Asymmetric Bilateral Telerobotic System with Shared Autonomy Control

Da Sun Qianfang Liao and Amy Loutfi

***Abstract*—The asymmetry in bilateral teleoperation, i.e., the differences of mechanical structures, sizes and number of joints between the master and slave robots, can introduce kinematics redundancy and workspace inequality problems. In this paper, a novel shared autonomy control strategy is proposed for handling the asymmetry of bilateral teleoperation, which has two main contributions. First, to deal with kinematics redundancy, the proposed strategy provides an self-regulation algorithm of orientation that allows the operator to solely use master position command to simultaneously control the slaves position and orientation. Second, to deal with workspace inequality, the proposed strategy enables the slave's workspace to be dynamically tunable to adapt to various task spaces without influencing the smoothness of the robots movement. The experiments on a platform consisting of a 6-Degree of Freedom (DoF) UR10 robot and a 3-DoF haptic device are given to validates the effectiveness of the proposed control strategy.**

***Index Terms*—Asymmetric bilateral teleoperation, Shared autonomy, Orientation regulation, Human-machine interaction, Workspace mapping**

I. Introduction

A. Background

WE are entering a new era of human-robot shared work, where robots aid in surpassing human physical limitations and providing necessary assistance. Completely autonomous robot systems usually require good sensory mechanism for goal identification [1], [2], long training process [3], [4], and high-level dexterity [5], [6], which usually have limited performance in cluttered environment. Teleoperation in combination with human intelligence can, on the other hand, deliver a safe, reliable and robust performance. As an extension of teleoperation, bilateral teleoperation denotes that the master haptic device is manipulated by an operator to control the remote slave robot, and meanwhile, the master receives information from the slave robot to enhance the operator's perception about the remote environment. Currently, the majority of existing studies of bilateral teleoperation focus on time delay based stability [7]–[10], motion synchronization [11]–[13], force reflection [14]–[16], and system modelling and uncertainties compensation [17], [18]. When directly applying the above approaches to an Asymmetric Bilateral Teleoperation (ABT), the problem, however, arises that a great workload is placed on the human operator [19].

An ABT is generally defined as that the master and slave robots involved in the teleoperation control loop are with different mechanical structures, joints and sizes. The different master-slave structures impel the operator to put extra efforts on mapping the movements between the master and the slave. For position mapping, it is an intuitive and relatively easy task for the operator to achieve. For orientation mapping, however, it is a much more difficult task for the operator in many industrial or surgical applications [20]. In the existing studies of the teleoperation with multiple DoF, generally, a joint-to-joint mapping is established by supposing that the master and the slave have same structures [14], [21], [22], or the kinematics redundancy is ignored and no orientation regulation is given [5], [23], [24], or the regulation cannot support full range of orientation control due to the limitation of the master's mechanical structure [25], [26]. Even if a good orientation mapping is available, it will be an exhausting job for an operator to simultaneously regulate position and orientation of a slave robot, especially in the situation of multiple robots control and large robot structures difference [27]. Therefore, we are motivated to develop a new shared autonomy strategy that the robot's orientation can be self-regulated and then the operator only needs to control the robot's position, which can reduce the operator's burden.

Another issue affecting an ABT system is the workspace inequality caused by the different sizes of the robots. For this issue, the existing studies generally utilize scaling control [28], such as the methods applied in micro-macro tele-surgical application where the master is utilized to control a surgical robot in a small scale [29]–[32], and the methods amplifying the master control signals via large scaling gains to control a slave robot with a larger size [33]–[36]. The above methods use constant scaling gains to adjust their commands' amplitude such that the slave robot can conduct motions in the desired workspaces. However, for the system where the slave robot is of a larger size, using large constant scaling gains in ABT control will cause two main problems. First, it is not easy to find a proper value of the gain that can exactly match the workspace in a certain application. Second, with a too large scaling gain, a small movement of the master will lead to an overlarge movement of the slave, which enhances the difficulty of teleoperation. In some systems, extra equipment such as foot pedal is utilized to dynamically tune the scaling gains [37], which however, require the operator to carefully regulate the extra equipment in different cases and may lead to safety problems. It is because directly tuning the scaling gains can easily introduce jerky movement of the slave to cause damage and fail the task. This fact motivates us to develop a new approach that can dynamically tune the workspace and meanwhile guarantee the movements smoothness and safety.

* This paper is funded by AI.MEE program: AutoDIVE.

