Online Telemanipulation Framework on Humanoid for both Manipulation and Imitation

Daegyu Lim[1], Donghyeon Kim[1] and Jaeheung Park[1,2]

***Abstract*— Teleoperation of the robot is a promising technology that brings the robot to the real life because the complex decision-making algorithm can be replaced by the supervision of the human operator. To teleoperate the high dimensional humanoid robot intuitively like moving the operator's own body, a motion retargeting algorithm is required to bridge the kinematics and dynamics differences between the robot and the operator. We propose the telemanipulation framework for the upper body of the humanoid only using 6 Virtual Reality (VR) trackers. The framework consists of the pose calibration procedure, the motion mapping method, and the multi-task control using Hierarchical Quadratic Programming (HQP). To validate the effect of the proposed method, three experiments are conducted including a user study for analysis of the hand position mapping, a dual-arm telemanipulation task in a real robot, and a comparison of the mapping methods in visual similarity.**

I. Introduction

Teleoperation of the robot is a promising technology which brings the robot to the real life because developing the general decision making and high-level planning algorithms can be replaced by the human's manual command. Using the teleoperation robot, dangerous missions for human beings, such as disaster relief and space exploration, can be conducted safely in remote place. Moreover, the teleoperation robot can be used as a means of remote physical and social interaction.

For successful remote control of humanoids, an intuitive and easy-to-use control interface is crucial. Taking the advantage of the human-like kinematic structure of the humanoid robot, the operator can control the humanoid robot similarly to the extent of using their bodies. But, due to the different kinematic and dynamic features between the robot and the operator, a motion retargeting algorithm is required which transforms the motion of the human to the motion of the robot. Many motion retargeting algorithms are studied mainly in the computer graphics and robotics fields.

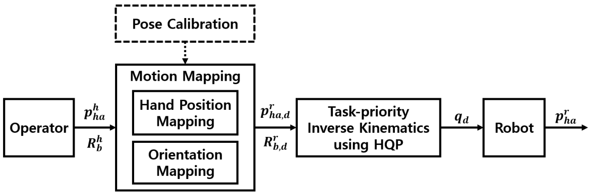


Fig. 1. Overview of the proposed teleoperation scheme.

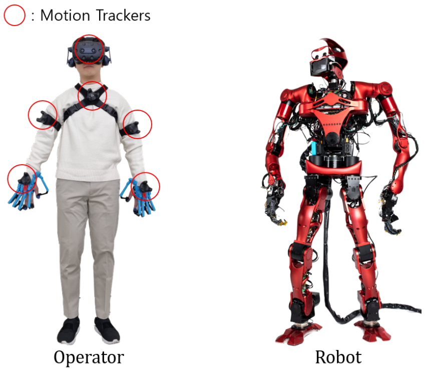


Fig. 2. Teleoperation system of the high dimensional humanoid.

In graphics studies, the prior goal of motion retargeting is the visual similarity between the actor and the character. As in [1], [2], various deep-learning based motion retargeting algorithms are presented. All these algorithms exploited encoder and decoder structures to train the network in an unsupervised learning manner using hidden or latent variables. The learning-based algorithms reconstruct the source motion to the target character naturally even though the character has a much different skeleton structure from the source skeleton. Also, from the Disney Research group, motion retargeting algorithms [3]–[5] to the real robots are proposed. In [3], an optimization-based offline motion retargeting algorithm is presented and experimented on the various robots. In [4] and [5], the deep latent variable is used for the motion retargeting and demonstrated successful results. However, these learning-based algorithms are not suitable for real-time teleoperation of the robot because they require source motion data from the operator and should be trained ahead of the test, which means whenever the operator is changed the retargeting network should be trained again for the new operator. Additionally, the nonlinear mapping algorithm makes it difficult for the operator to become familiar with teleoperation and perform the task intuitively.

On the other hand, in the robotics field, the main purpose of teleoperation is to perform remote tasks successfully. For the accurate telemanipulation of the full-size humanoid robot, the majority of works in robotics use one to one hand position mapping [6]–[8]. The one-to-one mapping generates

*This work was partly supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2021-0-00896, Development of 3D collaborative teleoperation technology for unstructured work in harsh environments) and a National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No. 2021R1A2C3005914).

[1] Daegyu Lim, Donghyeon Kim and Jaeheung Park are with the Department of Intelligence and Information, and Automation and Systems Research Institute, Seoul National University, Seoul, KS013, Republic of Korea (e-mail: dgyo3784, kdh0429, park73@snu.ac.kr).

[2] Jaeheung Park is also with the Advanced Institutes of Convergence Technology, Suwon, KS002, Republic of Korea.

the desired hand motion of the robot as the displacement of the operator's hand on an identical scale. Some of the studies applied hand position mapping in proportion to arm's length between the operator and the robot, especially for small robots; Nao [9] and iCub [10]. Joint angle mapping methods are also adopted for the simple telemanipulation and imitation task using RGBD camera-based motion capturing system and the Nao robot [11], [12]. In these papers, the joint angles of the human are calculated using the position of the joints of the skeleton. And the measured joint angles of the human are applied to the robot's corresponding axes. But, according to the experimental results from [13], the joint mapping method could result in a longer adaptation time in the teleoperation system than the task space mapping method although the subjects adapt to each mapping methods and show similar reaction time in the end.

In this paper, the teleoperation control framework is presented for the upper body of the full-size humanoid robot. The linear hand position mapping method based on the calibration pose data and the orientation mapping method is proposed for the fast adaptation to the teleoperation system and similar appearance at the same time. The desired pose of the robot from the mapping method are solved using the HQP. Our contribution is that the development of the entire motion mapping procedure which is capable of both telemanipulation and imitation. Also, the proposed mapping method and the control framework are validated by three experiments. Note that this telemanipulation framework is used as the main control algorithm of the *Team SNU* in the ANA AVATAR XPRIZE Semifinals.

II. Motion Mapping Method

In this chapter, the proposed motion mapping method is introduced. The motion mapping method consists of hand position mapping and orientation mapping, which are based on the pose calibration data as presented in Fig.1, a block diagram of the proposed teleoperation scheme. The hand position mapping translates the operator's hand position to the desired hand position of the robot enabling both manipulation and pose imitation. Except for the hand position, other body parts are retargeted to the robot using orientation mapping. In this chapter, more details about the pose calibration method, the hand position mapping method, and the orientation mapping method will be introduced. Before introducing the proposed mapping algorithm, the overview of the teleoperation system is described firstly.

