A Reconfigurable Interface for Ergonomic and Dynamic Tele-Locomanipulation

Soheil Gholami*[1,2], Francesco Tassi*[1,2], Elena De Momi[2], and Arash Ajoudani[1]

Abstract— **Prolonged remote tele-locomanipulation of multi degrees-of-freedom mobile manipulators requires a compromise between the system's performance and the operator's ergonomics. Neglecting this demand can significantly affect either the task completion or the level of comfort to achieve it. However, the simultaneous consideration of these key factors has received less attention in the literature. To respond to this demand, in this work, we introduce a new teleoperation setup, which integrates the features of an ergonomic and a highly maneuverable interface into a unified solution. The ergonomic part of the interface implements a 3D mouse-like functionality, enabling the execution of long navigation tasks for the floating base. The highly manoeuvrable interface instead, enables the operator to perform dynamic or more precise manipulation by moving his/her arm in space. The locomotion and manipulation modes of the follower robot are controlled separately, which can be easily and seamlessly switched by the operator by pressing a button at any moment. Furthermore, due to the follower manipulator's redundancy, this robot is controlled by a hierarchical quadratic programming technique which enables the definition of a set of secondary tasks to be executed in the robot's nullspace. Finally, to demonstrate the advantages and disadvantages of the proposed user interfaces, five participants are asked to perform two different experiments: (i) target selection task on a moving surface and (ii) remote path tracking on a fixed surface. The quantitative and qualitative analyses show the effectiveness of the proposed interface during the teleoperation tasks, especially when it comes to the precise and dynamic task execution.**

I. INTRODUCTION

Over the past decades, teleoperation systems have captured the attention of many research studies from a theoretical point of view. In particular, the stability and tracking [1], communication [2], autonomy [3], and interaction [4], are the topics covered broadly in the literature. Besides, the recent advancements in hardware and computing technologies have also opened up new possibilities such as extended realities [5] and learning techniques [6] to improve the remote execution of multi-Degrees-of-Freedom (DoF) robotic tasks. Despite these advancements, the usability of the teleoperation interfaces, in terms of ergonomics, ease-of-use, and the achieved performances, when controlling multi-DoF follower robots has received little attention.

Recent studies have made an attempt to approach these usability factors separately (e.g., see [3], [7], [8]), often resulting in unbalanced solutions. For instance, when designing comfortable interfaces for prolonged/repetitive operations (e.g., by using grounded systems such as a 3D mouse [9]), the maneuverability and the dynamic response of the operator are undermined. On the contrary, agile interfaces that exploit, for instance, body tracking suits [10], are inefficient for everyday or prolonged operations due to the one-to-one replication of the operator movements and the underlying fatigue. Indeed, a compromise between the system's performance and ease-of-use, and the operator's ergonomics must be reached to confront this big challenge.

A. *Related work*

Regarding the few available works on the ergonomics evaluation of teleoperation interfaces, the authors in [7] explored the ergonomics aspects of their developed setup for minimally invasive surgery through natural orifices. This includes the surgeon's pose, movement scaling, visual feedback, and haptic feedback. For the performance analyses, on the other hand, different criteria and interpretations are investigated by the researchers. For instance, in [11], the authors investigated the effects of shared autonomy on the user's physical workload during a whole-body motion-mapping teleoperation of a complex robot based on their previous studies on the muscles' fatigue [12]. As a result, they reported better user experience and task performance, which are justified by a set of subjective and objective indices. Besides, the virtual fixture approach was employed in [13] to enhance the system's control performance which, in turn, reduced the operator's pressure during the remote tasks, evaluated by tracking errors and task execution time.

Moreover, the "teleoperation manipulability index" was introduced in [14] as a quantifiable manipulability measure of leader-follower surgical systems, focusing on both design and performance levels. Using this in a simulation environment, an overall performance improvement was reported in terms of surgeon's control over the follower robot's force and velocity, leader robot's footprint, surgeon's control effort, and singularity avoidance. Also, [15] proposed a manipulability optimisation control method of a redundant manipulator to be used in robot-assisted minimally invasive surgeries. Here, the Manipulability Index (MI) and keeping the Remote Center of Motion (RCM) fixed were the constraints of the optimisation problem. Consequently, the system's performance was assessed in respect of MI and RCM status in an augmented reality environment, indicating significant improvements.

As mentioned earlier, the common issue with most of the aforementioned studies is the missing balance between the

This work was supported by the European Research Council (ERC) starting grant Ergo-Lean, Grant Agreement No. 850932

* : Both authors contributed equally to this work

[1] Human-Robot Interfaces and physical Interaction (HRI^2) lab, Istituto Italiano di Tecnologia, Genoa, Italy, Email: Soheil.Gholami@iit.it, Francesco.Tassi@iit.it

[2] NearLab, Dept. of Electronics, Information and Bioengineering, Politecnico di Milano, Milan, Italy

usability factors such as ergonomics, task performance, and ease-of-use, in the final design of the teleoperation interface.

B. Contributions

In this work, we introduce a new tele-locomanipulation setup, which integrates the features of an ergonomic and a dynamic/maneuverable interface into a unified solution. The former implements a 3D mouse-like function, enabling long-distance locomotion control of the follower robot. The latter permits the operator to move his/her arm freely in space to create dynamic movements or generate precise trajectories during manipulation. Both modes are implemented on a torque controlled robot arm, whose functionalities can be chosen at any time using a button. The design of the ergonomic interface is inspired by [9], in which a 3D mouse interface is used for a comfortable execution of remote loco-manipulation tasks. The second mode implements a zero-torque tracking mode (with gravity compensation) to compensate for the missing capabilities of the ergonomic interface, i.e., the creation of dynamic and precise motion profiles (following a circular path, creating physical impacts, etc.).

The follower robot is Mobile Collaborative Robot Assistant (MOCA), which is composed of an omni-directional mobile base and a torque controlled arm. The arm is controlled using an Hierarchical Quadratic Programming (HQP) based technique [16], to exploit the kinematic redundancy to achieve a set of secondary tasks such as singularity avoidance, joint limits avoidance etc. The proposed interface is validated by five subjects in two different experiments, and the results are compared with respect to an ergonomic interface called *3D mouse* interface.

II. Problem Statement

A. The studied teleoperation user interfaces

The unified evaluation scheme for comparing the proposed *reconfigurable (arm)* interface and the *3D mouse* interface, is illustrated in Fig. 1. Apart from the mentioned interfaces, the MObile Collaborative Robotic Assistant (MOCA), which is an integration of a Robotnik® SUMMIT-XL STEEL mobile platform and a 7-DoF Franka Emika robot manipulator (here, with the default gripper), is used as the follower mobile manipulator, whose robotic arm and mobile platform are controlled separately. Indeed, the arm is controlled by the HQP-based controller and the mobile base with its embedded low-level velocity controller. In what follows, the main elements of this block diagram are discussed in detail.

