Manipulability-Aware Shared Locomanipulation Motion Generation for Teleoperation of Mobile Manipulators

Davide Torielli[1,2], Luca Muratore[1], and Nikos Tsagarakis[1]

***Abstract*— The teleoperation of mobile manipulators may pose significant challenges, demanding complex interfaces and causing a substantial burden to the human operator due to the need to switch continuously from the manipulation of the arm to the control of the mobile platform. Hence, several works have considered to exploit shared control techniques to overcome this issue and, in general, to facilitate the task execution.**

This work proposes a manipulability-aware shared locomanipulation motion generation method to facilitate the execution of telemanipulation tasks with mobile manipulators. The method uses the manipulability level of the end-effector to control the generation of the mobile base and manipulator motions, facilitating their simultaneous control by the operator while executing telemanipulation tasks. Therefore, the operator can exclusively control the end-effector, while the underlying architecture generates the mobile platform commands depending on the end-effector manipulability level.

The effectiveness of this approach is demonstrated with a number of experiments in which the CENTAURO robot, a hybrid leg-wheel platform with an anthropomorphic upper body, is teleoperated to execute a set of telemanipulation tasks.

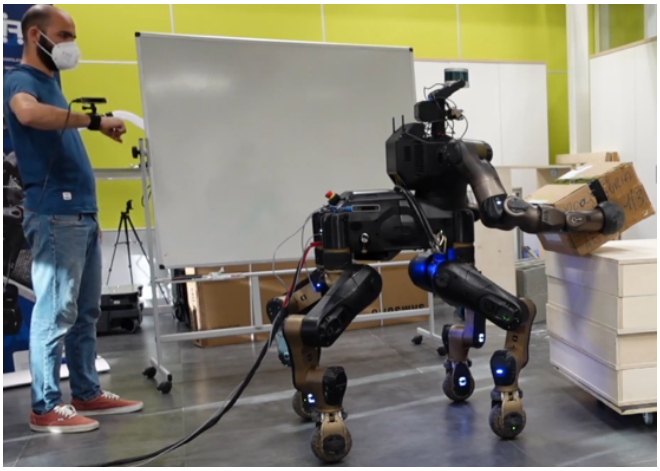

Fig. 1. The CENTAURO robot is being teleoperated to transport a box. The experiment is described in Section IV.

I. Introduction

The teleoperation of robotic mobile manipulators has shown great advancements over the past years. This has permitted to extend the use of mobile robots in different applications, including disaster response [1], clinical therapies [2], inspection and maintenance [3], industrial human-robot collaboration [4], and construction industry [5]. As the complexity of mobile/legged manipulator platforms has increased, improving their capabilities, additional challenges for the teleoperation of such kind of systems have been presented. These new challenges can be tackled by more intelligent human-robot teleoperation interfaces, with different levels of *shared control* or *shared autonomy* [6]. For example, the robot may autonomously point the gripper in a specific direction [7], or may override the commands of the human operator to avoid unsafe regions [8] and obstacles [9]. A certain level of local robot autonomy can also facilitate the accomplishment of the task itself, like to maintain the grasp on a object transported by a dual-arm system using a teleoperation interface [10], [11]. Similarly, the generation of motions for the mobile base and the on-board manipulation system can be assisted through autonomous modules that can permit to exploit the mobile base motions to augment the manipulation actions of the arm. For example, the mobile base may contribute to the end-effector reaching a target location with a good arm dexterity for eventually executing more effectively the requested manipulation task. The objective of maintaining a good dexterity level, related to the avoidance of kinematic singularities, is often considered as an objective among multiple ones. Often, whole-body controllers address this problem with the task priority strategy [12]–[14], which allows to operate the robot while considering other objectives, such as joint limits avoidance [15] and singularities avoidance [16]. In these solutions, the dexterity of the manipulator can be considered among the stack of tasks, but it does not directly influence the generation of the mobile base motions. In this paper, we explore the *manipulability* measure of the robot arm end-effector to make the mobile base actively coordinate its motions with the manipulator motions in order to assist the end-effector reaching the manipulation target location with a good level of dexterity.

There are different definitions of manipulability [17]; one of the most common describes it as a measure of the transmission ratio from the joint velocity to the end-effector velocity [18], [19]. This measure was investigated in many works and extended for the case of dual arm [20], closed-

*This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 101016007 CONCERT and grant agreement No. 871237 SOPHIA, and the Italian Fondo per la Crescita Sostenibile - Sportello "Fabbrica intelligente", PON I&C 2014 - 2020, project number F/190042/01-03/X44 RELAX.

[1]Davide Torielli, Luca Muratore, and Nikos Tsagarakis are with Humanoids and Human Centered Mechatronics (HHCM), Istituto Italiano di Tecnologia, Genova, Italy davide.torielli@iit.it, luca.muratore@iit.it, nikos.tsagarakis@iit.it

[2]Davide Torielli is also with Department of Informatics, Bioengineering, Robotics, and Systems Engineering (DIBRIS), University of Genova, Genova, Italy

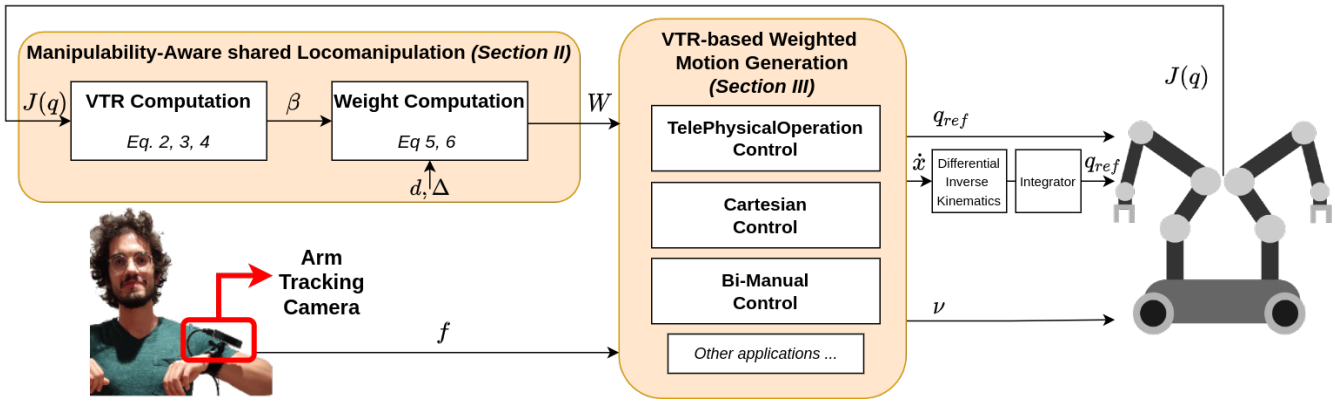

Fig. 2. Logic scheme of the proposed approach. To the left side, a weight $\boldsymbol{W}$ is computed from a manipulability measure, as explained in Section II. This weight can be exploited by different control laws, which, from a teleoperation interface command $\boldsymbol{f}$, can generate motions for the arms and the mobile base (Section III).