A. Overview of Teleoperation System

In this paper, the humanoid robot, TOCABI, is used for the teleoperation as shown in Fig.2. TOCABI has 33DoF (12 in legs, 3 in waist, 16 in arms, and 2 in neck) which is one of the highest dimensional humanoid robots. The robot weighs 100kg, and its height is 180cm. It is a torque-controlled robot without joint torque sensors and the frequency of torque command is 2000Hz.

The master system consists of Vive VR equipment including five trackers on the body and one Head Mount Display (HMD) as shown in Fig.2. From these 6 sensors, position and orientation information of each sensor can be obtained. There are two main reasons for choosing VR products as a motion capture system rather than the other types of system. First, wearing VR trackers with elastic bands are much more comfortable and easy than other types of motion capture equipment; wearing a full-body motion suit or attaching motion capture markers at the specific body points. Second, VR products are more stable than the inertial sensor-based motion capture system against the disturbance of the magnetic field. For dexterous telemanipulation, other kinds of equipment such as data gloves or haptic devices are used with motion tracking sensors, and magnetic materials in that equipment, such as a vibrator, can interfere with the inertial sensors. In some research, a mechanical master arm system is used for obtaining hand pose information and force feedback. The mechanical arm is more stable than inertial or optical sensor-based motion tracking systems including VR trackers against environmental disturbances such as magnetic fields and lights. But, it is expensive to build and hard to move and carry the system. Therefore, we chose to use VR trackers for a convenient and portable master system.

TABLE I
Notation of Variable

Symbol	Description
a	scalar a
$\mathbf{a}$	vector $\mathbf{a}$
$\mathbf{A}$	matrix $\mathbf{A}$
$\mathbf{p}_b$	position vector for the body frame b
$\mathbf{R}_b$	rotation matrix for the body frame b
i	refers whether left or right
$\mathbf{O}$	zero elements matrix
$\|\|\mathbf{x}\|\|^2$	L-2 norm of the vector x
$\|\|\mathbf{x}\|\|_Q^2$	L-2 norm of the vector x weighted by Q

B. Human Model and Pose Calibration

To measure the pose of the operator, six motion tracking sensors are attached to the operator's body parts (chest, head, each upper arm, and each hand) as shown in Fig.2. However, it is difficult to calibrate the structure of the operator's skeleton and reconstruct the entire motion of the operator correctly using only six sensor information. Therefore, a simplified human model is used for fast human model calibration and effective hand position mapping. The simplified human model consists of 5 points; one center of the chest ($\mathbf{p}^h_{ch}$), two shoulders ($\mathbf{p}^h_{sh,i}$), and two hands ($\mathbf{p}^h_{ha,i}$) as shown in Fig.3. Because the shoulder's movement is not observed directly, it is estimated by assuming that the two shoulder points and one chest point compose a rigid triangle, and the orientation of both shoulder frames is equal to the orientation of the chest frame. Then, the transformation of the shoulder frame can be estimated using the chest information and the offset vector between the chest and the shoulders.

Three different calibration pose is adopted to estimate the position offset of the shoulder joint relative to the chest as shown in Fig.3; Still-pose, T-pose, and Forward-pose. During

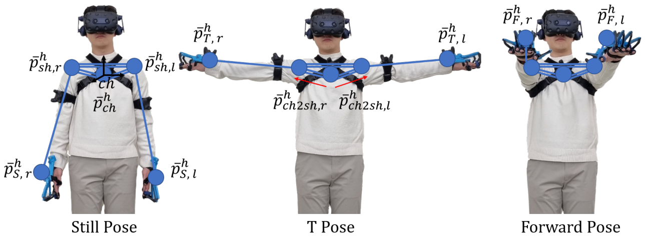
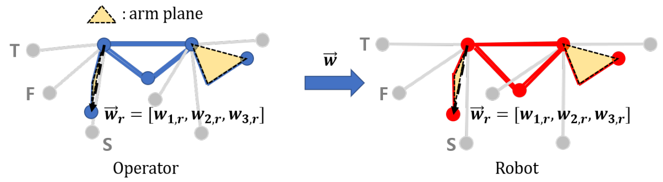


Fig. 3. Simplified human model and three calibration poses.



: arm plane
T
F
S
$\vec{w}_r = [w_{1,r}, w_{2,r}, w_{3,r}]$
Operator
$\vec{w}$
T
F
S
$\vec{w}_r = [w_{1,r}, w_{2,r}, w_{3,r}]$
Robot

Fig. 4. A diagram of hand position mapping based on the three calibration poses.

the calibration process, the hand position vector is saved corresponding to each pose ($\bar{\mathbf{p}}^h_{S,i}, \bar{\mathbf{p}}^h_{T,i}, \bar{\mathbf{p}}^h_{F,i}, i \in \{l, r\}$). Also, during the *Still-pose* calibration, not only hand positions but also rotation matrices of the chest and the upper arm trackers ($\bar{\mathbf{R}}^h_b$, $b \in \{ch, ua\}$) are saved to measure the angular displacement of the rotation matrix for the relative orientation mapping in Section II-D.

Using these three calibration data, the center of the shoulder joint ($\bar{\mathbf{p}}^h_{sh,i} = \{\bar{x}^h_{sh,i}, \bar{y}^h_{sh,i}, \bar{z}^h_{sh,i}\}$) is estimated according to the following steps. First, find the circumscribed circle (C_{cali}) of the triangle consisting of the three calibration points and the normal vector ($\vec{\mathbf{n}}_{cali}$) perpendicular to the circumscribed circle. Second, among the points on the line passing the center of C_{cali} and parallel to the $\vec{\mathbf{n}}_{cali}$, calculate the three candidate points of the shoulder. The two vectors from the candidate shoulder point to the one of the calibration points should be perpendicular. If three candidates of the shoulder point are obtained according to the three combinations of the calibration poses, finally, average them for the estimated center of the shoulder joint.