1) 3D mouse interface (user interface I): this interface enables the users to control a complicated multi-DoF follower robot with a simple and cheap 3D mouse device, namely SpaceMouse® Compact (3Dconnexion, UK). This device has two push buttons and a 6-DoF motion axis sensor. Here, the buttons are programmed for toggling the *control modes* (i.e., locomotion and manipulation) and *motion modes* (i.e., translation and rotation). The mouse's DoF are highly sensitive and coupled, making the generation of precise and decoupled reference trajectories, to be sent to the follower robot, nearly impossible. Hence, first, the *motion modes* are implemented in a way to separate the translational and rotational motions in the manipulation *control mode*. Second, the reference trajectories are generated after processing the raw mouse's motion axis information in the "motion axis processor" block (see Fig. 1).

More precisely, the mouse's raw displacement data are read and encapsulated in a vector $\delta\boldsymbol{x}_r = [\delta\boldsymbol{p}_r^T,\, \delta\boldsymbol{\epsilon}_r^T]^T$ with the frequency of $50.0\,\mathrm{Hz}$. $\delta\boldsymbol{p}_r \in \mathbb{R}^3$ and $\delta\boldsymbol{\epsilon}_r \in \mathbb{R}^3$ are the input translational and rotational displacements along and around the mouse's motion axis, respectively. These are, then, normalized to signed percentage values and assigned to a vector, denoted by $\delta\boldsymbol{x}_r^\dagger \in \mathbb{R}^6$. Afterwards, and if the manipulation *control mode* is chosen, the *motion mode* is checked, assigning $\delta\boldsymbol{\epsilon}_r^\dagger$ or $\delta\boldsymbol{p}_r^\dagger$ to zero when the translation or rotation *motion mode* is activated, respectively. For the locomotion *control mode*, however, the elements of $\delta\boldsymbol{x}_r^\dagger$ are reorganised as $\delta\boldsymbol{x}_r^\dagger = [\delta\boldsymbol{p}_{rx}^\dagger,\, \delta\boldsymbol{p}_{ry}^\dagger,\, \delta\boldsymbol{\epsilon}_{rz}^\dagger,\, \boldsymbol{0}^{1\times 3}]^T$. Then, a moving average filter, with the moving window of size N, i.e., $\boldsymbol{\Phi} \in \mathbb{R}^{6\times N}$ populated with the current $\delta\boldsymbol{x}_r^\dagger$ and its last $N-1$ values, is applied. Using this, the average value of each mouse's motion axis j over the past N samples, $\delta\bar{\boldsymbol{x}}_j$, is calculated by $\delta\bar{\boldsymbol{x}}_j(k) = \frac{1}{N}\sum_{i=k-N-1}^{k} \boldsymbol{\Phi}_{ji}$ where $j \in \{1, \cdots, 6\}$. At last, the maximum value of $\delta\bar{\boldsymbol{x}}_j$ and its corresponding axis $j^\star$ are fetched and the other values are set to zero. Consequently, the desired displacement $\delta\boldsymbol{x}_d = [\delta\boldsymbol{p}_d^T,\, \delta\boldsymbol{\epsilon}_d^T]^T$ is generated, where only the $j^\star$-th element is non-zero.

In the locomotion *control mode*, the scalars $\delta\boldsymbol{p}_{d_x}$, $\delta\boldsymbol{p}_{d_y}$, and $\delta\boldsymbol{\epsilon}_{d_z}$ are mapped to the reference twist vector of the follower mobile platform, denoted by $\boldsymbol{v}_e = [\boldsymbol{v}_{ex},\, \boldsymbol{v}_{ey},\, \boldsymbol{\omega}_{ez}]^T$. $\boldsymbol{v}_{ex}$, $\boldsymbol{v}_{ey}$, and $\boldsymbol{\omega}_{ez}$ are the linear velocities along x and y axes, and the angular velocity around z axis, respectively. This is carried out by the "velocity planner" block (Fig. 1) based on the pre-selected maximum values, i.e., $\boldsymbol{v}_{e_x}^{max}$, $\boldsymbol{v}_{e_y}^{max}$, and $\boldsymbol{\omega}_{e_z}^{max}$. Consequently, the built-in low-level velocity controller indirectly regulates the 2D pose (x-y position and heading angle) of the omni-directional mobile base.

For the manipulation *control mode*, on the other hand, all of the elements of $\delta\boldsymbol{x}_d$ may be used based on the chosen *motion mode*. To be more specific, the operator can adjust the desired end-effector's movements with respect to (w.r.t.) its own frame (at previous time instant $k-1$, denoted by EE_{k-1}). This is done using $\delta\boldsymbol{x}_d$ and the maximum pre-set values for the follower robot's Cartesian displacements, resulting in ${}^{EE_{k-1}}\delta\boldsymbol{x}_{EE_k}^f = \big[{}^{EE_{k-1}}\delta\boldsymbol{p}_{EE_k}^f,\, {}^{EE_{k-1}}\delta\boldsymbol{\epsilon}_{EE_k}^f\big]^T$[1]. Thus, the new pose of the robot's end-effector w.r.t. the world frame $\{W\}$ is obtained by the following homogeneous transformation matrix: ${}^{W}\boldsymbol{T}_{EE_k}^f = {}^{W}\boldsymbol{T}_{EE_{k-1}}^f\, {}^{EE_{k-1}}\boldsymbol{T}_{EE_k}^f$ where

[1] Hereafter, to distinguish the follower's and leader's variables, the f and l superscripts are used, respectively.