loop chain [21], and mobile manipulators. Within the last category, the exploitation of manipulability to generate the platform velocities was first studied in [22]. In [23], manipulability is used to derive the placement of a robot arm on the mobile base. The work in [24] proposes a new definition of manipulability for the mobile manipulator as a whole. Nevertheless, it may happen that the robot arm has anyway a low manipulability, even if the manipulability of the entire mobile manipulator is high. To cope with this problem, a weight is used to select the fixed arm manipulability with respect to the mobile arm one (or vice versa) depending on the manipulation task executed. A similar approach is followed in [25], where instead the product of the two different manipulability measures (fixed arm and mobile arm) is used. The work in [26] proposes a formulation based on quadratic programming with joint limits as a constraint and the manipulability as an objective to maximize. Finally, in [27] the problem is extended to a dual arm mobile platform, with the addition of a *virtual kinematic chain* to specify the common motion of the two arms.

In this work, we propose a new method that can assist in the teleoperation of mobile manipulation platforms, permitting the human operator to execute telemanipulation tasks while keeping the manipulability measure of the remote end-effector at a certain level. The method explores the mobile base motions that are generated in a shared control fashion based on the manipulability measure of the robot arm end-effectors. Therefore, the main idea is to use the manipulability measure as a way to *smoothly* scale down arm velocities in favour of mobile base velocities, when the end-effector enters in a region of low manipulability while teleoperated by the human operator. In this way, the proposed method can facilitate the task execution and reduce the effort of the operator in taking care of this aspect, e.g., by manually commanding the mobile base motions for this purpose. The method explores the manipulability ellipsoid in the three Cartesian directions, permitting to set different thresholds under which the mobile base velocities are generated. This provides flexibility in choosing the direction in which we want the manipulator arm to be more dexterous and accordingly regulate the mobile base motions in this specific direction. For simplicity, we neglect rotational motions of the end-effector throughout the paper. Hence, the manipulability measure is referred to the linear motions of the end-effector, and the generated scaled arm velocities and mobile base velocities are always considered linear velocities.

We have experimentally demonstrated our approach with a number of tasks performed by the CENTAURO robot [28], a hybrid leg-wheel system with an anthropomorphic upper body (Fig. 1). An overview schematic of the proposed method where the relevant computations are presented is illustrated in Fig. 2.

The paper is structured as follows: in Section II we present our manipulability-aware motion generation; in Section III we describe how the weighted motion generation is exploited; in Section IV we validate our approach with a number of experiments; in Section V conclusions are drawn.

II. Manipulability-Aware Shared Locomanipulation

In our previous work we have introduced the TelePhysicalOperation (TPO) concept [29] to control the teleoperated robot using a "marionette" type of interface. By selecting specific control points on the robot kinematic chains, we can apply virtual forces that in a remote manner resembles the human robot interaction way of guiding a robot during a teaching/collaborative task. The operator motions are tracked by a specific motion capture system composed of tracking cameras. The proposed method permits to command the robot from a distance by exploring the intuitiveness of the "marionette" based physical interaction with the robot in a virtual/remote manner. When this architecture is applied on mobile robots, a change of the control point is necessary each time the operator wants to "virtually push" the mobile base instead of the arm. This may augment the operator effort and the task execution time.

To deal with this, in this work, we propose to generate automatically the mobile base motions even when only the

arm is commanded. Without a change of the control point from the arm to the mobile base and without taking care about reaching the limits of the arm workspace, the human operator can continue to drag the arm in a certain direction. If the robotic arm reaches a postural configuration in which the manipulability in the direction of the applied virtual force is low, the mobile base starts contributing to the motion in the direction with low manipulability to ensure that the manipulability measure of the end-effector does not decrease beyond a defined threshold. This permits to reach the desired end-effector goal with less effort for the operator teleoperating the robot while maintaining a certain level of manipulability in the end-effector. Thus, the proposed approach can facilitate the execution of the manipulation task in a way that is transparent and does not require additional attention from the operator.

There are many different measures to evaluate the robot manipulability and dexterity [17]. In this work, we consider the Yoshikawa definition [18] of the manipulability ellipsoid:

$$\boldsymbol{u}^T \ (\boldsymbol{J}\boldsymbol{J}^T)^{-1} \ \boldsymbol{u} \leq 1 \tag{1}$$

where $\boldsymbol{J} \in \mathbb{R}^{3\times N}$ is the linear Jacobian matrix, and $\boldsymbol{u} \in \mathbb{R}^{3\times 1}$ a generic vector. This ellipsoid is strictly related to the arm singularities and to the *Velocity Transmission Ratio* (VTR) β:

$$\beta = (\boldsymbol{u}^T \ (\boldsymbol{J}\boldsymbol{J}^T)^{-1} \ \boldsymbol{u})^{-\frac{1}{2}} \tag{2}$$

Geometrically, the resulting scalar β is the distance along the vector $\boldsymbol{u}$ from the origin to the surface of the manipulability ellipsoid [19]. If the manipulator is seen as a mechanical transformer with joints velocity as input and Cartesian velocity as output, this ratio describe how much joints velocity is necessary to achieve a desired Cartesian velocity [19]. The thinner the ellipsoid is in a certain direction $\boldsymbol{u}$, the lower the VTR is in the $\boldsymbol{u}$ direction. This means that moving the end-effector in this direction not only would require a lot of manipulator effort, but also will make the manipulator always closer to a kinematic singularity; both are cases that are not desirable. In this work we propose to distribute the generated end-effector motion between the arm and the mobile base motions according to the VTR level: when the motion command for the manipulator (as received by the human operator) has a Cartesian direction in which the VTR of the robotic arm is under a certain threshold, the motion is *gradually* distributed and generated by the mobile base maintaining the manipulability of the end-effector above the defined threshold.