$$\begin{aligned} &(\bar{\mathbf{p}}^h_{S,i} - \bar{\mathbf{p}}^h_{sh,i,j}) \cdot (\bar{\mathbf{p}}^h_{T,i} - \bar{\mathbf{p}}^h_{sh,i,j}) = 0, \;\; j \in \{1,2,3\} \\ &\bar{\mathbf{p}}^h_{sh,i} = \frac{\bar{\mathbf{p}}^h_{sh,i,1} + \bar{\mathbf{p}}^h_{sh,i,2} + \bar{\mathbf{p}}^h_{sh,i,3}}{3}. \end{aligned} \tag{1}$$

Once the center of the shoulder joint is calculated, the offset vector from the chest position to the shoulder position with respect to the chest frame is saved as ${}^{ch}\bar{\mathbf{p}}_{ch2sh,i} = \bar{\mathbf{R}}^T_{ch}(\bar{\mathbf{p}}_{sh,i} - \bar{\mathbf{p}}_{ch})$ where $\bar{\mathbf{R}}^T_{ch}$, $\bar{\mathbf{p}}_{sh,i}$ and $\bar{\mathbf{p}}_{ch}$ are the initial values saved during calibration process. Finally, the operator's current shoulder position ($\mathbf{p}_{sh,i}$) can be estimated using the chest tracker position ($\mathbf{p}_{ch}$), orientation ($\mathbf{R}_{ch}$), and offset vector (${}^{ch}\bar{\mathbf{p}}_{ch2sh,i}$) as follows,

$$\mathbf{p}_{sh,i} = \mathbf{p}_{ch} + \mathbf{R}_{ch}\,{}^{ch}\bar{\mathbf{p}}_{ch2sh,i}. \tag{2}$$

C. Hand Position Mapping

The proposed hand position mapping from human to the robot utilizes primitive poses in task space for both human and the robot, and mapping parameters ($w_{n,i}$, $n \in \mathbb{N}$) between human and the robot. The three calibration poses in the shoulder frame can be used as primitive poses to express the current position of the operator's hands as a linear combination of the calibration vectors using mapping parameters. This approach has the advantage in that if the operator holds one of the primitive poses, the robot also poses the corresponding pose exactly that we design, and if the operator's hand is located somewhere between the primitive poses, the robot's hand position can also be interpolated using a mapping variable as shown in Fig.4. For example, the operator's left hand can be expressed in left shoulder frame as ${}^{sh,l}\mathbf{p}^h_{ha,l} = \mathbf{w}_{1,l}\,{}^{sh,l}\bar{\mathbf{p}}^h_{S,l} + \mathbf{w}_{2,l}\,{}^{sh,l}\bar{\mathbf{p}}^h_{T,l} + \mathbf{w}_{3,l}\,{}^{sh,l}\bar{\mathbf{p}}^h_{F,l}$. The reason for using the shoulder coordinate is that the three calibration vectors (${}^{sh,i}\bar{\mathbf{p}}^h_{S,i}, {}^{sh,i}\bar{\mathbf{p}}^h_{T,i}, {}^{sh,i}\bar{\mathbf{p}}^h_{F,i}$) can act as three orthogonal bases approximately.

In the same way, the robot's desired hand position can be obtained using primitive poses and the mapping vector. The primitive poses of the robot corresponding to the primitive poses of the operator can be designed using the maximum arm length and shoulder length of the robot without difficulty because the three calibration poses fully stretch out the arm. Consequently, the current position of both hands for the operator and the robot can be expressed with mapping vector $\mathbf{w}$ and basis matrix $\mathbf{B}^o$ as follows,

$$\mathbf{p}^o_{ha} = \mathbf{B}^o_{imit}\mathbf{w}, \tag{3}$$

$$\mathbf{p}^o_{ha} = \begin{bmatrix} {}^{sh,l}\mathbf{p}^o_{ha,l} \\ {}^{sh,r}\mathbf{p}^o_{ha,r} \end{bmatrix} \in \mathbb{R}^6, \tag{4}$$

$$\mathbf{B}^o = \begin{bmatrix} \mathbf{B}^o_{pri,l} & \mathbf{O} \\ \mathbf{O} & \mathbf{B}^o_{pri,r} \end{bmatrix} \in \mathbb{R}^{6\times6}, \tag{5}$$

$$\mathbf{B}^o_{pri,i} = \begin{bmatrix} {}^{sh,i}\bar{\mathbf{p}}^o_{S,i} & {}^{sh,i}\bar{\mathbf{p}}^o_{T,i} & {}^{sh,i}\bar{\mathbf{p}}^o_{F,i} \end{bmatrix} \in \mathbb{R}^{3\times3}, \tag{6}$$

$$\mathbf{w} = \begin{bmatrix} w_{1,l} & w_{2,l} & w_{3,l} & w_{1,r} & w_{2,r} & w_{3,r} \end{bmatrix}^T \in \mathbb{R}^6 \tag{7}$$

where right superscript $o \in \{h, r\}$ indicates the object whether the variable is for human or the robot, and $\mathbf{O}$ is the all-zero elements matrix.

Then, there are two steps to map the current hand position of the operator to the robot's target hand motions using (3). First, find a mapping vector, $\mathbf{w}^*$, satisfying the constraint, $\mathbf{p}^h_{ha} = \mathbf{B}^h\mathbf{w}^*$ for the operator. Second, product the mapping vector ($\mathbf{w}^*$) obtained from the first step to the basis matrix of the robot, $\mathbf{p}^r_{ha} = \mathbf{B}^r\mathbf{w}^*$. One of the straightforward method to obtain $\mathbf{w}^*$ in the first step is to use the Moore pseudo-inverse matrix of the basis matrix, $\mathbf{w} = \mathbf{B}^{h+}\mathbf{p}^h_{ha}$. However, using the pseudo-inverse matrix makes it difficult to impose constraints on the mapping vector.

Therefore, the quadratic programming (QP) framework is adopted to implement constraints over the mapping variable and hand velocity. In order to implement velocity limits, (3) is differentiated as,

$$\dot{\mathbf{p}}^h_{ha} = \mathbf{B}^h\dot{\mathbf{w}} \tag{8}$$

because $\mathbf{B}^h$ is constant. Then, $\dot{\mathbf{w}}^*$ can be calculated using (8) under the constraints, and the mapping variable is integrated numerically as $\mathbf{w}_k = \mathbf{w}_{k-1} + \dot{\mathbf{w}}_k^* \Delta t$ where the subscript k means control step in discrete-time domain, and Δt means time interval of the discrete control. But, this method causes integration drift without feedback term. Thus, a feedback term proportional to the hand position error is added to (8) as below.

$$\mathbf{B}^h \dot{\mathbf{w}}_k = \dot{\mathbf{x}}_{ha,d} = \dot{\mathbf{p}}_{ha}^h + \mathbf{K}_{ha}^h(\mathbf{p}_{ha}^h - \mathbf{B}^h \mathbf{w}_{k-1}). \tag{9}$$

The term $\mathbf{K}_{ha}^h \in \mathbb{R}^{6\times 6}$ is a diagonal control gain matrix for the operator's hand, and the $\dot{\mathbf{x}}_{ha,d}$ is the desired velocity vector to follow the current hand position of the operator.