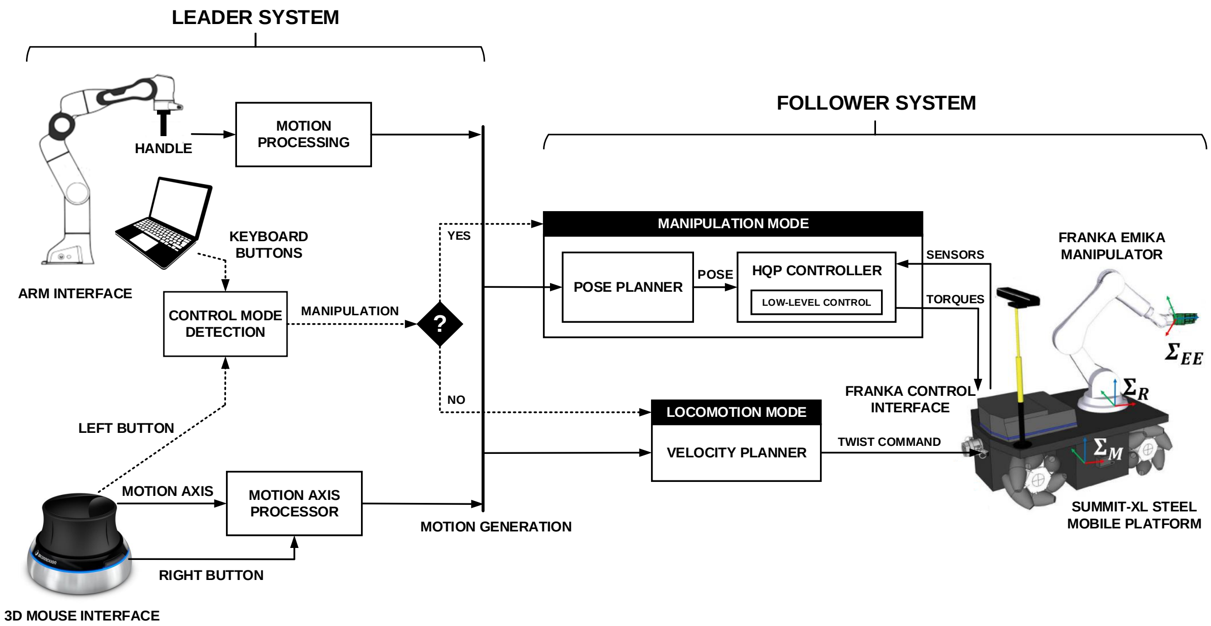


Fig. 1: Block diagram of the studied teleoperation interfaces, for controlling a mobile robotic manipulator.

${}^{EE_{k-1}}\boldsymbol{T}^{f}_{EE_k}$ is updated with the elements of ${}^{EE_{k-1}}\delta\boldsymbol{x}^{f}_{EE_k}$:

$$\begin{aligned} {}^{EE_{k-1}}\boldsymbol{T}^{f}_{EE_k} &= \left[\begin{array}{c|c} {}^{EE_{k-1}}\boldsymbol{R}^{f}_{EE_k} & {}^{EE_{k-1}}\delta\boldsymbol{p}^{f}_{EE_k} \\ \hline \boldsymbol{0}_{1\times 3} & 1 \end{array}\right], \\ {}^{EE_{k-1}}\boldsymbol{R}^{f}_{EE_k} &= R_x\left(\delta\boldsymbol{\epsilon}^{f}_{x}\right) R_y\left(\delta\boldsymbol{\epsilon}^{f}_{y}\right) R_z\left(\delta\boldsymbol{\epsilon}^{f}_{z}\right), \end{aligned} \tag{1}$$

where $\delta\boldsymbol{\epsilon}^{f}_{x}$, $\delta\boldsymbol{\epsilon}^{f}_{y}$, and $\delta\boldsymbol{\epsilon}^{f}_{z}$ are the first, second, and last element of vector ${}^{EE_{k-1}}\delta\boldsymbol{\epsilon}^{f}_{EE_k}$.

2) *The reconfigurable (arm) interface (user interface II):* this utilises a redundant Franka Emika manipulator with 7-DoF to capture the desired user's input motions. For this robot, a simple handle is designed as its end-effector to help the user during the motion generation. These motions are, then, mapped and sent to the follower robot as reference trajectories. Here, the *control modes* are selected by the buttons of a keyboard device. In what follows the separate controllers utilised for each *control mode* are explained.

Locomotion mode: the locomotion control of the follower robot is achieved through the Cartesian impedance control scheme for the leader's robotic arm. This is done by simulating the 3D mouse behaviour around/along an ergonomically selected pivot point of the leader's arm. To implement this behaviour, the following impedance control law is used [17]:

$$\begin{aligned} \boldsymbol{\tau}_c = \boldsymbol{g}(\boldsymbol{q}) &+ \boldsymbol{J}^T\left[\boldsymbol{H}_X(\boldsymbol{x})\ddot{\boldsymbol{x}}_d + \boldsymbol{C}_X(\boldsymbol{x},\dot{\boldsymbol{x}})\dot{\boldsymbol{x}}_d\right] \\ &- \boldsymbol{J}^T\left[\boldsymbol{K}_d\,\Delta\boldsymbol{x} + \boldsymbol{B}_d\,\Delta\dot{\boldsymbol{x}}\right] + \mathcal{N}(\boldsymbol{q})\,\boldsymbol{\tau}_n, \end{aligned} \tag{2}$$

where $\Delta\boldsymbol{x} := \boldsymbol{x} - \boldsymbol{x}_d$. $\boldsymbol{x} \in \mathbb{R}^m$ is the end-effector's pose. $\boldsymbol{x}_d \in \mathbb{R}^m$ is the virtual equilibrium pose. This pose is set before the task based on the users' body structure to help them accomplish the remote locomotion task ergonomically. $\boldsymbol{B}_d \in \mathbb{R}^{m\times m}$ and $\boldsymbol{K}_d \in \mathbb{R}^{m\times m}$ represent the symmetric positive-definite desired damping and stiffness matrices, respectively, to model the external interaction with the user's hand. $\boldsymbol{J} \in \mathbb{R}^{m\times n}$ is the analytical Jacobian matrix. $\boldsymbol{H}_X(\boldsymbol{x}) \in \mathbb{R}^{m\times m}$ and $\boldsymbol{C}_X(\boldsymbol{x},\,\dot{\boldsymbol{x}}) \in \mathbb{R}^{m\times m}$ are the robot's inertia, and the Coriolis and centrifugal matrices in the operational space, respectively, and $\boldsymbol{g}(\boldsymbol{q}) \in \mathbb{R}^n$ is the gravitational torques. $\boldsymbol{\tau}_n \in \mathbb{R}^n$ is the desired nullspace behaviour, and $\mathcal{N}(\boldsymbol{q}) \in \mathbb{R}^{n\times n}$ is the projection matrix which projects $\boldsymbol{\tau}_n$ into the robot's nullspace. Considering the "dynamically consistent nullspace mapping" [17], we have:

$$\begin{aligned} \boldsymbol{\tau}_n &= -\left[\boldsymbol{K}_n\left(\boldsymbol{q} - \boldsymbol{q}_{n,\,d}\right) + \boldsymbol{B}_n\,\dot{\boldsymbol{q}}\right], \\ \mathcal{N}(\boldsymbol{q}) &= \left(\boldsymbol{I}_n - \boldsymbol{J}^T(\boldsymbol{q})\,\boldsymbol{H}_X(\boldsymbol{x})\,\boldsymbol{J}(\boldsymbol{q})\,\boldsymbol{H}^{-1}(\boldsymbol{q})\right). \end{aligned} \tag{3}$$

$\boldsymbol{K}_n \in \mathbb{R}^{n\times n}$ and $\boldsymbol{B}_n \in \mathbb{R}^{n\times n}$ are the symmetric positive-definite desired nullspace stiffness and damping matrices respectively, and $\boldsymbol{q}_{n,\,d} \in \mathbb{R}^n$ is the nullspace virtual equilibrium.

Finally, the translational element of $\Delta\boldsymbol{x}$, i.e., $\Delta\boldsymbol{x}_x$, $\Delta\boldsymbol{x}_y$, and $\Delta\boldsymbol{x}_z$, are mapped to the elements of the reference velocity $\boldsymbol{v}_e$ for the follower mobile base ("velocity planner").