The authors are with the Center for Applied Autonomous Sensor Systems, Örebro University (cooresponding author: Qianfang Liao), Sweden. Email: Da.Sun@oru.se, Qianfang.Liao@oru.se, Amy.Loutfi@oru.se

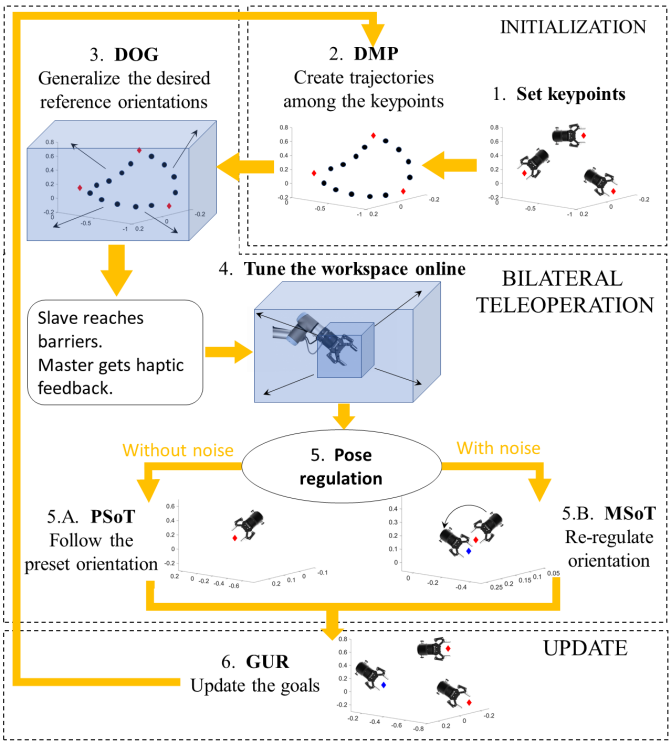


Fig. 1. Diagram of the overall strategies

In this paper, a novel shared autonomy control strategy is proposed for ABT. The main contributions are as follows. First, we provides a method to self-regulate the robot's orientation such that the operator can solely use the master's position command to control both of the position and orientation of the slave in the presence of kinematics redundancy. Second, we provides a workspace tuning approach that allows the slaves workspace to be adaptively updated to match the practical task space and simultaneously guarantee the smoothness of the robots motion and safety. In the following section, a brief description of the proposed strategy is provided.

B. Brief description

The brief diagram of the proposed strategy is shown in Fig. 1. The overall strategy is composed of six steps, where Step 4 is the workspace tuning approach, and the other steps constitute the orientation regulation approach.

Step 1: Set keypoints. The actual workspace of the slave robot can be divided into several sub-areas according to different task requirements. In each sub-area, we set one keypoint which is a vector containing the desired location and orientation of the robot at this area. Note that these keypoints can be roughly determined, and they are used as the initialization of the following steps, and can be further revised during teleoperation.

Step 2: Dynamic Movement Primitives (DMP). After the keypoints are set, the DMP [38] is automatically launched to create trajectories between every two keypoints with desired trajectory shapes. The trajectories support the robot to have smooth position and orientation transformation from one keypoint to another.

Step 3: Desired Orientation Generalization (DOG). Unlike the autonomous system in [38] which only moves along the trajectories created by DMP, an ABT system should allow the operator to drive the slave to move to everywhere within the practical workspace, which means, the slave robot can move out of the trajectories created by DMP. Accordingly, we propose the DOG algorithm that generalizes the orientations of the points of the trajectories to the overall workspace of the slave. The DOG algorithm is automatically launched after Step 2. It updates the desired orientation in each sampling time, and allows the slave robot to have a reasonable orientation in arbitrary location within the workspace during teleoperation.

Step 4: Tune the workspace online. At the beginning of teleoperation, we let the slave's workspace be a small area which has the same size as the master's workspace. Then, a workspace tuning approach is leveraged to adaptively expand the slave's workspace and enlarge the scaling gains. With this approach, the robot's workspace can dynamically match the practical task space, and the robot's movement can cover all the keypoints to perform the tasks. This approach also guarantee that the robot can move smoothly without jerky motions during the workspace tuning.

Step 5: Pose regulation. The keypoints of the task are roughly preset in *Step 1*, and they may not be fit for the changing environment. As a result, when the slave robot reaches the area of a keypoint, its pose may not be optimal to perform the task. We define this situation as "pose noise". Accordingly, we define two different cases of this step marked as *Step 5.A* and *Step 5.B*. For *5.A* that pose noise does not exist and the preset keypoint is satisfactory, we propose an algorithm called Primitive Stack of Tasks (PSoT) which automatically runs to allow the robot to directly perform the task. For *5.B* that pose noise exists and the preset keypoint is not satisfactory, we propose an algorithm called Motion-regulated Stack of Tasks (MSoT) that is launched by the operator for further regulation. MSoT allows a master with lower DoF to regulate the position and orientation of a slave robot with higher DoF using only position commands, which solves the kinematics redundancy problem.

Step 6: Goal Update Rule (GUR). For each area of a keypoint, after the keypoint is modified, we use a method called GUR, which is automatically launched, to evaluate the quality of the modified keypoint and update its value in the record such that the robot can perform tasks in a better pose in the next round.

The remainder of this paper is organized as follows. Section II describes a series of pose regulation and updating algorithms including DMP, DOG, PSoT, MSoT and GUR. The master-slave control laws are presented in Section III, which also involves the workspace tuning approach. The experiment results are demonstrated in Section IV and some conclusions are presented in Section V. The overall system's stability is proved in Appendix.