For the constraints, two different constraints are implemented. First, the velocity constraint on the hand of the operator is introduced, $\underline{\dot{\mathbf{x}}}_{ha}^h \leq \mathbf{B}^h \dot{\mathbf{w}} \leq \bar{\dot{\mathbf{x}}}_{ha}^h$, where $\underline{\dot{\mathbf{x}}}_{ha}^h$ and $\bar{\dot{\mathbf{x}}}_{ha}^h$ are lower and upper bound vectors of both hands, respectively. Second, the constraint on the mapping variable value is introduced. Assuming that the three calibration positions are the longest arm position, the constraint is imposed on the mapping parameters, $-\mathbf{1} \leq \mathbf{w} \leq \mathbf{1}$, where $\mathbf{1}$ is the vector whose all elements are 1. However, the constraint on the mapping vector should be implemented on the derivative of the mapping vector for the second constraint. The constraint of the mapping vector is implemented as $\mathbf{K}_w(\underline{\mathbf{w}} - \mathbf{w}_{k-1}) \leq \dot{\mathbf{w}} \leq \mathbf{K}_w(\bar{\mathbf{w}} - \mathbf{w}_{k-1})$, where $\mathbf{K}_w \in \mathbb{R}^{6\times 6}$ is a positive diagonal gain matrix for the convergence at the boundary, and $\underline{\mathbf{w}}$ and $\bar{\mathbf{w}}$ are the lower and upper boundary of the mapping vector respectively.

Finally, a QP problem for the derivative of the mapping variable is formulated as below.

$$\begin{aligned} \min_{\dot{\mathbf{w}}} \quad & \|\mathbf{B}^h \dot{\mathbf{w}} - \dot{\mathbf{x}}_{ha,d}\|^2 \\ \text{s. t.} \quad & \underline{\dot{\mathbf{x}}}_{ha}^h \leq \mathbf{B}^h \dot{\mathbf{w}} \leq \bar{\dot{\mathbf{x}}}_{ha}^h \\ & \mathbf{K}_w(\underline{\mathbf{w}} - \mathbf{w}_{k-1}) \leq \dot{\mathbf{w}} \leq \mathbf{K}_w(\bar{\mathbf{w}} - \mathbf{w}_{k-1}). \end{aligned} \tag{10}$$

D. Orientation Mapping

The desired rotation matrix of the robot's link is derived by mapping the orientation of the operator's body part. Except for the hand position task, all the other teleoperation tasks are orientation tasks. The advantage of orientation mapping compared with position mapping is that orientation mapping does not require a skeleton calibration procedure and can alleviate the restrictions on the mounting position of the tracker.

The orientation mapping can be divided into two methods; Absolute Orientation Mapping (AOM) and Relative Orientation Mapping (ROM). The AOM uses the orientation of the operator itself as the desired orientation of the robot. For the AOM, the local frame of the tracker attached to the target body parts of the operator and the corresponding frame of the robot should be the same:

$$\mathbf{R}_{b,d}^r = \mathbf{R}_b^h. \tag{11}$$

On the other hand, the ROM measures the difference of the current rotation matrix of the operator's tracker from the initial orientation, $\Delta\mathbf{R}_b^h = \mathbf{R}_b^h(\bar{\mathbf{R}}_b^h)^T$, and rotates the corresponding initial orientation of the robot by $\Delta\mathbf{R}_b^h$. In this case, the initial orientation of the robot link, $\bar{\mathbf{R}}_b^r$, corresponding to the initial pose of the operator, $\bar{\mathbf{R}}_b^h$, should be defined in advance. In this paper, the Still-pose is used for the initial pose:

$$\mathbf{R}_{b,d}^r = \Delta\mathbf{R}_b^h \bar{\mathbf{R}}_b^r. \tag{12}$$

TABLE II

PRIORITY OF THE TASKS

Priority	Task	Type	DoF
1 (highest)	Joint Limit, Joint Velocity Limit, Hand Velocity Limit	Inequality	23+12
2	Upper Body ROM	Equality	3
3	Hand Position Control, Hand AOM, Head AOM-AAC	Equality	12+2
4	Upper Arm ROM-AAC	Equality	4
5 (lowest)	Shoulder ROM-AAC	Equality	4

The AOM and the ROM have pros and cons respectively. The AOM does not need to define the initial pose for the robot link before, but the tracker should be mounted correctly so that the local frame should be aligned with the corresponding local frame of the robot link. The ROM has the advantage that the mounting orientation of the tracker can be arbitrary because ROM measures $\Delta\mathbf{R}_b^h$. However, the performance of the ROM relies on the quality of the initial pose calibration. In this paper, the AOM is used for the hand and head orientation mapping because most commercial data gloves have compatible vive-tracker-mount which can fix the orientation of the tracker relative to the hand regardless of the operator, and also the orientation of HMD relative to the head rarely changes regardless of the operator. However, it is difficult to mount the tracker consistently for the chest and the upper arm using the elastic band. Accordingly, ROM is used for the chest and the upper arm.

III. HQP-IK BASED HUMANOID UPPER BODY CONTROL

A. Priority of Tasks

The multi objectives of the robot obtained from the proposed motion mapping method are controlled by imposing a strict hierarchy. The control objectives can be expressed as linear equality or inequality constraints. Specifically, the position and orientation tasks that are obtained from the motion mapping algorithm (Chapter II) can be expressed in the linear equality constraints. Additionally, safety-related constraints (joint position and velocity constraints) can be described in inequality constraints. HQP is introduced to resolve the strict hierarchy on these control constraints.