Manipulation mode: the following joint torques, on the other hand, are generated during the manipulation *control mode* in which the user freely moves the leader arm's end-effector (handle) in the Cartesian space: $\boldsymbol{\tau}_c = \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{C}(\boldsymbol{q},\,\dot{\boldsymbol{q}})\,\dot{\boldsymbol{q}}$. While the user exerts force on the handle, the consequent translational displacement vector ${}^{EE_{k-1}}\delta\boldsymbol{p}^{l}_{EE_k} = {}^{W}\boldsymbol{p}^{l}_{EE_k} - {}^{W}\boldsymbol{p}^{l}_{EE_{k-1}}$ and the handle's absolute orientation w.r.t. the world frame $\{W\}$, i.e., ${}^{W}\boldsymbol{R}_{EE_k}$, are sent to the remote follower controller (see Section II-B). So, the desired reference trajectories are retrieved from the homogeneous transformation matrix as:

$${}^{EE_{k-1}}\boldsymbol{T}^{f}_{EE_k} = \left[\begin{array}{c|c} {}^{W}\boldsymbol{R}^{l}_{EE_k} & {}^{W}\boldsymbol{p}^{f}_{EE_k} + {}^{EE_{k-1}}\delta\boldsymbol{p}^{l}_{EE_k} \\ \hline \boldsymbol{0}_{1\times 3} & 1 \end{array}\right]. \tag{4}$$

B. HQP controller

Dealing with redundant robots allows to define a set of secondary tasks that can be executed in the robot nullspace without affecting the performance of the primary ones. This leads to the possibility of generating a stack of tasks that can be executed hierarchically, exploiting the whole redundancy of the robot. Indeed, by considering $k \in \{1, \dots p\}$ levels of priority where the importance decreases with k down to the last task p, we ensure that the solutions found at level k are always strictly enforced at lower priority levels, which constitutes the main reason why we choose a *strict priority* scheme [18]. Another common approach is instead to adopt *soft priorities*, a simple solution that weights each task as in [19] and that does not require the solution of nested QP problems. In addition to some of our latest work [20], here a *strict priority* scheme is implemented together with weighting strategies to define the importance between tasks at the same hierarchical level, so that all the solutions are influenced by each other proportionally to their weights. Using the same structure of [20], the k^{th} hierarchical problem is defined as:

$$\begin{aligned} &\min_{\boldsymbol{\chi}} \frac{1}{2} ||\boldsymbol{A_k\chi} - \boldsymbol{b_k}||^2 \\ s.t.\, &\boldsymbol{C_1\chi} \le \boldsymbol{d_1}, \ \dots, \ \boldsymbol{C_k\chi} \le \boldsymbol{d_k} \\ &\boldsymbol{E_1\chi} = \boldsymbol{f_1}, \ \dots, \ \boldsymbol{E_k\chi} = \boldsymbol{f_k} \end{aligned} \tag{5}$$

with generic matrices $\boldsymbol{A}_k, \boldsymbol{C}_k, \boldsymbol{E}_k \in \mathbb{R}^{n\times n}$, and vectors $\boldsymbol{b}_k, \boldsymbol{d}_k, \boldsymbol{f}_k \in \mathbb{R}^n$, while $\boldsymbol{\chi} \in \mathbb{R}^n$ is the generic variable to optimise, subject to equality and inequality constraints. The previous $1, \dots, k-1$ solutions are considered with the optimality condition between successive tasks $\boldsymbol{A_{k-1}\chi} = \boldsymbol{A_{k-1}\chi^*_{k-1}}$, whose demonstration is reported in [18]. In this way, the optimality of the tasks with higher priority is not altered by the actual solution, and it can be added in (5) as a set of equality constraints by considering $\boldsymbol{E_1} = \boldsymbol{0}, \boldsymbol{f_1} = \boldsymbol{0}$, up to $\boldsymbol{E_k} = \boldsymbol{A_{k-1}}, \boldsymbol{f_k} = \boldsymbol{A_{k-1}}\,\boldsymbol{\chi^*_{k-1}}$.

Considering now the Inverse Kinematics (IK) problem of a redundant robot: $\dot{\boldsymbol{q}} = \boldsymbol{J}^\dagger \dot{\boldsymbol{x}}$, this represents one of the possible solutions to the problem:

$$\min_{\dot{\boldsymbol{q}}} \|\boldsymbol{J}\dot{\boldsymbol{q}} - \dot{\boldsymbol{x}}\|^2. \tag{6}$$

Being written in least-squares form, it is possible to exploit HQP based techniques to solve the robot kinematics problem, even in fast and real-time control applications. In addition, a Closed-loop IK (CLIK) scheme is used to recover from position errors between the desired and actual behaviour, rewriting equation (6) as:

$$\min_{\dot{\boldsymbol{q}}} \|\boldsymbol{J}\dot{\boldsymbol{q}} - (\dot{\boldsymbol{x}}_d + \boldsymbol{K_p}(\boldsymbol{x}_d - \boldsymbol{x}_a))\|^2, \tag{7}$$

where $\boldsymbol{x}_a \in \mathbb{R}^m$ is the actual Cartesian position and $\boldsymbol{K_p} \in \mathbb{R}^{m\times m}$ is the positive-definite diagonal gain matrix that is responsible for the error convergence.

C. Stack of tasks definition

It is now possible to identify a set of objective functions that define the hierarchical stack of tasks. In [21], the authors present an exhaustive list of indices that can be useful in evaluating the performance of robotic manipulators. Based on this work, some of these indices are here formulated in QP form as the possible objective functions of the HQP control scheme, as it will be explained in this section.

1) Joint Range Availability: as mentioned in [21], the Joint Range Availability index (JRA) follows the joint angles' deviation from their mid-range values. This is useful to determine whether a joint reaches a stop point and to study the status of joint range distribution. Thereby, the normalised JRA is here expressed in QP form as:

$$\min_{\dot{\boldsymbol{q}}} \left\| \frac{(\boldsymbol{q} - \bar{\boldsymbol{q}})}{\boldsymbol{q}^{max}} \right\|^2. \tag{8}$$

where $\bar{\boldsymbol{q}}$, and $\boldsymbol{q}^{max}$ are the joint mid-range and joint limit positions, respectively. Alternatively, the same definition can be used to express a postural task, which exploits the redundancy of the robot to keep the joint space configuration as close as possible to a fixed target one. This can be achieved by simply defining $\bar{\boldsymbol{q}} = \boldsymbol{q}_{target}$ as the target robot posture.