II. Slave Control Strategies with Autonomous Orientation Regulation

Some key terms are pre-defined in this section. the slave robot pose $X_s \in R^7$ is $X_s = [{}^tX_s{}^T, {}^oX_s{}^T]^T$, where

${}^tX_s = [{}^tX_{sx}, {}^tX_{sy}, {}^tX_{sz}]^T$ denotes the slave robot's position in Catesian space, and ${}^oX_s = [{}^oX_{sx}, {}^oX_{sy}, {}^oX_{sz}, {}^oX_{sw}]^T$ denotes the slave robot's orientation in quaternion. The position and orientation are derived via the robot's kinematics. The reference orientation ${}^oX_r \in R^7$, denoted by ${}^oX_r = [{}^oX_{rx}, {}^oX_{ry}, {}^oX_{rz}, {}^oX_{rw}]^T$, is the output of DOG. The pose trajectory $X_{dr} \in R^{7\times n}$, which contains n points created by DMP, is $X_{dr} = [{}^tX_{dr}{}^T, {}^oX_{dr}{}^T]^T$ with the position ${}^tX_{dr} = [{}^tX_{drx}, {}^tX_{dry}, {}^tX_{drz}]^T$ and the orientation ${}^oX_{dr} = [{}^oX_{drx}, {}^oX_{dry}, {}^oX_{drz}, {}^oX_{drw}]^T$. The master robot position ${}^tX_m \in R^3$ is ${}^tX_m = [{}^tX_{mx}, {}^tX_{my}, {}^tX_{mz}]^T$.

After setting keypoints in *Step 1*, the model-free trajectory generation algorithm, DMP [38], is applied to build a trajectory between every two keypoints as shown in Algorithm 1.

Algorithm 1 Dynamic Movement Primitives (DMP)

$$\ddot{x}_t = \kappa_t(\alpha_t(\beta_t(g_t - x_t) - \dot{x}_t) + \upsilon_t^T\theta_t) \tag{1}$$

$$\upsilon_{t,j} = \frac{\omega_{t,j}s_t}{\sum_{k=1}^{p} w_{t,k}}(g_t - x_0) \tag{2}$$

$$\omega_{t,j} = e^{(-\frac{1}{2h_{t,j}}(s_t - c_{t,j})^2)} \tag{3}$$

$$\dot{s}_t = -\kappa_t(\alpha_t s_t) \tag{4}$$

Eq. (1) is the transformation system that generates the trajectory x_t, where x_t in this paper denotes the pose trajectory X_{dr} produced by DMP. g_t denotes the goal of the trajectory, where g_t in this paper represents the desired pose in the keypoints. The last term $\upsilon_t^T\theta_t$ of (1) determines the shape of the trajectory, where υ_t is a basis function defined in (2) and θ_t is a parameter vector. κ_t, α_t and β_t are positive gains. in (2), (3), (4), $\upsilon_{t,j}$ is the j-th element of υ_t, in which $\omega_{t,j}$ is the Gaussian kernel with its center $c_{t,j}$ and width $h_{t,j}$.

Each of the built trajectories consists of a certain number of points, where a point is a vector including position and orientation. We propose DOG algorithm as shown in Algorithm 2 to generalize the orientations of the points to the overall workspace. As a result, the slave robot will have a reasonable orientation at arbitrary positions of its workspace.

The detail of the proposed DOG algorithm is descibed as follows. When the slave robot is at an arbitrary location ${}^tX_s$, first, multiply ${}^tX_{si}$ by a unit vector $\mathbb{I} \in R^n$. Then, from $|{}^tX_{si}\mathbb{I} - {}^tX_{ri}|$, we can get the error vectors of x, y, z directions in Cartesian space. Based on (5), in each of x, y, z directions, the minimum error e_{si} and their related order ${}^1o_i$ can be derived, as well as the order of the second minimum error ${}^2o_i$. Normally, ${}^1o_i$ neighbours to ${}^2o_i$. Then, from the orders ${}^1o_i$ and ${}^2o_i$ in the trajectory created by DMP, we can get two points. The first point with order ${}^1o_i$ has its position ${}^{t1}X_{dri}$ and orientation ${}^{o1}X_{dri}$, and the second point with order ${}^2o_i$ has its position ${}^{t2}X_{dri}$ and orientation ${}^{o2}X_{dri}$. Equations (6) and (7) determine that when the robot's current position ${}^tX_{si}$ is in the interval between ${}^{t1}X_{dri}$ and ${}^{t2}X_{dri}$, an orientation ${}^{oi}X_r$ can be derived by the variable gains δ_{d1} and δ_{d2}, which can smoothly vary inside the interval between ${}^{o1}X_{dri}$ and ${}^{o2}X_{dri}$. Therefore, we can get a reference orientation in continuous

Algorithm 2 Desired Orientation Generalization (DOG)