In TABLE II are the priorities of the control objective, the type of constraints, and the DoF of the task. The inequality constraints on the joint position, velocity, and hand velocity are the most important priority because it is related to safety. The second task is chest orientation, and the third priority is hand position and orientation control. The other third priority is head orientation control for realistic

TABLE III

JACOBIAN MATRICES AND COMMAND VELOCITY OF THE TASKS

Priority(p)	$\mathbf{J}_p$	$\dot{\mathbf{x}}_{p_d}$
2	$\mathbf{J}_{ub_\omega}$	$\omega_{ub,d}$
3	$\begin{bmatrix} \mathbf{J}_{ha_v} \\ \mathbf{J}_{ha_\omega} \\ \mathbf{S}_x\mathbf{R}_{he}^T\mathbf{J}_{he_\omega} \end{bmatrix}$	$\begin{bmatrix} \dot{\mathbf{p}}_{ha,d}^r + \mathbf{K}_{ha}^r(\mathbf{p}_{ha,d}^r - \mathbf{p}_{ha}^r) \\ \omega_{ha,d} \\ \mathbf{S}_x\mathbf{R}_{he}^T\omega_{he,d} \end{bmatrix}$
4	$\mathbf{S}_x\mathbf{R}_{ua}^T\mathbf{J}_{ua_\omega}$	$\mathbf{S}_x\mathbf{R}_{ua}^T\omega_{ua,d}$
5	$\mathbf{S}_x\mathbf{R}_{sh}^T\mathbf{J}_{sh_\omega}$	$\mathbf{S}_x\mathbf{R}_{sh}^T\omega_{sh,d}$

visual feedback through HMD. The fourth and fifth tasks are about the orientation of the upper arm and shoulder. The upper arm and shoulder do not affect the performance of the manipulation but these impact the visual similarity and can handle the redundancy of the humanoid robot arm. The head, upper arm, and shoulder are not controlled in three-dimensional orientation but only one of the axis of the local frame is controlled by axis alignment control (AAC). The axis alignment control has two degrees of freedom which is less than the DoF of the three-dimensional orientation control. The details about the axis alignment control are discussed in the following Section III-B.

B. Axis Alignment Control

The axis alignment control is the method to make one axis parallel to the target axis. There are two reasons to use AAC rather than three-dimensional orientation control. The ratio of the upper arm length to the lower arm length is different between the robot and the operator. Therefore, it is difficult for the robot to imitate the operator's arm pose perfectly even if the orientation of the upper arm is matched. Then, it is found that aligning the arm-plane is more effective for imitation rather than matching all three-dimension orientation of the upper arm. The arm-plane consists of three points as shown in Fig.4; hand, elbow, shoulder. To make the arm-plane of the robot parallel to the operator's arm-plane, the normal vector of the arm-plane is controlled by AAC based on the fact that controlling the plane is equivalent to controlling its normal vector. The second reason to use AAC is that the AAC requires two DoF while the orientation control requires three DoF. For the robot's head, there are only two neck joints without a roll joint. The shoulder is in the same condition. If the orientation of the head or the shoulder is controlled, the task is not fully controllable. Even though the resultant motion of the robot is the same whether the ACC is used or not for the head and the shoulder, it is desirable to decrease the dimension of the equality constraints for efficient computation of HQP. Therefore, the DoF of the task is reduced by using AAC.

The AAC equation can be derived using the selection matrix from the general orientation control as

$$\mathbf{S}_x\mathbf{R}_b^T\mathbf{J}_{b_\omega}\dot{\mathbf{q}} = \mathbf{S}_x\mathbf{R}_b^T\omega_{b,d}, \ \mathbf{S}_x = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{13}$$

where $\mathbf{S}_x$ is the specific selection matrix for the x-axis AAC, $\mathbf{R}_b$ is the rotation matrix of the target link, $\mathbf{J}_{b_\omega}$ is the Jacobian matrix of target link for the angular velocity, and $\omega_{b,d}$ is the desired angular velocity of target link for the three-dimensional orientation control. $\omega_{b,d}$ can be calculated as $\omega_{b,d} = \mathbf{K}_\omega^r \cdot \Phi(\mathbf{R}_{b,d}^r, \mathbf{R}_b^r)$ where $\mathbf{K}_\omega^r$ is the diagonal gain matrix, and $\Phi(\cdot)$ is the angle-axis function that calculates angular velocity from the difference of two current and desired rotation matrices. The $\mathbf{S}_x$ ignores the first row of the orientation control in the local frame so that AAC for the x-axis does not control the x-axis but just align the x-axis using the y and z components of the orientation control.

C. HQP-IK formulation

In this section, all the tasks described in TABLE II are solved as the inverse kinematics problem. In order to control the tasks according to their priorities, the IK problem is also formulated using HQP. Because the first priority only has inequality constraints, the HQP-IK problem can be expressed as follows:

$$\begin{aligned} \min_{\dot{\mathbf{q}}} \quad & \rho_p\|\mathbf{J}_p\dot{\mathbf{q}} - \dot{\mathbf{x}}_{p,d}\|^2 + \|\dot{\mathbf{q}}\|_{\mathbf{A}}^2 \\ \text{s. t.} \quad & \underline{\dot{\mathbf{x}}}_{ha}^r \leq \mathbf{J}_{ha}\dot{\mathbf{q}} \leq \bar{\dot{\mathbf{x}}}_{ha}^r \\ & \mathbf{K}_q(\underline{\mathbf{q}} - \mathbf{q}_{k-1}) \leq \dot{\mathbf{q}} \leq \mathbf{K}_q(\bar{\mathbf{q}} - \mathbf{q}_{k-1}) \\ & \underline{\dot{\mathbf{q}}} \leq \dot{\mathbf{q}} \leq \bar{\dot{\mathbf{q}}} \\ & \mathbf{J}_n\dot{\mathbf{q}} = \mathbf{J}_n\dot{\mathbf{q}}_{p-1}^*, \ \forall n \in 2, ..., p-1, \ (p \geq 3) \end{aligned} \tag{14}$$

where p is the priority of the constraints ($p \geq 2$), $\mathbf{J}_p$ is the task Jacobian matrix of p-th priority, $\dot{\mathbf{x}}_{p,d}$ is the desired velocity vector of p-th priority, ρ_p is the weighting value for the control error, $\mathbf{A}$ is the inertia matrix of the robot, $\mathbf{J}_{ha}$ is the Jacobian matrix of the robot hand, $\underline{\dot{\mathbf{x}}}_{ha}^r$ and $\bar{\dot{\mathbf{x}}}_{ha}^r$ are the lower and upper bound of the velocity of the robot hand, $\underline{\mathbf{q}}$ and $\bar{\mathbf{q}}$ are the lower and upper bound of the joint position, $\underline{\dot{\mathbf{q}}}$ and $\bar{\dot{\mathbf{q}}}$ are the lower and upper bound of the joint velocity, $\mathbf{J}_n$ is the Jacobian matrix of the higher priority than p, and $\dot{\mathbf{q}}_{p-1}^*$ is the optimal solution obtained from the previous iteration. The Jacobian matrices and the command velocities of all the equality constraints are summarized in the TABLE III.