2) Joint Velocity Index: similar to JRA, the joint velocity index can be formulated by making use of the joints' current velocities and their maximum values to track the joints' velocities distribution and status in time. This is defined as:

$$\Psi^\star := \sum_{i=1}^{n} \frac{\dot{\boldsymbol{q}}_i^{\,2}}{\dot{\boldsymbol{q}}_i^{max}}, \tag{9}$$

which reflects the regularization term

$$\min_{\dot{\boldsymbol{q}}} ||\dot{\boldsymbol{q}}||^2, \tag{10}$$

useful for stability as discussed in [18], and that is then weighted with other objectives at the same hierarchical level.

3) Manipulability Index: the vastly used Manipulability Index (MI) is based on the Jacobian matrix $\boldsymbol{J(q)}$, and in spite of its limitations [21], is defined as follows:

$$m(\boldsymbol{q}) := \sqrt{|\boldsymbol{J}(\boldsymbol{q})\,\boldsymbol{J}(\boldsymbol{q})^T|}, \text{ or, } m(\boldsymbol{q}) := \sigma_1\,\sigma_2 \,\dots\, \sigma_m. \tag{11}$$

σ_i are the singular values of $\boldsymbol{J}$ such that $\sigma_1 \ge \sigma_2 \dots \ge \sigma_m$.

This index is important for improving the quality of the task and the performances of the robot since it is proportional to the volume of the manipulability ellipsoid, defined by the singular values of $\boldsymbol{J}$. Thereby, it is possible to maximize the volume of the ellipsoid by formulating the problem:

$$\min_{\dot{\boldsymbol{q}}} \ -m(\boldsymbol{q}), \tag{12}$$

which becomes a nonlinear optimisation problem, hard to be solved in real-time applications. Following [22], it is instead possible to formulate this problem in QP form, by linearizing the generic nonlinear function as follows:

$$m(\boldsymbol{q}) = m(\boldsymbol{q}_{t-\Delta t}) + \Delta t\,(\nabla m)^T \dot{\boldsymbol{q}} + \frac{1}{2}\,\Delta t^2\,\dot{\boldsymbol{q}}^T\,\boldsymbol{H}_m\,\dot{\boldsymbol{q}}, \tag{13}$$

where ∇m and $\boldsymbol{H}_m$ are the gradient vector and Hessian matrix of m, respectively, while Δt is the control loop sampling time. Thus, the maximization problem can be written in the standard form as:

$$\min_{\dot{\boldsymbol{q}}} - \left(\frac{1}{2}\Delta t^2 \dot{\boldsymbol{q}}^T H_m \dot{\boldsymbol{q}} + \Delta t(\nabla m)^T \dot{\boldsymbol{q}}\right). \tag{14}$$

The authors in [22] also demonstrate how to obtain an equivalent optimisation problem to account for nonlinear functions:

$$\min_{\dot{\boldsymbol{q}}} \frac{1}{2}\, \Delta t^2\, \dot{\boldsymbol{q}}^T \nabla m\, (\nabla m)^T \dot{\boldsymbol{q}} - m\, \Delta t\, (\nabla m)^T \dot{\boldsymbol{q}}, \tag{15}$$

which allows to avoid the increased computational burden due to the calculation of the Hessian matrix H_m, and thus complying with real-time requirements.

4) Dynamic Manipulability: To consider the manipulator's dynamics in its manipulating capabilities, an extension of the aforementioned index m is reported. This quantifies the manipulator's capacity to produce the acceleration in response to the joints' driving forces/torques [21]. Considering H as the manipulator's inertia matrix, the Dynamic Manipulability m_d is obtained with the following equation:

$$m_d(\boldsymbol{q}) := \sqrt{|\boldsymbol{J}(\boldsymbol{q})\ \left(\boldsymbol{H}(\boldsymbol{q})\, \boldsymbol{H}(\boldsymbol{q})^T\right)^{-1}\ \boldsymbol{J}(\boldsymbol{q})^T|}. \tag{16}$$

It is clearly possible to include this index in the stack of tasks similarly as it was done for $m(\boldsymbol{q})$ (11).

D. Constraints definition

Task feasibility is limited in the real scenario of a robot working in an environment with obstacles and interaction bounds, in addition to the already existing technological limits inherent to the machine. *Kinematic constraints* are generally represented by the actuator range of motion, velocity and acceleration constraints, which can be written as:

$$\begin{aligned} \boldsymbol{q}_{min} \leq \boldsymbol{q}(t - \Delta t) + \dot{\boldsymbol{q}}(t)\Delta t \leq \boldsymbol{q}_{max}, \\ \dot{\boldsymbol{q}}_{min} \leq \dot{\boldsymbol{q}} \leq \dot{\boldsymbol{q}}_{max}, \\ \ddot{\boldsymbol{q}}_{min} \leq \frac{\dot{\boldsymbol{q}}(t) - \dot{\boldsymbol{q}}(t - \Delta t)}{\Delta t} \leq \ddot{\boldsymbol{q}}_{max}. \end{aligned} \tag{17}$$

E. Overall control structure

In conclusion, in the overall optimisation problem that will be used in the experiments (Section III), the first hierarchical level is always occupied by the CLIK defined in (7), which is followed by both the JRA and manipulability tasks described in (8) and (15), respectively. These two objectives are considered together in a *soft hierarchy* fashion, using relative weights, as it will be further discussed in the experiments (Section III-B). The last task of the stack in *strict* sense is the regularization term (10). The constraints are instead defined as in (17). Finally, the obtained optimal joint velocities $\dot{\boldsymbol{q}}^*$ are passed to a low-level decoupled joint impedance controller.

III. EXPERIMENTS

To investigate the cons and pros of each studied interface (Section II-A), in terms of performance requirements, two sets of subjective experiments were conducted using five healthy subjects (four male and one female subjects). These experimental sets were: (i) target selection on a moving surface and (ii) path tracking. Each participant was asked to perform the experiments in two independent trials, i.e., with the *mouse* interface and the *reconfigurable* interface. In the next sections, the experimental setups and results are reported.

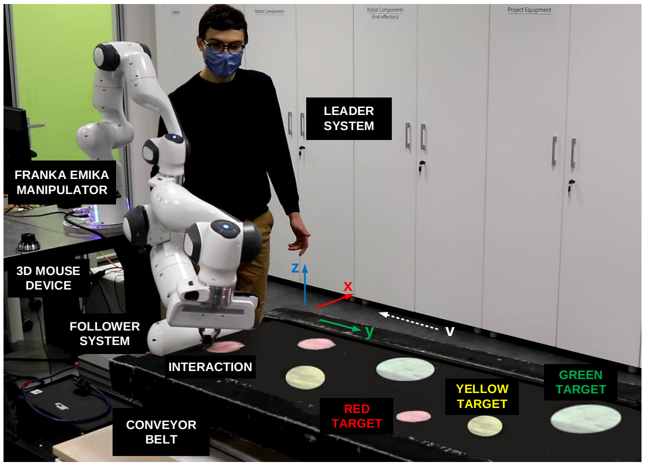


Fig. 2: The setup of experiment 1: the user selects the moving targets one after another, using the arm interface. The colorful targets are marked on the surface of a conveyor belt which is running at constant speed ($v = 0.28\,\mathrm{m\,s^{-1}}$).