1. Determine the minimum error ${}^1e_{si}$ and its order ${}^1o_i$, and the order of the second minimum error ${}^2o_i$ inside the vectors $|{}^tX_{sx}\mathbb{I} - {}^tX_{drx}|$, $|{}^tX_{sy}\mathbb{I} - {}^tX_{dry}|$ and $|{}^tX_{sz}\mathbb{I} - {}^tX_{drz}|$

$$[{}^1e_{si}, {}^1o_i, {}^2o_i] = \min(|{}^tX_{si}\mathbb{I} - {}^tX_{dri}|), \quad i = x, y, z \tag{5}$$

2. From the position ${}^{t1}X_{dri}$ in the order ${}^1o_i$ and the position ${}^{t2}X_{dri}$ in the order ${}^2o_i$, create the variable gains δ_{d1} and δ_{d2}, where $0 \le \delta_{d1,2} \le 1$.

$$\begin{cases} \delta_{d1} = \frac{{}^tX_{si}}{{}^{t1}X_{dri} - {}^{t2}X_{dri}} - \frac{{}^{t2}X_{dri}}{{}^{t1}X_{dri} - {}^{t2}X_{dri}} \\ \delta_{d2} = \frac{{}^tX_{si}}{{}^{t2}X_{dri} - {}^{t1}X_{dri}} - \frac{{}^{t1}X_{dri}}{{}^{t2}X_{dri} - {}^{t1}X_{dri}} \end{cases} \tag{6}$$

3. Based on the orientation ${}^{o1}X_{dri}$ in the order ${}^1o_i$ and the orientation ${}^{o2}X_{dri}$ in the order ${}^2o_i$, and the variable gains δ_{d1} and δ_{d2}, derive

$${}^{oi}X_r = \delta_{d1}{}^{o1}X_{dri} + \delta_{d2}{}^{o2}X_{dri} \tag{7}$$

4. Calculate the probabilities $\mathbb{P}_i$

$$\mathbb{P}_i = \frac{\kappa_{si}e^{-\frac{1}{w_i}|{}^1e_{si}|}}{\sum_{l=x}^{z}\kappa_{sl}e^{-\frac{1}{w_l}|{}^1e_{sl}|}} \tag{8}$$

5. Achieve the reference orientation ${}^oX_r$

$${}^oX_r = \mathbb{P}_x{}^{ox}X_r + \mathbb{P}_y{}^{oy}X_r + \mathbb{P}_z{}^{oz}X_r \tag{9}$$

6. Set a range for each keypoint to fix the orientation.

$${}^oX_r = \begin{cases} {}^{ok}X_r & \text{if } \max(\begin{bmatrix} \min(|{}^tX_{sx}\mathbb{I} - {}^tX_{gx}|) \\ \min(|{}^tX_{sy}\mathbb{I} - {}^tX_{gy}|) \\ \min(|{}^tX_{sz}\mathbb{I} - {}^tX_{gz}|) \end{bmatrix}) \le {}^kr \\ {}^oX_r & \text{if else} \end{cases} \tag{10}$$

state and will not cause sudden jump. By calculating the weight function $\kappa_{si}e^{-\frac{1}{w_i}e_{si}}$, where κ_{si} and w_i are positive gains used to regulate the values of the weight function, the possibility $\mathbb{P}_i$ can be calculated in (8). Then, the final reference orientation ${}^oX_r$ is decided by the orientation ${}^{oi}X_r$ with the highest probability. When the slave robot approaches a keypoint, its orientation is better to be fixed rather than varying such that it is easier for the operator to drive the slave robot to perform a task. According to (10), we define a small positive parameter ${}^kr$ determining the radius of the range of a keypoint. ${}^tX_g = [{}^tX_{gx}, {}^tX_{gy}, {}^tX_{gz}]$ is a matrix aligning all the positions of the keypoints. When the current slave position ${}^tX_s$ is at or near one element of ${}^tX_g$ (inside the range ${}^kr$), the reference orientation will be ${}^{ok}X_r$, where ${}^{ok}X_r$ is the orientation of a keypoint. If ${}^tX_s$ is outside the range, ${}^oX_r$ is still the orientation created by DOG. The small positive parameter ${}^kr$ will not introduce large sudden jump between ${}^{ok}X_r$ and ${}^oX_r$.

By employing DOG, the slave robot's position ${}^tX_s$ and orientation ${}^oX_s$ track the master robot's position ${}^tX_m$, and the reference orientation ${}^oX_r$, respectively. In the case that the preset keypoints is satisfactory without pose noise, the

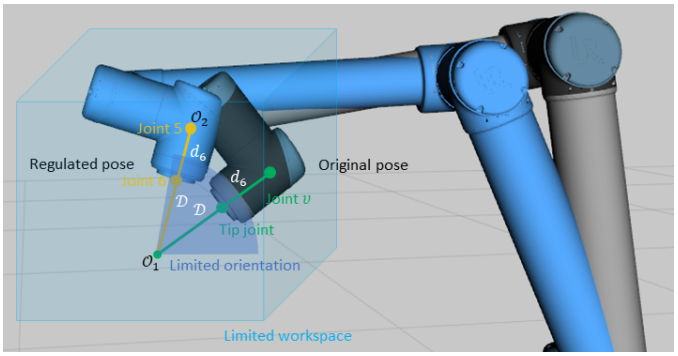


Fig. 2. Regulation of the robot's orientation. The industrial robot with its original color stands for its original predefined pose, while the robot in blue stands for the regulated pose. the cyan cube represents a small workspace centering the tip of the robot with original pose. The navy blue cone denotes the orientation boundary of the industrial robot.