HQP, (14), is calculated iteratively by increasing p from 2 to 5. The last equality constraint is activated when $p \geq 3$ to ensure that the higher priority task is not affected by the lower priority task. The final optimal solution is usually obtained at $p = 4$ generating infeasible error from the QP solver at $p = 5$ because DoF of the arm (8 DoF) is fully used for hand pose (6 DoF) and upper arm AAC (2 DoF) resulting in that the rank of the null space for the shoulder task is zero. However, when the robot arm goes near the singularity poses, for instance, the T-Pose, the null space for the shoulder is not empty and the lowest shoulder task can regulate the shoulder orientation. Otherwise, the shoulder could be flipped backward and stuck at the joint limit.

The cost function consists of two terms; the control error term, $\|\mathbf{J}_p\dot{\mathbf{q}} - \dot{\mathbf{x}}_{p,d}\|^2$, and the regularization term, $\|\dot{\mathbf{q}}\|_{\mathbf{A}}^2$. The control error term is an L2-norm of the velocity error weighted by ρ_p. Also, the regularization term is an L2-norm of the joint velocity weighted by the inertia matrix of the robot, $\mathbf{A}$. The regularization term is added for the smooth and low-energy motion but it results in the deteriorated control performance. It is decided to use the regularization term for

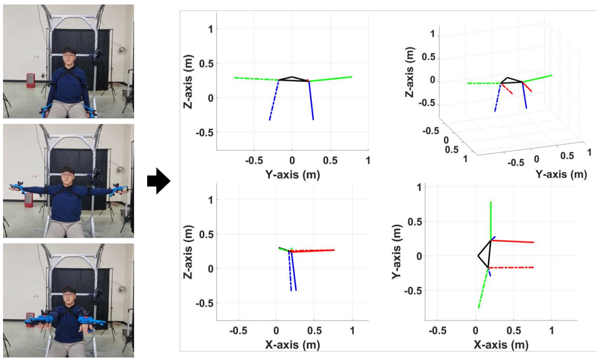


Fig. 5. 3D visualization of three calibration poses and the estimated human model.

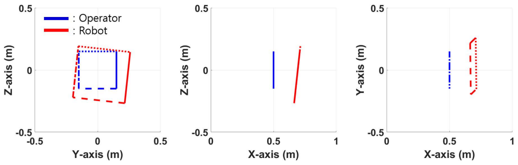


Fig. 6. Square trajectory mapping result using the proposed hand position mapping.

TABLE IV

CIRCLE INDEX FOR EACH SUBJECTS AND ITS VARIANCE AFTER THE PROPOSED HAND POSITION MAPPING

Circle index		**Subjects**			
		A	**B**	**C**	**D**
Left Hand	Original	0.1533	0.1708	0.1256	0.1403
	Retargeted	0.1466 (-4.5%)	0.1613 (-5.9%)	0.1194 (-5.2%)	0.1734 (+19.1%)
Right Hand	Original	0.1260	0.2689	0.1659	0.1285
	Retargeted	0.1571 (+19.8%)	0.2471 (-8.8%)	0.1549 (-7.1%)	0.1135 (-13.2%)

the stable teleoperation although the tracking performance decreases because generating a smooth trajectory is more important than its tracking performance, and the small tracking error can be adapted by the operator using visual feedback.

After obtaining the optimal solution, $\dot{\mathbf{q}}^*$ from above HQP, it is integrated to the desired joint position. Finally, the desired torque is calculated using the joint PD (proportional–derivative) control method with gravity torque compensation.

IV. Experiments

Three experiments are conducted to validate the proposed teleoperation framework in the simulation and the real robot. In the first experiment, the hand position mapping method is analyzed about its distortion of the 3D motion of the operator and the effect on the desired behavior. Second, a box carrying scenario, one of the dual-arm coordination tasks, is performed using the real robot to show the telemanipulation performance of the proposed method. Third, motion similarity between the operator and the robot is compared between the proposed mapping method and the other methods. Two linear mapping methods, the one-to-one position mapping and the arm length proportional mapping which are mainly used for teleoperation in the robotics field, are compared with the proposed hand position mapping in the MuJoCo physics simulation.

A. *Pose Calibration and Hand Position Mapping Evaluation*

In this section, the pose calibration and hand position mapping method are validated to show how the hand position mapping transforms the motion of the operator. Although the pose calibration result is different according to the individuals, one of the author's calibration result is demonstrated as an example. The calibration result is displayed in Fig.5. The estimated left and right arm lengths of the operator are 0.5688m and 0.5995m respectively. The angles between each calibration poses (Still Pose - T Pose, T Pose - Forward Pose, Forward Pose - Still Pose) are estimated as [90.7 92.51 86.96]° and [82.24 102.49 88.09]° for left and right arm respectively.

Using this calibration result, an ideal square trajectory is retargeted to the robot's desired hand trajectory through the proposed hand position mapping. As shown in Fig.6, the blue square trajectory is transformed to the red rectangle according to the calibration basis vectors. This distortion of the trajectory can be misunderstood that the proposed method would result in undesired motion of the robot which is different from the intended motion of the operator. However, a human can not draw the perfect square because they have an inherent sense of space different from the actual three-dimensional space. So, it is hard to determine how the proposed hand position mapping distorts the desired motion of the operator from this ideal trajectory mapping.

Accordingly, user studies are conducted to measure the effect of the proposed algorithm on the operator's intended motion. Four subjects were asked to draw 10 perfect same circles for each hand with a diameter of 30 cm and with a constant speed in the front (y-z plane) after the pose calibration. The subjects conducted the experiment with their eyes closed to exclude the effect of the visual feedback, and only to test the sense of the space and the effect of the hand position mapping method on it. The hand trajectory of the subject is recorded and mapped to the robot to compare which trajectory of the circles is more circle-like. For a fair quantitative comparison, a dimensionless index was introduced.