A. Target selection on a moving surface

The objective of the first experimental set-up (see Fig. 2) is to evaluate, both qualitatively and quantitatively, the user performance and acceptability during the use of each aforementioned teleoperation interface (2 trials) in highly dynamic tasks. To do so, the subjects were asked to carry out a dynamic target selection task. The colorful targets, highlighted on the surface of a conveyor belt, have different sizes, and move with a constant velocity in the direction specified in Fig. 2 (dotted white arrow v). aWe used three colors in a fixed pattern to indicate the difficulty level. In particular, each green target, i.e., the biggest filled circle, has 1 point, the yellow one has 2 points, and the red one has 3 points (the smallest). In a game-like experience, we asked the participants to hit the targets as many as they can in a given one-minute time window, taking the following rules into account: (i) selecting the targets by moving the end-effector along the x and y axes and (i) not hitting the unmarked area. Neglecting these rules counts as 1 negative point in the scoring system. In addition, after each trial (each of the used interfaces), we asked the participants to fill a NASA-TLX questionnaire form to evaluate the effectiveness and performance of each leader system, which in turn rate the user's perceived workload during the task. This assesses the workload based on mental demand (MD), physical demand (PD), temporal demand (TD), performance (PE), effort (EF), and frustration (FR) scales. Each scale is scored from 1 to 21, indicating the strength with low, medium, and high labels. An extra question is also added to the standard form examining the naturality (NT) of the motion-mapping system: *"how natural was the interface based on your desired input motions?"*

The results of the qualitative NASA-TLX subjective analyses are shown in Fig. 3. The participants appear to feel more mental and temporal demands with the *3D mouse* user interface (MD and TD, respectively). However, the "physical demanding" scale is quite similar for both systems with a

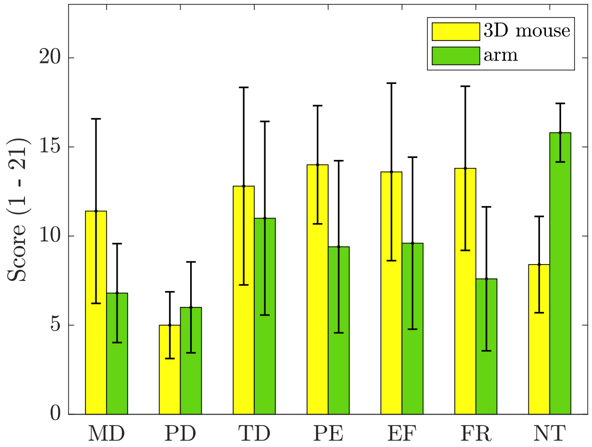


Fig. 3: NASA-TLX (plus naturality scale (NT)) results for the first experiment, i.e., comparing the interfaces in doing the target selection task on a moving surface.

slight higher score for the *arm* user interface. Regarding the "performance" scale, and it was expected, the subjects report a better state for the user interface which utilises a robotic manipulator as its leader interface (for this particular measure, the scoring scale is reversed: 1.0 and 21.0 indicate "perfect" and "failure", respectively). This is due to the higher similarity between the generated motions of the *arm* interface and the ones followed by the robot, which might help the users feel better about their commands and performances when this user interface was used. This is also noticed in the last scale "naturality" for which the *arm* interface has a better score. Moreover, the perceived "effort" and "frustration" scales are higher in the trials with the *mouse* interface, which is, again, because of the motion retargeting behaviour of the 3D mouse and its difference with the follower robot.

With regard to the quantitative subjective analyses, the results are summarised in Table I. In this table, the number of hits for each target and the number of wrong decisions (i.e., neglecting the specified rules) are presented. The mean values μ of the final "score" indicate a higher score (points) for the *arm* user interface with the ratio of 2.21 with respect to the other one. One interesting observation here is that the statistical values related to the "wrong decisions" factor are quite the same in both interfaces, i.e., 11.60 ± 2.30 and 10.00 ± 2.54, respectively. One immediate conclusion is that the human supervisory decisions and the perception feedback perform similarly in both trials. However, the number of correct hits is larger using the *reconfigurable (arm)* interface which indicates faster end-effector motions (task execution) in this trial.

TABLE I: The subjective analyses of the first experiment. Here, *pts* stands for points, and μ and σ represent the mean and standard deviation, respectively.

Subject	Interface	Red (3 pts)	Yellow (2 pts)	Green (1 pt)	Wrong (-1 pt)	Score (pts)
S1	3D mouse	3	3	8	9	14
	arm	7	3	5	10	22
S2	3D mouse	1	2	7	10	4
	arm	2	3	10	6	16
S3	3D mouse	5	4	5	12	16
	arm	9	3	5	11	27
S4	3D mouse	4	2	4	15	5
	arm	5	5	7	13	19
S5	3D mouse	5	2	6	12	13
	arm	6	7	9	10	31
$\mu \pm \sigma$	3D mouse	3.60 ± 1.67	2.60 ± 0.89	6.00 ± 1.58	11.60 ± 2.30	10.4 ± 5.50
	arm	5.80 ± 2.58	4.20 ± 1.78	7.20 ± 2.28	10.00 ± 2.54	23.0 ± 6.04