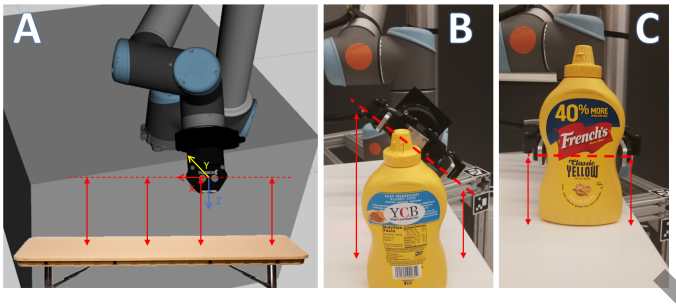


Fig. 3. A. The X axis of the tip coordination of the slave robot is parallel to the ground, which is the surface created by the X axis and Y axis of the base coordinate of the slave robot. B. unparalleled tool tip. C. paralleled tool tip.

following quadratic programming with hierarchical stack of tasks in (11) can be used for teleoperating the slave robot.

$$\begin{aligned} &\min_{\tau_s} \int_0^\infty \frac{1}{2}||\tau_s||_2 dt \\ &\text{Subject to Task 1} \\ &\text{Subject to Task 2} \end{aligned} \tag{11}$$

where τ_s is the control input to the slave. Task 1 and Task 2 are included in PSoT as shown in Algorithm 3.

Algorithm 3 Primitive Stack of Tasks (PSoT)

Task 1: $\underline{{}^tX_{os}} - {}^tX_s \leq {}^tJ_s\tau_s \leq \overline{{}^tX_{os}} - {}^tX_s$
where $J_s = [{}^tJ_s^T, {}^oJ_s^T]^T$ denotes the jacobian matrix of the slave with ${}^tJ_s$ for translation and ${}^oJ_s^T$ for orientation. $\underline{{}^tX_{os}}$ and $\overline{{}^tX_{os}}$ denote the lower and upper boundaries. The workspace created by the boundaries can adapt to the practical task space by using the proposed workspace tuning approach, which will be explained in next section.
Task 2: $J_s\tau_s = F_s$
where F_s is the controller in Cartesian space that will be introduced later. This task is used to map the Cartesian space control input F_s to the joint space control input τ_s. It can be further written as $\tau_s = J_s^\top F_s$ for simplicity, where $J_s^\top$ can be regarded as J_s^{-1}.

According to *Step 5.B*, the pose noise in the area of one keypoint can lead to task failure if the robot using the preset orientation. We take the grasping task as an example: If the object's orientation is changed due to external disturbances, or the preset keypoints in *Step 1* is not satisfactory, the robot with the preset orientation is unable to successfully grasp the object. Therefore, further regulation on the slave robot's motion is needed. This further regulation is performed using an algorithm called MSoT and described in Algorithm 4.

Algorithm 4 Motion-regulated Stack of Tasks (MSoT)

Task 1^*: $\underline{{}^tX_{cs}} - {}^tX_s \leq {}^tJ_s\tau_s \leq \overline{{}^tX_{cs}} - {}^tX_s$
The workspace created by the lower and upper boundaries $\underline{{}^tX_{cs}}$ and $\overline{{}^tX_{cs}}$ are small and its location is variable according to the recorded position ${}^tx_{rec}$ as $\underline{{}^tX_{cs}} = {}^tX_{rec} - {}^t\epsilon$ and $\overline{{}^tX_{cs}} = {}^tX_{rec} + {}^t\epsilon$, where ${}^t\epsilon$ is a vector with positive small elements. This task is utilized to set a small workspace for fine movement.
Task 2^*: $\underline{{}^oX_{cs}} - {}^oX_s \leq {}^oJ_s\tau_s \leq \overline{{}^oX_{cs}} - {}^oX_s$
where $\underline{{}^oX_{cs}}$ and $\overline{{}^oX_{cs}}$ are lower and upper boundaries for the slave orientation. $\underline{{}^oX_{cs}}$ and $\overline{{}^oX_{cs}}$ are derived as $\underline{{}^oX_{cs}} = {}^oX_{rec} - {}^o\epsilon$ and $\overline{{}^oX_{cs}} = {}^oX_{rec} + {}^o\epsilon$, where ${}^o\epsilon$ is a vector with small and positive elements. This task is utilized to restrict the slave orientation.
Task 3^*: ${}^tJ_s\tau_s = {}^tF_s$
where ${}^tF_s$ is the translation control part of the Cartesian space controller F_s. This task is to let the tip of the robot reach the desire position.
Task 4^*: ${}^{t\nu}J_s\tau_s = {}^{t\nu}F_s$
where ${}^{t\nu}J_s$ is the Jacobian matrix from the slave's base to Joint ν. The controller ${}^{t\nu}F_s$ is the Cartesian space controller that allows the position of Joint ν ${}^{t\nu}X_s$ to track $\mathcal{O}_2$, which will be explained in next section.
Task 5^*: Let the X axis of the coordinate of slave end effector be parallel to the ground.