To calculate the circle index, each circle trajectories of the human and the robot are regressed to a circle in the y-z plane. The center of the circle is found by averaging all the trajectories. The radius of the circle is calculated by projecting all trajectories to the frontal plane (y-z plane) and averaging all distances from the center of the circle to all projected trajectories. Lastly, the circle index is obtained as an RMS error of the circle regression divided by the radius of the circle which indicates that the smaller the index value, the closer the given trajectory is to a circle.

The result of the four subjects is summarized in the TABLE IV. Surprisingly, among the 8 cases, 6 cases show that the retargeted trajectory of the circle is closer to the circle than the original trajectory. On the other hand, in only two cases

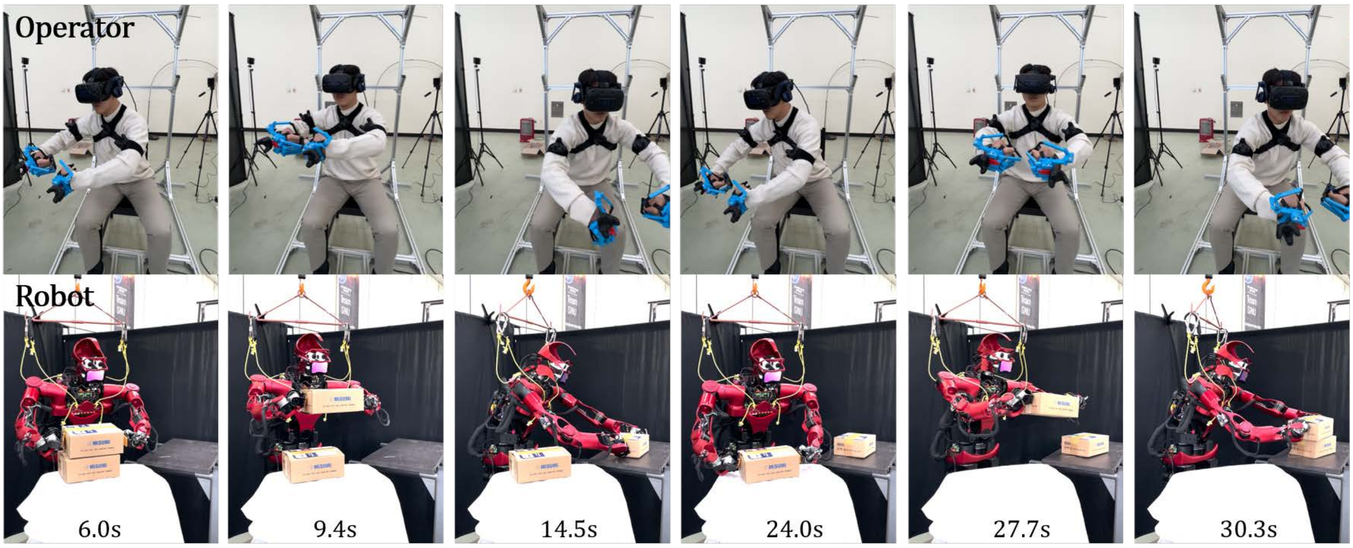


Fig. 7. Snapshots of dual-arm coordination task (box carrying) using the proposed telemanipulation framework.

(A's right hand, D's left hand), the original trajectory was closer to a circle than the trajectory mapped to the robot. Even though only four people are tested in this experiment, this result can be interpreted that a person has their own distorted sense of space and this can be canceled out by using the calibration pose space which is also affected by the subject's sense of space. Therefore, the proposed calibration-based mapping method can extract the intention of the operator in the mapping variable, **w**, and retarget the intended motion more correctly than the one-to-one task space mapping and the arm length proportional mapping method because both have the same circle index value with the original motion.

B. Dual-Arm Coordination Task using the Proposed Method

To demonstrate the telemanipulation capability of the proposed method, a simple box transportation scenario is demonstrated. Although task-oriented tools or general robot hands can be attached to the end-effector of the robot arm, dual-arm manipulation is conducted without tools to show the dual-arm coordination performance of the proposed mapping algorithm. The legs of the robot are controlled to maintain the center of mass of the robot at the center of the support polygon and the upper body is controlled by the telemanipulation framework proposed in this paper. The operator wears the HMD on the head and obtains visual information of the remote space using the stereo cameras on the robot head. The operator also wears the blue haptic gloves only to fix the trackers pose on the hands. As shown in Fig.7, the operator was enabled to move two boxes successfully. Note that the operator can manipulate the box remotely only with visual feedback. For the full video see `https://youtu.be/zkADn9E_Z4g`.

C. Comparison on Visual Similarity According to the Hand Position Mapping Algorithm

In this section, three hand position mapping methods are compared with respect to the visual similarity between the source motion of the human and the retargeted motion of the robot. The arm length proportional mapping and the one-to-one hand position mapping are chosen for the comparison because the two methods are the most widely used methods in robotics. Except for the hand position mapping, the orientation mapping and the HQP-IK controller are applied equally. The full video is available in this link: `https://youtu.be/mzJUSlp3d4Y`

In Fig.8, eight different motion clips of the operator and three hand position mapping methods are displayed. The motions of the three methods in the fourth and the sixth columns appear similarly when the hand of the operator is near the upper body in which the origin of the coordinate lies. But, when the hand of the operator stretches out from the body, the appearance of the robot in the one-to-one mapping method (third row) is different from the operator because the arm lengths between the operator (0.58m) and the robot (0.78m) are different. Therefore, the one-to-one mapping can not uses the entire workspace of the robot effectively. The arm length proportional mapping method complements the shortcoming of the one-to-one position mapping and imitates the motion of the operator more similarly. However, for the T pose and similar poses as shown in the second and the last columns of Fig.8, the proportional mapping method also can not imitate the motion of the operator well because the shoulder lengths are different. On the other hand, the proposed method is almost superior in appearance similarity except for the clap motion in the fourth column. The proposed hand position mapping uses different coordinate frames for each hand by locating the origin at the center of the shoulder of each arm. Thus, the proposed method can mimic the T pose better but makes the gap between hands in the clap pose also. In the seventh column of Fig.8, the arm-plane of the robot can follow the operator's using the AAC of the upper arm. But, in the other cases, the arm-plane of the robot is not parallel to the operator's due to the joint limit.