Without loss of generality, we consider the three principal axes $\hat{x}, \hat{y}, \hat{z}$ as the directions of interest, and we compute the VTR along these three directions:

$$\begin{aligned}
\beta_x &= ([1\ 0\ 0] \ (\boldsymbol{J}\boldsymbol{J}^T)^{-1} \ [1\ 0\ 0]^T)^{-\frac{1}{2}} \\
\beta_y &= ([0\ 1\ 0] \ (\boldsymbol{J}\boldsymbol{J}^T)^{-1} \ [0\ 1\ 0]^T)^{-\frac{1}{2}} \\
\beta_z &= ([0\ 0\ 1] \ (\boldsymbol{J}\boldsymbol{J}^T)^{-1} \ [0\ 0\ 1]^T)^{-\frac{1}{2}}
\end{aligned} \tag{3}$$

Eq. 3 can be rewritten as:

$$\beta_i = \left([(\boldsymbol{J}\boldsymbol{J}^T)^{-1}]_{i,i}\right)^{-\frac{1}{2}} \qquad \text{for } i = x, y, z \tag{4}$$

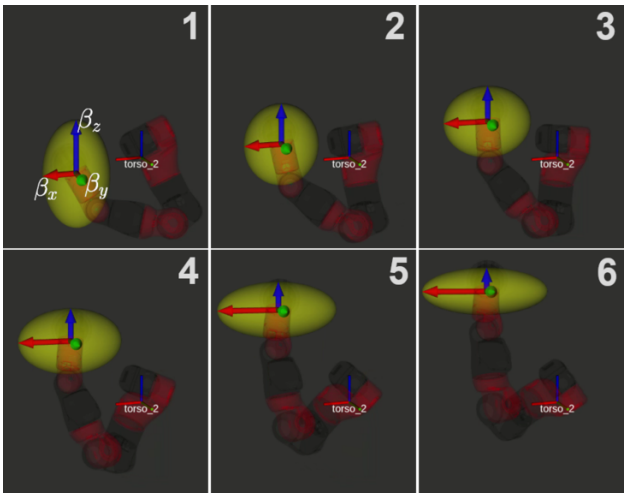

Fig. 3. The VTR along the principal directions $\hat{x}, \hat{y}, \hat{z}$ is changing while the CENTAURO arm is moving upward. The red, green and blue arrows are the yellow manipulability ellipsoid axes, and represent the VTR along the principal directions. The *torso_2* frame is the reference frame.

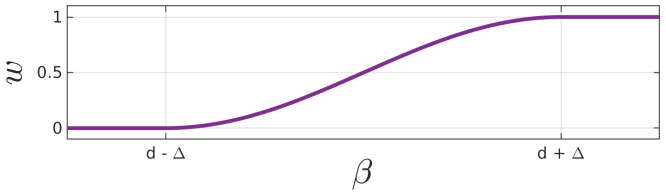

Fig. 4. The function of Eq. 5 used to compute the weight

where $[(\boldsymbol{J}\boldsymbol{J}^T)^{-1}]_{i,i}$ for $i = x, y, z$ indicates the first, second, and third element of the diagonal of $(\boldsymbol{J}\boldsymbol{J}^T)^{-1}$. Fig. 3 introduces an example of the VTR along the selected three directions, changing accordingly to the CENTAURO arm movements. It can be observed how the β_z value (represented by the blue arrow) decreases in the upward direction, indicating the difficulty of the end-effector to move along that direction in specific postures.

From the β_i we compute a weight $w_i \in [0, 1]$, for $i = x, y, z$:

$$w_i = \begin{cases} 1 & \text{if } \beta_i \geq d_i + \Delta_i \\ \xi(\beta_i) & \text{if } d_i - \Delta_i < \beta_i < d_i + \Delta_i \\ 0 & \text{if } \beta_i \leq d_i - \Delta_i \end{cases} \tag{5}$$

where $d_i + \Delta_i$ is a upper threshold over which the VTR is said to be sufficiently high, $d_i - \Delta_i$ is a lower threshold under which the VTR is too low, and $\xi(\beta_i)$ is a sigmoid function introduced to smooth the transition. Eq. 5 is represented in Fig. 4. The idea is to scale the input commands given to the manipulator multiplying their linear Cartesian components x, y, z by the weight w_i. When the VTR is over the threshold $d_i + \Delta_i$ it results $w_i = 1$ and the command's component i is unchanged, while when the VTR is below the minimum threshold $d_i - \Delta_i$ it results $w_i = 0$ and the command component i for the manipulator will be nullified. A smooth transition is assured by the sigmoid function $\xi(\beta_i)$. Similarly, the mobile base velocities will be generated from the same

user command, but with the components multiplied by a $1 - w_i$ factor. For the sake of notation, we will refer in the following paragraphs to the weight diagonal matrix $\boldsymbol{W} \in \mathbb{R}^{3\times3}$:

$$\boldsymbol{W} = \begin{pmatrix} w_x & 0 & 0 \\ 0 & w_y & 0 \\ 0 & 0 & w_z \end{pmatrix} \tag{6}$$

and its dual $\boldsymbol{I} - \boldsymbol{W}$, where $\boldsymbol{I} \in \mathbb{R}^{3\times3}$ is the identity matrix.

Note that with this weight definition the choice of the thresholds $d_i + \Delta_i$ is dependent on the arm characteristics, because the VTR values depend on the arm structure, and hence should be tuned accordingly. To have more robot-independent parameters, it is possible to use a normalized version of the VTR, by scaling the values accordingly to the maximum value achievable. This procedure would be similar to the concept of *normalized manipulability* [17], since the VTR β used in this work is based on the classical manipulability [18].

A characteristic of our approach is that the thresholds can be set differently for the three different directions. This implies that it is possible to maintain different levels of VTR in the different directions, a feature whose usefulness will be shown in the experiments described in Section IV.

Based also on the control law used to command the robot, the weights can be exploited in different ways, as explained in Section III.

III. VTR-based Weighted Motion Generation

The motion generation based on the weight $\boldsymbol{W}$ is discussed in this section within the framework of TelePhysicalOperation introduced in our previous work [29].