As shown in Fig. 2, when the slave robot enters the range ${}^kr$ of a keypoint, the operator can launch MSoT for motion regulation, where Task 1^* allows the operator to conduct fine movement in a small workspace on the condition that the scaling gains are not amplified. The small workspace can limit the slave robot's position in the range of a key point so that the slave robot can be less likely to be affected by the amplified master reference signals. Also, it is easier for the operator to perform the task.

Task 2^* defines the orientation constraints which allows the orientation of the slave robot to vary in a cone.

Task 3^* and Task 4^* work together to re-regulate the slave robot's orientation. The work process is as follows.

1). At beginning, record the current position and orientation of the slave robot as ${}^tX_{rec}$, ${}^oX_{rec}$.

2). Joint ν is the joint next to the tip joint. Create a vector with a constant length $\mathcal{D}$ which starts from Joint ν, along the link with its length d_6 between Joint ν and the tip joint, to the endpoint $\mathcal{O}_1$.

3). Then, when the operator drives the slave robot to move to a new position, another vector is created, which starts from $\mathcal{O}_1$, passes the tip joint and is with a constant length $\mathcal{D}$.

Accordingly, the location of the endpoint $\mathcal{O}_2$ of this vector can be determined.

Therefore, by letting Joint ν closely track $\mathcal{O}_2$, the orientation of the slave robot can be re-regulated.

To perform tasks better, it is necessary that the slave's tool effector with the re-regulated orientation is parallel to the ground as shown in Fig. 3A. Otherwise, the operator cannot regulate the slave orientation to an optimal pose as shown in Figs. 3B and 3C. The method that lets the tool tip be parallel to the ground is described in Appendix, which is set as a constraint in Task 5^* of Algorithm 4.

The following quadratic programming is used to hierarchically include Task 1^* to Task 5^* in MSoT as constraints.

$$\begin{aligned} &\min_{\tau_s} \int_0^\infty \frac{1}{2}||\tau_s||_2 dt \\ &\text{Subject to Task } 1^* \\ &..... \\ &\text{Subject to Task } 5^* \end{aligned} \tag{12}$$

After the slave's orientation is re-regulated and the task is performed, the reset keypoint g_t is automatically updated, which allows the slave robot to perform the task at the keypoint with a better orientation in the next round. We propose an algorithm called GUR to update g_t as shown in Algorithm 5.

Algorithm 5 Goal Update Rule (GUR)

1. At the range of a keypoint, treat each re-regulated orientation as a trial (totally $\mathbb{K}$ trials, $k = 1...\mathbb{K}$), record the pose as $X_{g,k}$, and calculate their cost-to-go $\mathbb{S}_k$

$$\mathbb{S}_k = \phi_{M,k} + \{\sum_{h=1}^{H}\sum_{m=0}^{M-1} {}^h r_{t,m}\}_k \tag{13}$$

2. When a new trial appears, compare its cost-to-go $\mathbb{S}_{new}$ with $\mathbb{S}_\mathbf{k}$ ($\mathbb{S}_\mathbf{k} = max([\mathbb{S}_1, \mathbb{S}_2, ..., \mathbb{S}_\mathbb{K}])$), and then update $\mathbb{S}_\mathbf{k}$ as

$$\mathbb{S}_\mathbf{k} = \begin{cases} \mathbb{S}_{new} & \text{if } \mathbb{S}_\mathbf{k} > \mathbb{S}_{new} \\ \mathbb{S}_\mathbf{k} & \text{if } \mathbb{S}_\mathbf{k} \leq \mathbb{S}_{new} \end{cases} \tag{14}$$

3. Calculate the probabilities $\mathbb{P}_k$ for each trial.

$$\mathbb{P}_k = \frac{e^{-\frac{1}{\gamma_g}\mathbb{S}_k}}{\sum_{l=1}^{\mathbb{K}} e^{-\frac{1}{\gamma_g}\mathbb{S}_l}} \tag{15}$$

4. Update the new goal g_t at a keypoint

$$g_t = \sum_{k=1}^{\mathbb{K}} \mathbb{P}_k X_{g,k} \tag{16}$$

5. Insert g_t into DMP to create new sequences of trajectories.

In (13), $\phi_{M,k}$ stands for the terminal cost, which can be freely designed according to different tasks. For example, for pick-and-place tasks, $\phi_{M,k}$ can be designed as follows.

$$\phi_{M,k} = \alpha_{c1}(1 - |\kappa_{gr} F_{gr,k}|^{\gamma_{gr}}) \tag{17}$$

where α_{c1} is a gain, which determines the weight of $\phi_{M,k}$. $F_{gr,k}$ denotes the grasp force of the gripper at k-th trial. In this paper, the gripper's grasp force is estimated by using the force observer in [16]. κ_{gr} is a gain to normalize $|\kappa_{gr} F_{gr,k}|$ to be no more than 1. $\gamma_{gr} \geq 1$ is a positive constant.