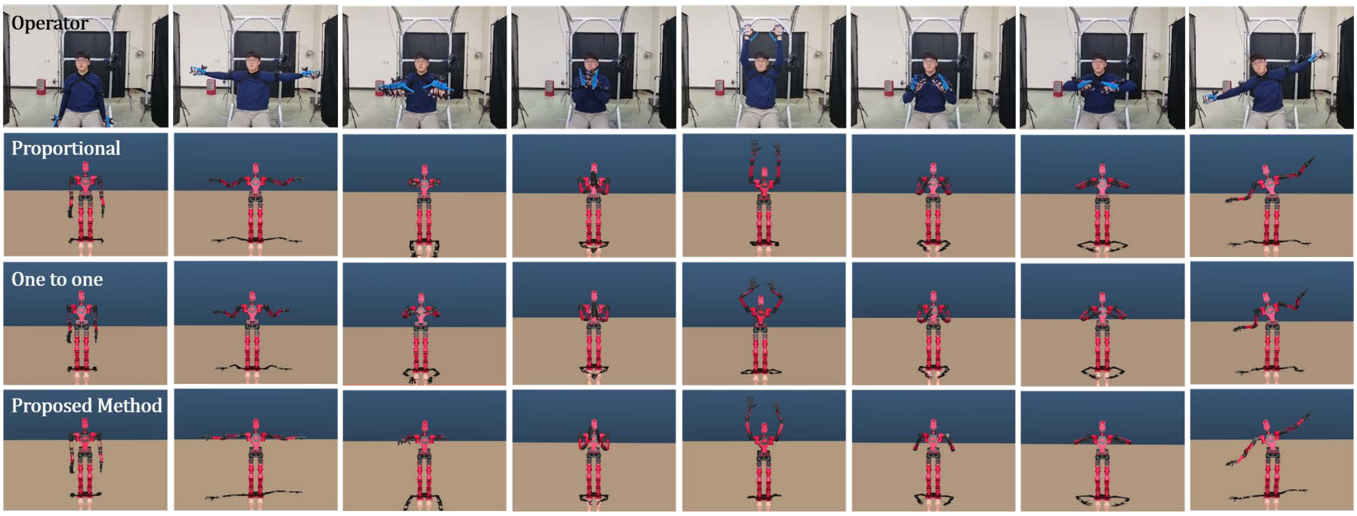


Fig. 8. Comparison of motion appearance similarity between the operator and the robot according to the hand position mapping method.

V. Discussion and Conclusion

In this paper, we propose the motion retargeting method and control framework for teleoperation. The motion retargeting algorithm consists of two mapping methods, the hand position mapping, and orientation mapping. The proposed hand position mapping utilizes the individual's calibration poses as a basis vector to express the current position of the operator's hand, which removes the distorted sense of space each person has and allows the robot to move as desired by the operator. The experiments show the benefits of the proposed mapping method over the two baseline mapping methods on the desired motion generation and appearance similarity. Also, the dual-arm telemanipulation task, moving two boxes, is performed successfully using the proposed method only with visual feedback.

However, several issues remain as future works. First, more user studies and theoretical analysis are needed to verify the hypothesis that the proposed hand position mapping cancels out the distorted sense of space of individuals. Second, it is necessary to resolve the problem of a gap between the robot's two hands when the operator holds their hands together. It can be resolved by switching the mapping algorithm according to the intention of the operator. But, estimating the intention of the operator is difficult and the operator feels strange when the mapping method changes during teleoperation. Lastly, measuring the effect of the retargeting algorithm on the performance of the remote control of the robot is one of the challenging problems. The performance of the teleoperation is affected by not only the retargeting algorithm but also other components, for instance, visual feedback, haptic feedback, acoustic feedback, past experience, personal differences in the motion sickness of the VR system, etc. Therefore, it is important to design the experimental environment and set the appropriate control variable to analyze the effect on the telemanipulation performance.

References

[1] R. Villegas, J. Yang, D. Ceylan, and H. Lee, "Neural kinematic networks for unsupervised motion retargetting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8639–8648.

[2] K. Aberman, P. Li, D. Lischinski, O. Sorkine-Hornung, D. Cohen-Or, and B. Chen, "Skeleton-aware networks for deep motion retargeting," *ACM Transactions on Graphics (TOG)*, vol. 39, no. 4, pp. 62–1, 2020.

[3] S. Choi and J. Kim, "Towards a natural motion generator: A pipeline to control a humanoid based on motion data," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 4373–4380.

[4] S. Choi, M. Pan, and J. Kim, "Nonparametric motion retargeting for humanoid robots on shared latent space," *Proceedings of Robotics: Science and Systems (R: SS) 2020*, 2020.

[5] S. Choi, M. J. Song, H. Ahn, and J. Kim, "Self-supervised motion retargeting with safety guarantee," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 8097–8103.

[6] C. Lenz and S. Behnke, "Bimanual telemanipulation with force and haptic feedback and predictive limit avoidance," in *2021 European Conference on Mobile Robots (ECMR)*. IEEE, 2021, pp. 1–7.

[7] J. Oh, O. Sim, H. Jeong, and J.-H. Oh, "Humanoid whole-body remote-control framework with delayed reference generator for imitating human motion," *Mechatronics*, vol. 62, p. 102253, 2019.

[8] F. Abi-Farrajl, B. Henze, A. Werner, M. Panzirsch, C. Ott, and M. A. Roa, "Humanoid teleoperation using task-relevant haptic feedback," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 5010–5017.

[9] J. Koenemann, F. Burget, and M. Bennewitz, "Real-time imitation of human whole-body motions by humanoids," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 2806–2812.

[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] L. Fritsche, F. Unverzag, J. Peters, and R. Calandra, "First-person teleoperation of a humanoid robot," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2015, pp. 997–1002.

[12] R. Elbasiony and W. Gomaa, "Humanoids skill learning based on real-time human motion imitation using kinect," *Intelligent Service Robotics*, vol. 11, no. 2, pp. 149–169, 2018.

[13] S. Wang, K. Murphy, D. Kenney, and J. Ramos, "A comparison between joint space and task space mappings for dynamic teleoperation of an anthropomorphic robotic arm in reaction tests," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 2846–2852.