Consider the following control law used in our previous work to teleoperate a remote/collaborative N-joints manipulator:

$$\ddot{\boldsymbol{q}}_{ref}(t) = \boldsymbol{M}_{tpo}^{-1}\big(\boldsymbol{K}_{tpo}(\boldsymbol{q}_{eq} - \boldsymbol{q}(t)) - \boldsymbol{D}_{tpo}\dot{\boldsymbol{q}}_{ref}(t-1) + \boldsymbol{\tau}\big) \tag{7}$$

where $\boldsymbol{q}_{ref}(t) \in \mathbb{R}^{N\times1}$ is the joint position reference vector; $\boldsymbol{M}_{tpo}, \boldsymbol{K}_{tpo}, \boldsymbol{D}_{tpo} \in \mathbb{R}^{N\times N}$ are diagonal matrices of the mass, stiffness, and damping parameters of the joint mass-spring-damper model; $\boldsymbol{q}, \boldsymbol{q}_{eq} \in \mathbb{R}^{N\times1}$ are the current position of the joints and the equilibrium set point where a stiffness greater than zero will drag the joints; $\ddot{\boldsymbol{q}}_{ref}$ is integrated twice to obtain the joint command $\boldsymbol{q}_{ref}$.

The torques $\boldsymbol{\tau} \in \mathbb{R}^{N\times1}$ are derived by:

$$\boldsymbol{\tau} = \boldsymbol{J}^T\ \boldsymbol{f} \tag{8}$$

where $\boldsymbol{f} \in \mathbb{R}^{3\times1}$ is a virtual force computed from the displacement of the operator arm gathered from a motion tracking device:

$$\boldsymbol{f} = k_{cam}\ s(\boldsymbol{r}) \tag{9}$$

where $s(\cdot)$ is a simple filter to smooth out the behaviour, and k_{cam} is a positive gain. For more details about these equations, the reader can refer to [29].

In this work we extend the above approach by scaling the virtual force $\boldsymbol{f}$ to generate the torques as follows:

$$\boldsymbol{\tau} = \boldsymbol{J}^T\ \boldsymbol{W}\boldsymbol{f} \tag{10}$$

while the reference velocity $\boldsymbol{\nu} \in \mathbb{R}^{3\times1}$ for the mobile base is computed as:

$$\boldsymbol{\nu} = \boldsymbol{K}_{\nu}(\boldsymbol{I} - \boldsymbol{W})\boldsymbol{f} \tag{11}$$

where $\boldsymbol{K}_{\nu} \in \mathbb{R}^{3\times3}$ is a diagonal matrix to maintain the consistency between the different physical quantities. When the VTR in one or more of the three principal directions is below the threshold, the corresponding elements in $\boldsymbol{W}$ change from 1 to 0. This would result in the virtual force scaled down to compute the torques (Eq. 10), and accordingly the base velocity ν scaled up (Eq. 11). Hence, the arm motion will be reduced along the direction of low VTR, and the mobile base motion will compensate with a proper velocity. If in some directions the VTR is above the threshold, the respective component of the manipulator reference is not modified, and the base will not move in these directions.

Note that the weight $\boldsymbol{W}$ will not affect the returning elastic component of the joint mass-spring-damper model of Eq. 7. Therefore, with a proper setting of the diagonal elements of $\boldsymbol{K}$, the arm is further helped to recover from a posture where the VTR is low thanks to the returning elastic element that brings the arm back toward the $\boldsymbol{q}_{eq}$ equilibrium, which should be set accordingly to result in an arm posture of good manipulability.

Consider now the more general case, in which a control law generates Cartesian velocities for the manipulator. These velocities can be scaled according to the VTR in a similar way as the virtual forces are scaled above for the case of the TelePhysicalOperation control. Hence, given a desired end-effector linear Cartesian velocity $\dot{\boldsymbol{x}}^* \in \mathbb{R}^{3\times1}$, the scaled reference Cartesian velocity $\dot{\boldsymbol{x}} \in \mathbb{R}^{3\times1}$ and the scaled reference Cartesian velocity for the mobile base $\boldsymbol{\nu}$ can be derived as follows:

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{W}\dot{\boldsymbol{x}}^* \\ \boldsymbol{\nu} &= (\boldsymbol{I} - \boldsymbol{W})\dot{\boldsymbol{x}}^* \end{aligned} \tag{12}$$

Similarly to the case of the TelePhysicalOperation control, this will result to the robotic arm slowing down towards the directions in which the VTR is below the threshold if the desired end-effector linear Cartesian velocity $\dot{\boldsymbol{x}}^*$ has some components towards these directions. Accordingly, the velocity of the mobile base will increase towards these directions of low VTR level as compensation.

So far, we have silently considered the case of single arm operations. A similar approach for the weighted motion generation can though be used for bimanual tasks, e.g., for bimanual grasping and transportation of an object. In such a task scenario, it may be more comfortable to consider directly the grasped object as a reference point for the commanded velocities. These velocities can be generated with the TelePhysicalOperation motion tracking device as:

$$\dot{\boldsymbol{x}}^* = \boldsymbol{K}_{cart}\boldsymbol{f} \tag{13}$$

where $\boldsymbol{K}_{cart} \in \mathbb{R}^{3\times3}$ is a diagonal matrix of positive gains. The commanded velocities applied to the grasped object can be then scaled according to Eq. 12; where the computation

of the weight $\boldsymbol{W}$ can be derived from the arm with the worst condition of manipulability.

IV. Experiments

We have validated our manipulability-aware shared loco-manipulation motion generation by performing a number of tasks with the CENTAURO platform, a quadruped body with wheels and a human-like torso with two arms [28]. The communication with the robotic system has been made possible thanks to the XBot Architecture [30]. A Cartesian task has been set with the *CartesI/O* Control Framework [31] to command the quadruped body as a mobile platform capable of following planar directions and the vertical direction with a *squatting* motion.

A. Reaching Specific Locations with the End-Effector

In the first task, the robot is teleoperated to precisely reach with the left arm three specific locations in the surrounding environment, where three buttons must be pressed. Given the distances between the three locations, reaching them requires both manipulator arm and mobile base motions, as shown in Fig. 5. The task is executed twice: with and without the manipulability-aware motion generation proposed in this work. When it is not used, the operator must repeatedly change the control between the arm and the mobile base. Instead, with the new approach, the operator can continuously drive only the arm while the underlying architecture of the proposed method generates the mobile base velocities according to the manipulator VTR level. Fig. 6 presents the inputs generated by the operator without (top plot) and with (bottom plot) the manipulability-aware motion generation. In the top plot, the dead times during the operation are more evident due to the change of the control point from *End-Effector* to *Locomotion* and vice versa, to respectively generate end-effector postural motion with Eqs. 7 and 8, and mobile body Cartesian motion with $\dot{\boldsymbol{x}} = \boldsymbol{K}_{cart}\ \boldsymbol{f}$ [29]. This results in the operator performing more actions and putting more effort, also slowing the execution of the task. Instead, in the bottom plot, we can see that there are no control point switches; the operator controls always the *End-Effector* while the proposed method generates the mobile base motions when the end-effector is in a region of low manipulability, according to Eqs. 7, 10, and 11. In Fig. 7, the details of the manipulability-aware motion generation are illustrated. The columns represent the three principal axes $\hat{x}, \hat{y}, \hat{z}$, so the data in each row is divided in the x, y, z components. In the top row, the VTR measure (Eq. 4), represented together with the threshold margins $d - \Delta, d, d + \Delta$ (dotted lines), governs the activation of the manipulability-aware motion generation. The highlighted areas represent the time intervals when the VTR is below the $d + \Delta$ threshold, which triggers the activation of the weight w (represented in the second row) according to Eq. 5. In the third row, the user input desired virtual forces (Eq. 9) are shown (note that the same data is also shown in the bottom plot of Fig. 6). According to the weight computed, the forces are scaled to generate the joint torques of the arm $\boldsymbol{\tau}$ (Eq. 10) and the mobile base velocity ν (Eq. 11). The scaled forces and the velocity ν are plotted in the last two bottom plots, respectively. It can be observed in the highlighted areas how the desired forces are diminished in favour of the mobile base velocities.