M in (13) denotes the length of a trajectory vector created by DMP from g_t at one keypoint to the next. H denotes the total number of keypoints. H can be freely adjusted according to the applications. The immediate cost ${}^h r_{t,i}$ is

$${}^h r_{t,i} = \alpha_{c2}||{}^h\delta_{traj,m} - {}^h g_t||^2 \tag{18}$$

where α_{c2} is a constant gain. ${}^h\delta_{traj,i}$ denotes each element of the trajectory vector from the goal ${}^h g_t$ of the other keypoint to that of the current keypoint. In (13), the terminal cost $\phi_{M,k}$ evaluates that whether the task is performed. Lower $\phi_{M,k}$ means larger grasp force, and implies that the slave robot has tightly grasped the object in a good orientation. The immediate cost ${}^h r_{t,i}$ evaluates the orientation's smoothness. The low cost means that the slave robot needs not to twist itself too much from other keypoints to the current one, which is more efficient.

After the orientation is re-regulated by using MSoT, its cost-to-go $\mathbb{S}_{new}$ is evaluated, and is compared with $\mathbb{S}_\mathbf{k}$. If $\mathbb{S}_{new}$ is lower than $\mathbb{S}_\mathbf{k}$, it will replace $\mathbb{S}_\mathbf{k}$ as shown in (14). Then, calculate the probabilities based on those $\mathbb{S}_k$, in which lower $\mathbb{S}_\mathbf{k}$ determines larger probability. γ_g in (15) is a positive constant gain that determines the weight. With these probabilities, the new goal g_t can be calculated from (16). Finally, update the new g_t into DMP to calculate the new sequences of trajectories.

III. Master-Slave Controllers

This section describes the proposed workspace tuning approach to resolve the problem of workspace inequality in *Step 4*, and then introduces the control laws. Due to the page limit and the different academic focus, the environmental force detection and reflection, and the related transparency issue will not be discussed in this paper, which has been analyzed in depth in [16]. The force controller proposed in [16] can be directly added into the control law of this paper. The human force F_h and environmental force F_e in this paper are assumed to be fully estimated by the observer in [16].

We define the asymmetric position control errors and orientation errors $e_s = [{}^t e_s^T, {}^o e_s^T]^T$ between master and slave for the use of PSoT

$$\begin{aligned} {}^t e_s(t) &= {}^t X_s(t) - \Gamma_A(t) {}^t X_m(t - T_f(t)) + \xi_{off} \\ {}^t e_m(t) &= \Gamma_A(t - T_b(t)) {}^t X_m(t) - {}^t X_s(t - T_b(t)) + \xi_{off} \\ {}^o e_s(t) &= {}^o X_s(t) - {}^o X_r(t) \end{aligned} \tag{19}$$

where T_f and T_b are the feed-forward and feedback time-varying delays between the master and the slave. The differentials of time delays are bounded. That is, $0 \leq |\dot{T}_{f,b}| \leq \bar{d}_{f,b}$. The time delays $T_{f,b}$ also have their upper and low bounds as $\underline{T}_{f,b} \leq T_{f,b} \leq \overline{T}_{f,b}$. The term ξ_{off} is a offset vector that maps the origin of the master manipulator to the center of the desired original workspace of the slave robot.

The term $\Gamma_A = diag([\Gamma_{Ax}, \Gamma_{Ay}, \Gamma_{Az}])$ is an adaptive diagnal matrix gain, which is used to amplify the master ref-

erence position at X, Y, Z directions. We define the following adaptive laws for the gain Γ_A

$$\Gamma_{Ai} = \begin{cases} \Gamma_{Ai1} & \text{if } {}^tX_{mi}(t-T_f) \geq 0 \\ \Gamma_{Ai2} & \text{if } {}^tX_{mi}(t-T_f) < 0 \end{cases} \quad i = x, y, z \tag{20}$$

$$\begin{aligned} &\Gamma_{Aij} = {}^1\gamma_{Aij} + {}^2\gamma_{Aij}, \quad j = 1, 2 \\ &{}^1\gamma_{Aij} = \max(\Gamma_{Aij}) - {}^2\gamma_{Aij} \\ &{}^2\dot{\gamma}_{Aij} = \alpha_A(-\kappa_{A1}{}^2\gamma_{Aij} + \kappa_{A2}\mathscr{P}_j(F_{hi}(t-T_f))) \\ &\quad + (1-\alpha_A)(\kappa_{A3}\mathscr{P}_j(F_{hi}(t-T_f))) \end{aligned} \tag{21}$$