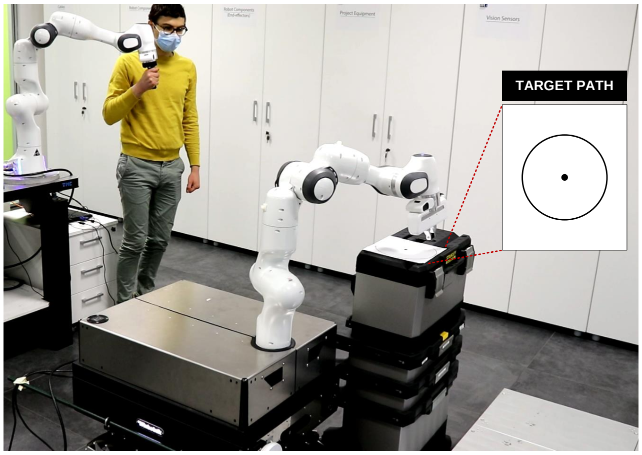


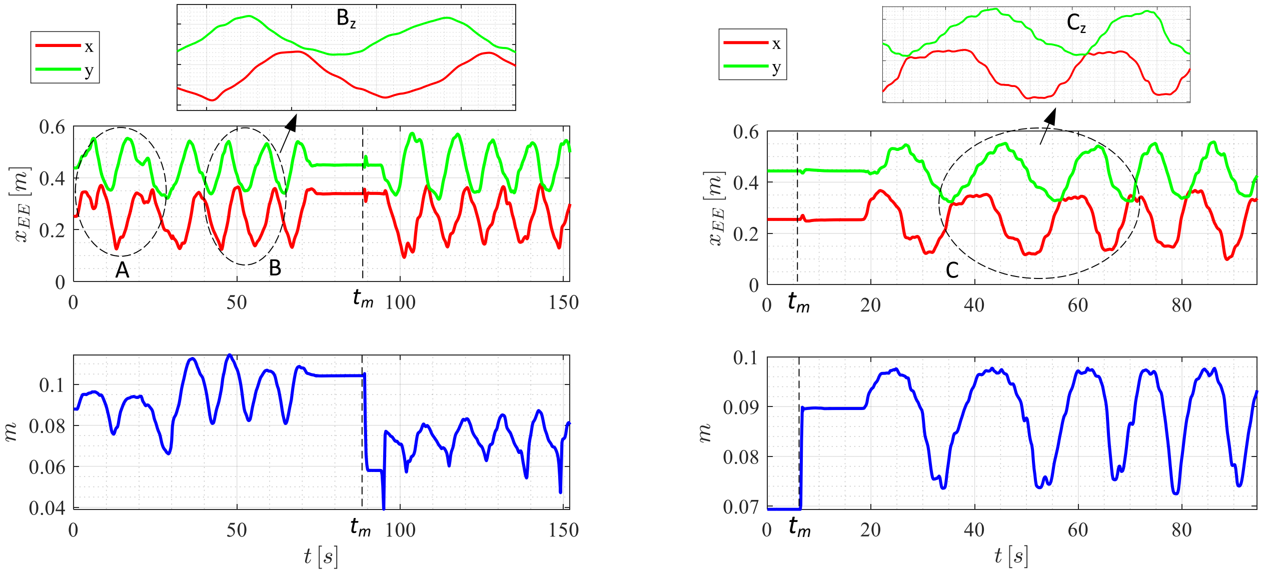
Fig. 4: The setup of experiment 2: after approaching the target location in *locomotion mode*, the user manipulates the remote robot to follow the specified target path.

TABLE II: The quantitative experimental results for the second subjective experiments. Here, the results are shown as a "$\mu \pm \sigma$" pair among all the participants. Also, path tracking error is defined as $\tilde{r} = r_{EE} - r_d$, being r_{EE} the end-effector's displacement w.r.t. the desired path center and r_d is the path's radius.

Metric name	Metric Definition	Arm without $m(\boldsymbol{q})$	Arm with $m(\boldsymbol{q})$	3D Mouse with $m(\boldsymbol{q})$
Path Tracking Error (RMS) [m]	$e_{RMS} = \sqrt{\frac{1}{T}\int_0^T \tilde{r}^2\, dt}$	0.022 ± 0.008	0.029 ± 0.006	0.047 ± 0.006
Path Tracking Error (average) [m]	$\mu_e = \frac{1}{T}\int_0^T \tilde{r}\, dt$	0.014 ± 0.006	0.015 ± 0.004	0.036 ± 0.028
Path Tracking Error (standard deviation) [m]	$\sigma_e = \sqrt{\frac{1}{T}\int_0^T (\tilde{r} - \mu_e)^2\, dt}$	0.013 ± 0.009	0.016 ± 0.008	0.041 ± 0.021
Manipulability Index (average)	$\mu_m = \frac{1}{T}\int_0^T m\, dt$	0.07 ± 0.018	0.09 ± 0.006	0.08 ± 0.007
Task Execution Time [s]	$T_{task} = t_{final} - t_{initial}$	63.1 ± 5.6	61.3 ± 6.9	94.5 ± 7.0

B. Path tracking

The aim of the second experiment is to show the different accuracy between the two teleoperation interfaces, and to eventually confirm the results already obtained in the previous experiment. The goal is to first approach the MOCA to the target location from a remote position, using the *locomotion mode* of the interface related to the current trial. Once the target area is approached (Fig. 4), the user switches to the *manipulation mode* and tries to draw along the target path for 5 consecutive turns. The user was asked to perform this operation as accurately and as fast as possible, in order to properly evaluate each interface both in terms of accuracy and motion dynamics. Analogously to the first experiment, five healthy subjects were asked to carry out this task, and the results are analyzed using the metrics reported in Table II, which are reported as mean μ and standard deviation σ values considered among all the subjects. Fig. 5 shows the comparison between the experiments conducted using the *reconfigurable arm* (Fig. 5a) and the *3D mouse* (Fig. 5b) interfaces, respectively, for



(a) Experiment 2 while using the *reconfigurable (arm)* interface. The first 5 turns are performed with the manipulability maximization task activated. After t_m, the manipulability index is deactivated and the same experiment is conducted for comparison.

(b) Experiment 2 while using the *3D mouse* interface, and considering only the case in which the manipulability index is active (the task is activated at t_m).

Fig. 5: Comparison between the studied user interfaces in *manipulation mode* while performing 5 turns around the target path for a random subject: (a) *reconfigurable (arm)* interface and (b) *3D mouse* interface. The top plots show the actual trajectories followed by the end-effector $\boldsymbol{x}_{EE}$, while the bottom plots depict the behaviour of the manipulability index m. Specific regions B and C are considered together with their magnified versions (B_z and C_Z) to compare the trajectories' smoothness when using the *arm* and *3D mouse* interfaces, respectively. In addition, region A reflects the learning curve period for the user to get acquainted to the employed interface, after which fewer fluctuations are observed in the trajectories.

an individual random subject. The top plots show the actual end-effector trajectories of the follower robot $\boldsymbol{x}_{EE}$ along the x and y directions, while the bottom plots show the behaviour of the manipulability index $m(\boldsymbol{q})$. In particular, in order to study the possible effects and improvements in adopting an HQP control strategy, each subject was asked to perform 5 turns along the specified target path for two times with each interface. Indeed, as visible from Fig. 5a, the first set of 5 turns is conducted with the maximum manipulability objective defined in (15) already activated. At t_m instead, the manipulability task is deactivated, leaving only the JRA task (8) as secondary priority. This is visible from the reconfiguration of the kinematic chain that occurs after t_m leading to a drop of $m(\boldsymbol{q})$, thus approaching an arm configuration that is closer to singularity. The values of Table II related to the averaged manipulability index μ_m reflect the aforementioned improvement throughout all the subjects, with an increase from $\mu_m = 0.072 \pm 0.018$ to $\mu_m = 0.093 \pm 0.006$ when considering the *arm interface* without and with $m(q)$ activation, respectively. This is a remarkable improvement, noticing that the manipulability index is robot-dependant and not bounded, and thus to provide a normalized value a mapping in the entire workspace is necessary [21]. For this reason we decided to only provide the relative values for each configuration, despite the small range between $m(\boldsymbol{q})_{min}$ and $m(\boldsymbol{q})_{max}$. In addition, when repeating the same experiments but with the target path in an harder-to-reach location, and performing repetitive tasks without the activation of the manipulability index, it is common that the arm approaches near-singular configurations, physically hindering the successful completion of the experiment. Thereby, choosing an hierarchical scheme as the one proposed allows to improve the overall task performances both quantitatively in terms of accuracy and qualitatively in terms of successful task completion and user's satisfaction.