B. Box Approaching and Transporting

In this task, the robot must approach a box, grasp it with the two arms, and transport it safely to another location. The box approaching and grasping is a phase performed autonomously, while the box transportation is teleoperated by the operator with the goal to pose the box into a specific location. During the approaching phase, the box pose is detected and tracked by the robot vision system based on *ArUco* markers[1]. From the end-effectors given goals (a point near the box sides), Cartesian velocities for each arm are generated. During the execution, based on the VTR of each end-effector, the Cartesian velocities commanded are balanced between each arm and the mobile base using Eq. 12. The final velocity commanded to the mobile base is the average of the two ν computed from the weight of each arm. The related plots for both arms are visible in Fig. 9, where the plot disposition has already been explained in Section IV-A for Fig. 7. The difference is that in these graphs the third row presents the desired Cartesian velocities $\dot{\boldsymbol{x}}^*$ generated, and the fourth row illustrates their scaled version (Eq. 12). Note that the noise visible in the first part of the z component is related to the uncertainties in the marker vision tracking system. Concerning the VTR thresholds, in this approaching phase of the experiment, the threshold along the $\hat{y}$ direction was set to zero, to prevent the motion of the mobile base laterally, which may cause loss of the marker tracking. This shows the flexibility of our approach: thanks to the decomposition of the manipulability analysis into the three directions we can better adapt to the specific task, instead of using the classical scalar manipulability value [18].

Having positioned the end-effectors of the robot on the sides of the object, the box is squeezed until a user-defined initial force is reached by the end-effectors, and finally the box is lifted up by squatting up the robot. At this point, the box can be transported by teleoperating the robot. A desired box velocity is generated by the operator according to Eq. 13. Using the introduced approach, we generate a scaled box velocity and a velocity for the mobile base (Eq. 12), according to the weight computed from the robot arm with the worst manipulability. The arms of the robot move cooperatively to follow the scaled box velocity while keeping the grasping. Fig. 11 shows equivalent data as Fig. 7 and Fig. 9, where now the third row shows the box velocities generated by the operator, and the fourth row presents their scaled version. Concerning the VTR thresholds, we have set high values for $\hat{x}$ and $\hat{y}$ directions, to prevent significant motions in the arms along these directions in order to help maintaining the grasp of the box. In this way the velocity

[1] `http://wiki.ros.org/aruco_detect`

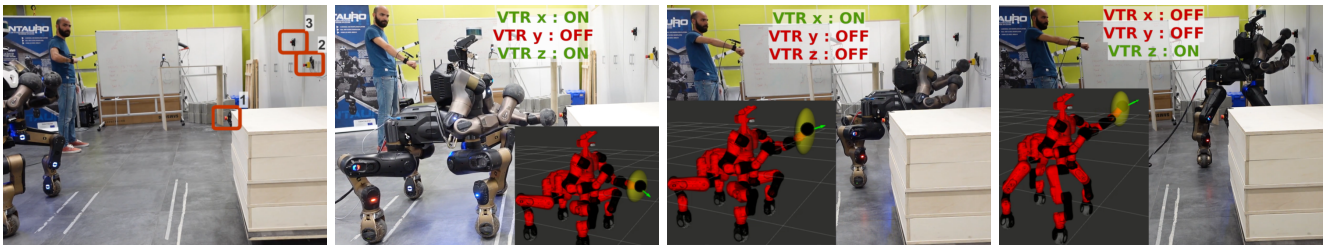

Fig. 5. Sequences of the end-effector locations reaching experiment, performed with the manipulability-aware motion generation. The left arm of the robot is teleoperated by the user right arm with the aim to press the three buttons highlighted in the leftmost image. In the other images, it is indicated in which directions the manipulability-aware motion generation is active during the teleoperation. The robot kinematic visualization shows the input directions commanded by the user (green arrow) and the manipulability ellipsoid of the robot left end-effector (yellow shape).

