in bilateral teleoperation under time-varying communication delays," *Ieee/asme transactions on mechatronics*, vol. 20, no. 1, pp. 245–253, 2014.
[48] M. Franken, S. Stramigioli, S. Misra, C. Secchi, and A. Macchelli, "Bilateral telemanipulation with time delays: A two-layer approach combining passivity and transparency," *IEEE transactions on robotics*, vol. 27, no. 4, pp. 741–756, 2011.
[49] G. Christiansson and F. Van Der Helm, "The low-stiffness teleoperator slave—a trade-off between stability and performance," *The International Journal of Robotics Research*, vol. 26, no. 3, pp. 287–299, 2007.
[50] E. Naerum, O. J. Elle, and B. Hannaford, "The effect of interaction force estimation on performance in bilateral teleoperation," *IEEE transactions on haptics*, vol. 5, no. 2, pp. 160–171, 2011.
[51] B. Willaert, D. Reynaerts, H. Van Brussel, and E. B. Vander Poorten, "Bilateral teleoperation: Quantifying the requirements for and restrictions of ideal transparency," *IEEE transactions on control systems technology*, vol. 22, no. 1, pp. 387–395, 2013.
[52] B. Hannaford, "A design framework for teleoperators with kinesthetic feedback," *IEEE transactions on Robotics and Automation*, vol. 5, no. 4, pp. 426–434, 1989.
[53] P. Mitra and G. Niemeyer, "Model-mediated telemanipulation," *The International Journal of Robotics Research*, vol. 27, no. 2, pp. 253–262, 2008.
[54] X. Xu, B. Cizmeci, C. Schuwerk, and E. Steinbach, "Model-mediated teleoperation: toward stable and transparent teleoperation systems," *IEEE Access*, vol. 4, pp. 425–449, 2016.
[55] D. C. Ruspini, K. Kolarov, and O. Khatib, "The haptic display of complex graphical environments," in *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, 1997, pp. 345–352.
[56] D. Ruspini, K. Kolarov, and O. Khatib, "Haptic interaction in virtual environments," in *Proceedings of the 1997 IEEE/RSJ International Conference on Intelligent Robot and Systems. Innovative Robotics for Real-World Applications. IROS'97*, vol. 1. IEEE, 1997, pp. 128–133.
[57] D. Ruspini and O. Khatib, "Haptic display for human interaction with virtual dynamic environments," *Journal of Robotic Systems*, vol. 18, no. 12, pp. 769–783, 2001.
[58] C. B. Zilles and J. K. Salisbury, "A constraint-based god-object method for haptic display," in *IEEE/RSJ international conference on intelligent robots and systems. Human robot interaction and cooperative robots*, vol. 3. IEEE, 1995, pp. 146–151.
[59] T. Kröger, *On-Line Trajectory Generation in Robotic Systems: Basic Concepts for Instantaneous Reactions to Unforeseen (Sensor) Events*. Springer, 2010, vol. 58.
[60] D. Verscheure, J. Swevers, H. Bruyninckx, and J. De Schutter, "On-line identification of contact dynamics in the presence of geometric uncertainties," in *2008 IEEE International Conference on Robotics and Automation*. IEEE, 2008, pp. 851–856.
[61] S. Thrun, W. Burgard, and D. Fox, "Probabilistic robotics, vol. 1," 2005.
[62] B. Willaert, J. Bohg, H. Van Brussel, and G. Niemeyer, "Towards multi-dof model mediated teleoperation: Using vision to augment feedback," in *2012 IEEE International Workshop on Haptic Audio Visual Environments and Games (HAVE 2012) Proceedings*. IEEE, 2012, pp. 25–31.
[63] N. Hogan, "Stable execution of contact tasks using impedance control," in *Proceedings. 1987 IEEE International Conference on Robotics and Automation*, vol. 4. IEEE, 1987, pp. 1047–1054.
[64] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE Journal on Robotics and Automation*, vol. 3, no. 1, pp. 43–53, 1987.
[65] R. Balachandran, M. Jorda, J. Artigas, J.-H. Ryu, and O. Khatib, "Passivity-based stability in explicit force control of robots," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 386–393.
[66] M. Jorda, R. Balachandran, J.-H. Ryu, and O. Khatib, "New passivity observers for improved robot force control," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 2177–2184.
[67] J. M. Hyde and M. R. Cutkosky, "Controlling contact transition," *IEEE Control Systems Magazine*, vol. 14, no. 1, pp. 25–30, 1994.

Mikael Jorda Mikael Jorda received his PhD degree in Robotics from the Robotics Laboratory at Stanford University in 2021. He graduated from Stanford University with a M.S. degree in Mechanical Engineering in 2016, and from École Polytechnique (Palaiseau) with a Masters degree in Mechanical Engineering and Computer Science in 2015. His research interests include Robotic Manipulation, Force Control, Multi-Contact Physical Interactions, Haptic Teleoperation and Grasping.

Margot Vulliez Margot Vulliez is an Associate Professor of Mechanical Engineering at Pprime Institute (CNRS, University of Poitiers) within the RoBioSS team. She was a postdoctoral scholar in the Robotics Laboratory at Stanford University in 2019. She received her PhD degree in Robotics from University of Poitiers in 2018. She graduated from the Ecole Normale Supérieure de Cachan with a M.S. degree in Advanced Systems and Robotics (2015), and a teacher certification and a M.S. degree of Faculty Training for Education in Mechanical Engineering (2014). Her research interests include Mechatronic Design, Haptics, Human-Robot Interaction, and Task-level Robot Control.

Oussama Khatib Oussama Khatib received his PhD from Sup'Aero, Toulouse, France, in 1980. He is Professor of Computer Science and Director of the Robotics Laboratory at Stanford University. His research focuses on methodologies and technologies in human-centered robotics, haptic interactions, artificial intelligence, human motion synthesis and animation. He is President of the International Foundation of Robotics Research (IFRR) and a Fellow of the Institute of Electrical and Electronic Engineers (IEEE). He is Editor of the Springer Tracts in Advanced Robotics (STAR) series, and the Springer Handbook of Robotics, awarded the American Publishers Award for Excellence in Physical Sciences and Mathematics. He is recipient of the IEEE Robotics and Automation (IEEE/RAS) Pioneering Award (for his fundamental contributions in robotics research, visionary leadership and life-long commitment to the field), the IEEE/RAS George Saridis Leadership Award, the Distinguished Service Award, the Japan Robot Association (JARA) Award, the Rudolf Kalman Award, and the IEEE Technical Field Award. Professor Khatib is a member of the National Academy of Engineering.