When considering the *3D mouse* interface as well, the same experiment is assigned to all the subjects. We report in Fig. 5b only the experiment with active manipulability index, for the sake of brevity. The path following task starts after t_m, when the manipulability task is activated and the robot arm reconfigures in optimal sense. It is now visible from both $\boldsymbol{x}_{EE}$ and $m(\boldsymbol{q})$ trajectories that the overall end-effector trajectories are *less smooth* and *more discontinuous* (see magnified region C_z in Fig. 5b) with respect to the previous interface, i.e., the *arm* interface (see B_z in Fig. 5a). This indicates a lower accuracy, also confirmed from the averaged path tracking errors among all the subjects in Table II, which is more than doubled with respect to the *arm* interface, going from $\mu_e = 0.015 \pm 0.004\,\mathrm{m}$ to $\mu_e = 0.036 \pm 0.028\,\mathrm{m}$. The same goes for the standard deviation which increases from $\sigma_e = 0.016 \pm 0.008\,\mathrm{m}$ to $\sigma_e = 0.041 \pm 0.021\,\mathrm{m}$, indicating much less accuracy and consistency throughout the tracking. As expected, also the task execution time, shown in Table II, undergoes a substantial increase of approximately 30 s on average.

IV. CONCLUSION

In this work, we introduced a reconfigurable user interface for remote loco-manipulation of mobile collaborative robots. The proposed interface merged the advantages of an ergonomic teleoperation system for prolonged locomotion tasks, and a maneuverable one for more precise and/or dynamic control of manipulation, into a unified solution. Through a button, the users could easily reconfigure between these two functionalities to implement a desired comfort-performance trade-off. The experiments with quantitative and qualitative analyses demonstrated the effectiveness of the proposed interface in two real-world subjective experiments.

REFERENCES

[1] L. Chan, F. Naghdy, and D. Stirling, "Position and force tracking for non-linear haptic telemanipulator under varying delays with an improved extended active observer," *Robotics and Autonomous Systems*, vol. 75, pp. 145–160, 2016.

[2] 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.

[3] M. Laghi, M. Maimeri, M. Marchand, C. Leparoux, M. Catalano, A. Ajoudani, and A. Bicchi, "Shared-autonomy control for intuitive bimanual tele-manipulation," in *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2018, pp. 1–9.

[4] A. Ajoudani, N. Tsagarakis, and A. Bicchi, "Tele-impedance: Teleoperation with impedance regulation using a body–machine interface," *The International Journal of Robotics Research*, vol. 31, no. 13, pp. 1642–1656, 2012.

[5] F. Hu, Y. Deng, H. Zhou, T. H. Jung, C.-B. Chae, and A. H. Aghvami, "A vision of an xr-aided teleoperation system toward 5g/b5g," *IEEE Communications Magazine*, vol. 59, no. 1, pp. 34–40, 2021.

[6] I. Havoutis and S. Calinon, "Learning from demonstration for semi-autonomous teleoperation," *Autonomous Robots*, vol. 43, no. 3, pp. 713–726, 2019.

[7] C. Hatzfeld, C. Neupert, S. Matich, M. Braun, J. Bilz, J. Johannink, J. Miller, P. P. Pott, H. F. Schlaak, M. Kupnik *et al.*, "A teleoperated platform for transanal single-port surgery: Ergonomics and workspace aspects," in *2017 IEEE World Haptics Conference (WHC)*. IEEE, 2017, pp. 1–6.

[8] L. Peternel, C. Fang, M. Laghi, A. Bicchi, N. Tsagarakis, and A. Ajoudani, "Human arm posture optimisation in bilateral teleoperation through interface reconfiguration," in *2020 8th IEEE RAS/EMBS International Conference for Biomedical Robotics and Biomechatronics (BioRob)*. IEEE, 2020, pp. 1102–1108.

[9] S. Gholami, M. Lorenzini, E. De Momi, and A. Ajoudani, "Quantitative physical ergonomics assessment of teleoperation interfaces," *arXiv preprint arXiv:2105.09809*, 2021.

[10] L. Penco, B. Clément, V. Modugno, E. M. Hoffman, G. Nava, D. Pucci, N. G. Tsagarakis, J.-B. Mouret, and S. Ivaldi, "Robust real-time whole-body motion retargeting from human to humanoid," in *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2018, pp. 425–432.

[11] T.-C. Lin, A. U. Krishnan, and Z. Li, "Shared autonomous interface for reducing physical effort in robot teleoperation via human motion mapping," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 9157–9163.

[12] T. Lin, A. U. Krishnan, and Z. Li, "Physical fatigue analysis of assistive robot teleoperation via whole-body motion mapping," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 2240–2245.

[13] J. Luo, C. Yang, N. Wang, and M. Wang, "Enhanced teleoperation performance using hybrid control and virtual fixture," *International Journal of Systems Science*, vol. 50, no. 3, pp. 451–462, 2019.

[14] A. Torabi, M. Khadem, K. Zareinia, G. R. Sutherland, and M. Tavakoli, "Manipulability of teleoperated surgical robots with application in design of master/slave manipulators," in *2018 international symposium on medical robotics (ISMR)*. IEEE, 2018, pp. 1–6.

[15] H. Su, S. Li, J. Manivannan, L. Bascetta, G. Ferrigno, and E. De Momi, "Manipulability optimization control of a serial redundant robot for robot-assisted minimally invasive surgery," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 1323–1328.

[16] A. Escande, N. Mansard, and P.-B. Wieber, "Hierarchical quadratic programming: Fast online humanoid-robot motion generation," *The International Journal of Robotics Research*, vol. 33, no. 7, pp. 1006–1028, 2014.

[17] 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.

[18] O. Kanoun, F. Lamiraux, and P.-B. Wieber, "Kinematic control of redundant manipulators: Generalizing the task-priority framework to inequality task," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 785–792, 2011.

[19] J. Salini, V. Padois, and P. Bidaud, "Synthesis of complex humanoid whole-body behavior: A focus on sequencing and tasks transitions," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 1283–1290.

[20] F. Tassi, E. De Momi, and A. Ajoudani, "Augmented hierarchical quadratic programming for adaptive compliance control," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021.

[21] S. Patel and T. Sobh, "Manipulator performance measures-a comprehensive literature survey," *Journal of Intelligent & Robotic Systems*, vol. 77, no. 3, pp. 547–570, 2015.

[22] K. Dufour and W. Suleiman, "On maximizing manipulability index while solving a kinematics task," *Journal of Intelligent & Robotic Systems*, vol. 100, no. 1, pp. 3–13, 2020.