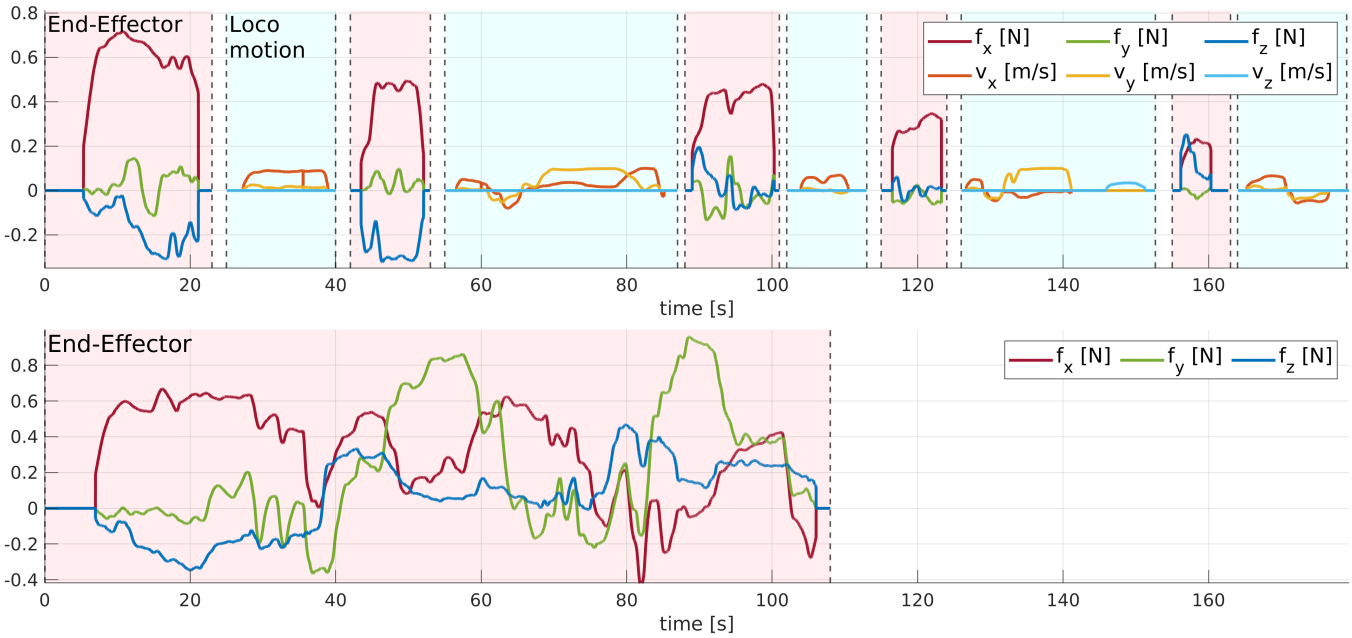

Fig. 6. Plots for the end-effector locations reaching experiment. The data shows the inputs provided by the user with the TPO suit motion tracking interface. At the top, the proposed manipulability-aware motion generation is not used. It can be noticed how the operator has to switch repeatedly between End-Effector control and Locomotion control, increasing dead times and the overall the execution time. At the bottom, the manipulability-aware motion generation is used, such that the operator does not have to comply with the control of the mobile base, and can transparently command only the end-effector.

commands sent by the operator along $\hat{x}$ and $\hat{y}$ mainly result in the generation of mobile base velocities, effectively enabling the transport of the grasped object to another location.

All the described experiments have been recorded, resulting in the video attached with this paper, available also at `https://youtu.be/7YqfVn8XvNk`.

V. Conclusions

This work presented a framework to generate combined locomanipulation motions in a shared control fashion while a human operaror is teleoperating a mobile manipulator. The method explores the concept of manipulability, to distribute the motion input of the operator to the robotic arm also to the mobile base when necessary, i.e., when the end-effector is in a region of low manipulability. To augment the flexibility of the approach and permit the operator to choose to maintain a high manipulability in certain directions, the method does not explore the classical scalar manipulability measure, but the *Velocity Transmission Ratio* (VTR) in the three principal directions. This results in the robot arm end-effector maintaining higher manipulability along specific directions, which can be selected based on the teleoperation task requirements. Such a choice will result in generating more mobile base motions in these directions in order to keep the robotic arm end-effector manipulability high along these directions.

The method was demonstrated and evaluated in a number of different tasks, performed with the TelePhysicalOperation interface [29], and with the CENTAURO robot, demonstrating the efficacy of scaling the robot arm and mobile base motions in a transparent manner to the human operator, eliminating the need to switch between manipulator and mobile base control modes. Future works will explore the proposed method in dual arm non-symmetric tasks in combination with additional local autonomy features to assist in the execution, diminishing the task completion time, and to reduce the operator burden in these more complex tasks.

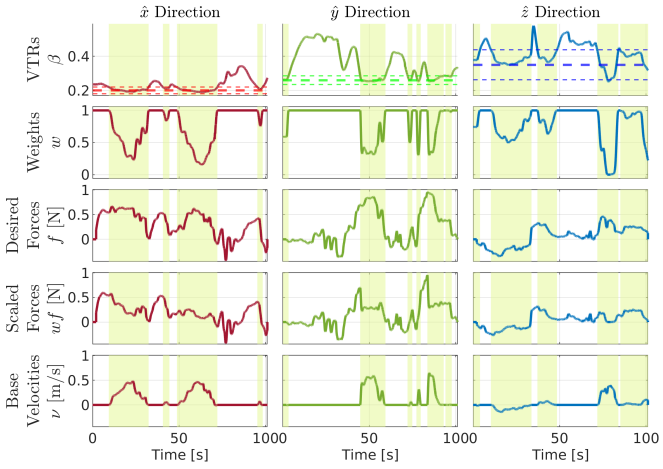

Fig. 7. Plots for the end-effector locations reaching experiment. Each column shows the data for a particular direction, $\hat{x}, \hat{y}, \hat{z}$. From top to bottom rows, in each column there are shown the x, y, z components of: the VTR (Eq. 4), where the thresholds margins $d - \Delta, d, d + \Delta$ are represented by the dotted horizontal lines; the weight computed from the VTR (Eq. 5); the desired forces commanded by the operator (Eq. 9); the scaled desired forces (included in the Eq. 10); the weighted mobile base velocities (Eq. 11). The coloured region represents where the VTR is below the given thresholds.

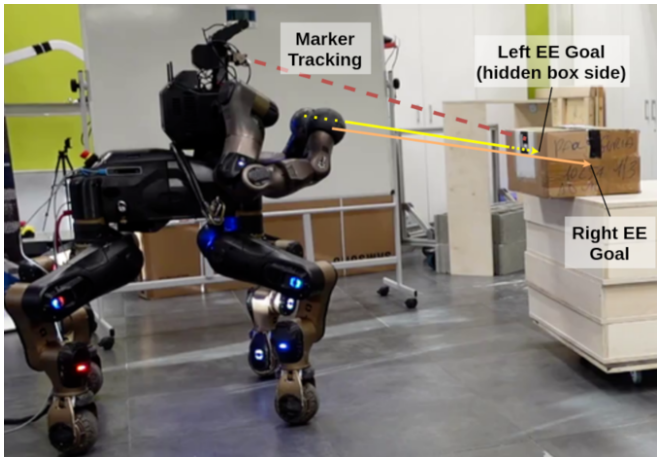

Fig. 8. The robot is autonomously approaching the box and placing the end-effectors on the box sides.

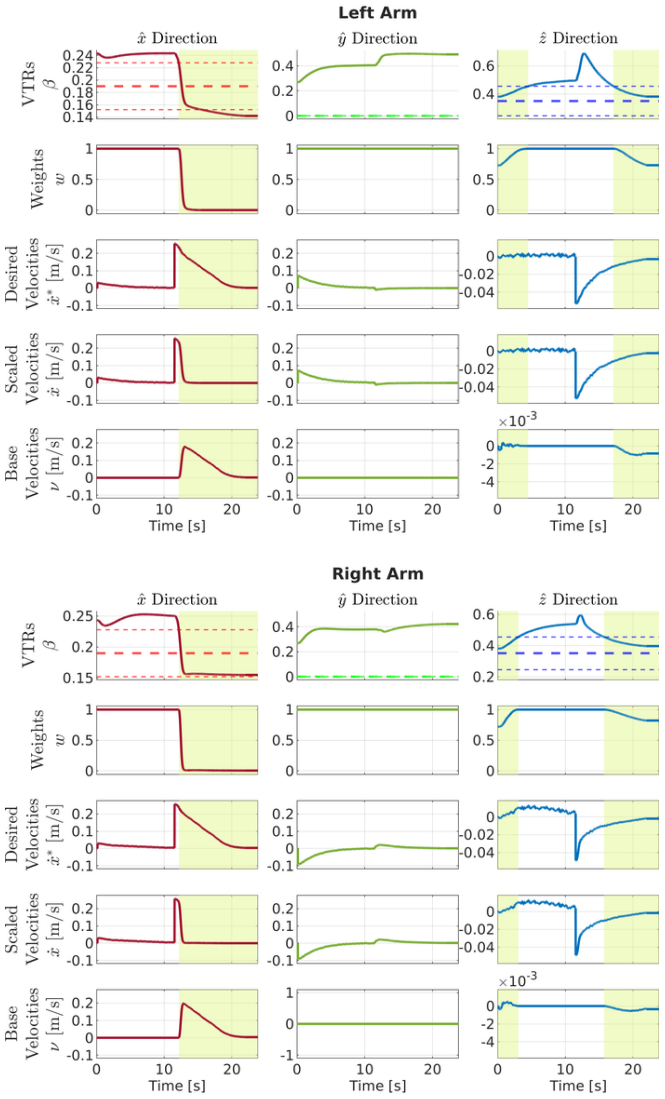

Fig. 9. Plots for the box approaching task. For each arm, each column shows the data for a particular direction, $\hat{x}, \hat{y}, \hat{z}$. From top to bottom rows, in each column there are shown the x, y, z components of: the VTR (Eq. 4), where the thresholds margins $d - \Delta, d, d + \Delta$ are represented by the dotted horizontal lines; the weight computed from the VTR (Eq. 5); the desired end-effector velocities to reach the box sides; the scaled desired velocities (Eq. 12); the weighted mobile base velocities (Eq. 12). The coloured region represents where the VTR is below the given thresholds.

REFERENCES

[1] Y. Liu and G. Nejat, "Robotic urban search and rescue: A survey from the control perspective," *J. Intell. Robot. Syst.*, vol. 72, pp. 147–165, 2013.

[2] M. A. Goodrich, J. W. Crandall, and E. Barakova, "Teleoperation and beyond for assistive humanoid robots," *Rev. Human Factors Ergonom.*, vol. 9, no. 1, pp. 175–226, 2013.

[3] S. Lu, Y. Zhang, and J. Su, "Mobile robot for power substation inspection: a survey," *IEEE/CAA J. Autom. Sin.*, vol. 4, no. 4, pp. 830–847, 2017.

[4] V. Villani, F. Pini, F. Leali, and C. Secchi, "Survey on human-robot collaboration in industrial settings: Safety, intuitive interfaces and applications," *Mechatronics*, vol. 55, pp. 248–266, 2018.

[5] G. Carra *et al.*, "Robotics in the construction industry: State of the art and future opportunities," in *Int. Symp. Automat. Robot. Construction*, 2018, pp. 866–873.

[6] M. Selvaggio, M. Cognetti, S. Nikolaidis, S. Ivaldi, and B. Siciliano, "Autonomy in physical human-robot interaction: A brief survey," *IEEE Robot. Autom. Lett.*, vol. 6, no. 4, pp. 7989–7996, 2021.

[7] F. Abi-Farraj, N. Pedemonte, and P. Robuffo Giordano, "A visual-based shared control architecture for remote telemanipulation," in *IEEE Int. Conf. Intell. Robots Syst.*, 2016, pp. 4266–4273.

[8] A. Broad, T. Murphey, and B. Argall, "Highly parallelized data-driven mpc for minimal intervention shared control," in *Robot. Sci. Syst.*, 2019.

[9] C. Masone, M. Mohammadi, P. R. Giordano, and A. Franchi, "Shared planning and control for mobile robots with integral haptic feedback," *Int. J. Robot. Res.*, vol. 37, no. 11, pp. 1395–1420, 2018.

[10] M. Shahbazi, J. Lee, D. Caldwell, and N. Tsagarakis, "Inverse

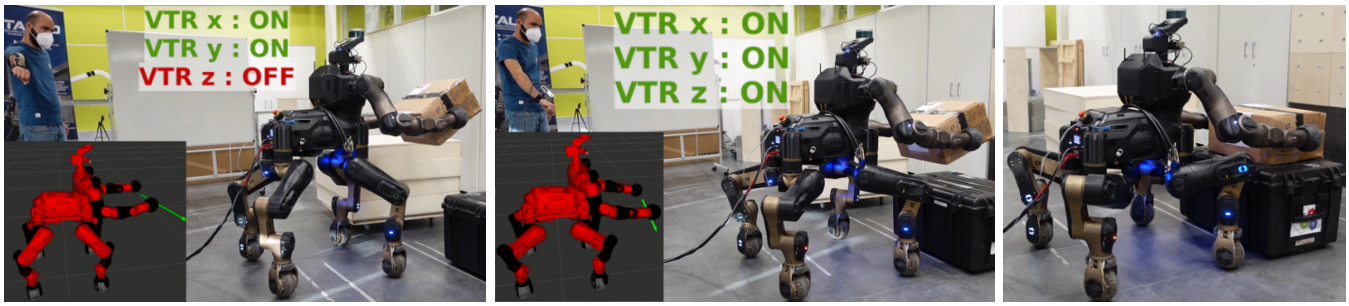

Fig. 10. Sequences from the box transportation task. From left to right: (1) the VTR is below the threshold on the x and y directions, making the mobile base moving in these directions according to the user commanded box velocity; (2) the VTR is below the threshold also on the z direction, making the robot to squat to follow the box velocity in the z direction; (3) the box has been placed in the final location.

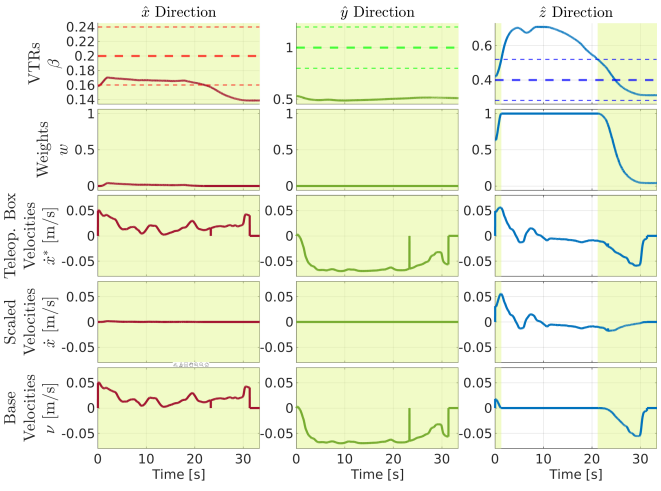

Fig. 11. Plots for the box transportation task. The data is relative to the arm which has the lowest VTR at each instant. Each column show the data for a particular direction, $\hat{x}, \hat{y}, \hat{z}$. From top to bottom rows, in each column there are shown the x, y, z components of: the VTR (Eq. 4), where the thresholds margins $d - \Delta, d, d + \Delta$ are represented by the dotted horizontal lines; the weight computed from the VTR (Eq. 5); the box Cartesian velocities commanded by the operator (Eq. 13); the scaled box velocities (Eq. 12); the weighted mobile base velocities (Eq. 12). The colored region represents where the VTR is below the given thresholds.

dynamics control of bimanual object manipulation using orthogonal decomposition: An analytic approach," in *IEEE Int. Conf. Intell. Robots Syst.*, 2017, pp. 4791–4796.

[11] M. Laghi, M. Maimeri, M. Marchand, C. Leparoux, M. Catalano, A. Ajoudani, and A. Bicchi, "Shared-autonomy control for intuitive bimanual tele-manipulation," in *IEEE-RAS Int. Conf. Humanoid Robots*, 2018, pp. 1–9.

[12] Y. Nakamura, H. Hanafusa, and T. Yoshikawa, "Task-priority based redundancy control of robot manipulators," *The International Journal of Robotics Research*, vol. 6, no. 2, pp. 3–15, 1987.

[13] B. Siciliano and J.-J. Slotine, "A general framework for managing multiple tasks in highly redundant robotic systems," in *Int. Conf. Adv. Robot.*, vol. 2, 1991, pp. 1211–1216.

[14] N. Mansard, O. Khatib, and A. Kheddar, "A unified approach to integrate unilateral constraints in the stack of tasks," *IEEE Trans. Robot.*, vol. 25, no. 3, pp. 670–685, 2009.

[15] L. Sentis and O. Khatib, "A whole-body control framework for humanoids operating in human environments," in *IEEE Int. Conf. Robot. Autom.*, 2006, pp. 2641–2648.

[16] A. Dietrich, T. Wimbock, A. Albu-Schaffer, and G. Hirzinger, "Reactive whole-body control: Dynamic mobile manipulation using a large number of actuated degrees of freedom," *IEEE Robot. Autom. Mag.*, vol. 19, no. 2, pp. 20–33, 2012.

[17] S. Patel and T. Sobh, "Manipulator performance measures - a comprehensive literature survey," *J. Intell. Robot. Syst.*, vol. 77, no. 3, pp. 547–570, Mar 2015.

[18] T. Yoshikawa, "Manipulability of robotic mechanisms," *Int. J. Robot. Res.*, vol. 4, no. 2, pp. 3–9, 1985.

[19] S. L. Chiu, "Task compatibility of manipulator postures," *Int. J. Robot. Res.*, vol. 7, no. 5, pp. 13–21, 1988.

[20] S. Lee, "Dual redundant arm configuration optimization with task-oriented dual arm manipulability," *IEEE Trans. Robot. Autom.*, vol. 5, no. 1, pp. 78–97, 1989.

[21] A. Bicchi and D. Prattichizzo, "Manipulability of cooperating robots with unactuated joints and closed-chain mechanisms," *IEEE Trans. Robot. Autom.*, vol. 16, no. 4, pp. 336–345, 2000.

[22] Y. Yamamoto and X. Yun, "Coordinating locomotion and manipulation of a mobile manipulator," in *IEEE Conf. Decis. Control*, vol. 3, 1992, pp. 2643–2648.

[23] J. F. Gardner and S. A. Velinsky, "Kinematics of mobile manipulators and implications for design," *J. Robot. Syst.*, vol. 17, no. 6, pp. 309–320, 2000.

[24] B. Bayle, J.-Y. Fourquet, and M. Renaud, "Manipulability of wheeled mobile manipulators: Application to motion generation," *Int. J. Robot. Res.*, vol. 22, no. 7-8, pp. 565–581, 2003.

[25] J. Leoro and T. Hsiao, "Motion planning of nonholonomic mobile manipulators with manipulability maximization considering joints physical constraints and self-collision avoidance," *Appl. Sci.*, vol. 11, no. 14, 2021.

[26] Y. Zhang, X. Yan, D. Chen, D. Guo, and W. Li, "Qp-based refined manipulability-maximizing scheme for coordinated motion planning and control of physically constrained wheeled mobile redundant manipulators," *Nonlinear Dyn.*, vol. 85, no. 1, pp. 245–261, 2016.

[27] Y. Wang, C. Smith, Y. Karayiannidis, and P. Ögren, "Whole body control of a dual-arm mobile robot using a virtual kinematic chain," *Int. J. Humanoid Robot.*, vol. 13, no. 01, p. 1550047, 2016.

[28] T. Klamt *et al.*, "Flexible disaster response of tomorrow: Final presentation and evaluation of the centauro system," *IEEE Robot. Autom. Mag.*, vol. 26, no. 4, pp. 59–72, 2019.

[29] D. Torielli, L. Muratore, A. Laurenzi, and N. Tsagarakis, "Telephysicaloperation: Remote robot control based on a virtual "marionette" type interaction interface," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 2479–2486, 2022.

[30] L. Muratore, A. Laurenzi, E. Mingo Hoffman, and N. Tsagarakis, "The XBot real-time software framework for robotics: From the developer to the user perspective," *IEEE Robot. Autom. Mag.*, vol. 27, no. 3, pp. 133–143, 2020.

[31] A. Laurenzi, E. M. Hoffman, L. Muratore, and N. Tsagarakis, "CartesI/O: A ROS Based Real-Time Capable Cartesian Control Framework," in *IEEE Int. Conf. Robot. Autom.*, 2019, pp. 591–596.