where $\max(\Gamma_{Aij})$ denotes the highest historical value of Γ_{Aij}. The gains κ_{A1}, κ_{A2} and κ_{A3} are positive constant gains. α_A is a variable gain ($0 \leq \alpha_A \leq 1$) expressed as

$$\alpha_A = \frac{1}{2}tanh(h_A\chi_e) + \frac{1}{2} \tag{22}$$

where h_A is a constant parameter. χ_e is derived as

$$\chi_e = {}^kr - \max(\begin{bmatrix} \min(|{}^tX_{sx}\mathbb{I} - {}^tX_{gx}|) \\ \min(|{}^tX_{sy}\mathbb{I} - {}^tX_{gy}|) \\ \min(|{}^tX_{sz}\mathbb{I} - {}^tX_{gz}|) \end{bmatrix}) \tag{23}$$

The term F_{hi} denotes the human felt force at i direction. The function $\mathscr{P}_j(F_{hi}))$ in (21) is

$$\begin{aligned} \mathscr{P}_1(F_{hi})) &= \begin{cases} |F_{hi}| - \bar{F}_{hi} & \text{if } |F_{hi}| \geq \bar{F}_{hi} \,\&\, {}^tX_{mi} \geq 0 \\ 0 & \text{else} \end{cases} \\ \mathscr{P}_2(F_{hi})) &= \begin{cases} |F_{hi}| - \bar{F}_{hi} & \text{if } |F_{hi}| \geq \bar{F}_{hi} \,\&\, {}^tX_{mi} < 0 \\ 0 & \text{else} \end{cases} \end{aligned} \tag{24}$$

where $\bar{F}_{hi}$ denotes the threshold.

The slave position barriers ${}^t\underline{X_{os}} = [{}^t\underline{X_{osx}}, {}^t\underline{X_{osy}}, {}^t\underline{X_{osz}}]^T$ and ${}^t\overline{X_{os}} = [{}^t\overline{X_{osx}}, {}^t\overline{X_{osy}}, {}^t\overline{X_{osz}}]^T$ in Task 1 of PSoT are updated according to Γ_{Ai} as

$$\begin{aligned} {}^t\underline{X_{osi}} &= \Gamma_{Ai}({}^t\underline{x_{osi}} - p_{oi}) + p_{oi} \\ {}^t\overline{X_{osi}} &= \Gamma_{Ai}({}^t\overline{x_{osi}} - p_{oi}) + p_{oi} \end{aligned} \tag{25}$$

where ${}^t\underline{x_{osi}}$ and ${}^t\overline{x_{osi}}$ are the initial lower and upper bounds that build the initial workspaces for the slave robot. $p_o = [p_{ox}, p_{oy}, p_{oz}]^T$ is the original position of the slave robot.

To further guarantee the motion smoothness, we define the velocity boundary $\mathbf{B}_v = [\mathbf{B}_{vx}, \mathbf{B}_{vy}, \mathbf{B}_{vz}]^T$ as

$$\mathbf{B}_{vi} = (\mathbf{b}_u - \mathbf{b}_l)e^{-\iota|{}^te_{si}|} + \mathbf{b}_l, \quad i = x, y, z \tag{26}$$

where b_u, and b_l are the upper bound and lower bound. ι is positive gain. $\mathbf{B}_v$ decreases along with the position error ${}^te_s$ increasing with the rate ι.

The proposed workspace tuning approach is (20)-(26), whose logic is as follows. The value of Γ_{Ai} is separated based on the original position ($[0,0,0]^T$) of the master in (20) to the extent that the gain Γ_{Ai1} at one direction will not influence the gain Γ_{Ai2} at the opposite direction. In (21), the initial value of ${}^1\gamma_{Aij}$ is 1, and ${}^2\gamma_{Aij}$ is 0 at first. Therefore, the gain Γ_{Ai} is 1 at the beginning which means the workspace of the slave is small and has the same size as that of the master. (In the paper, the master's workspace is a preset workspace, which is a little smaller than its practical workspace. This workspace is built using (31).) When the slave robot reaches its initial boundaries ${}^t\underline{x_{osi}}$ or ${}^t\overline{x_{osi}}$ and stops, if the human operator keeps moving the master manipulator forward, the operator can feel a spring-like feedback force, which makes the human applied force F_h increases at a certain direction. The spring-like feedback force is caused by the master-slave position error tuned by a variable gain $\mathcal{K}_m$ in (33). Therefore, $\mathscr{P}_j(F_{hi}))$ in (24) can be increased by the human force. As illustrated in (10), χ_e in (23) is a criteria that evaluates whether the slave robot is at the area of a keypoint, in which $\chi_e \geq 0$ denotes that the slave robot is at the area and vice versa. Accordingly, $\chi_e \geq 0$ leads the variable gain α_A in (22) to converge to 1, which simplifies the adaptive law ${}^2\dot{\gamma}_{Aij} = -\kappa_{A1}{}^2\gamma_{Aij} + \kappa_{A2}\mathscr{P}_j(F_{hi})$ in (21). It means that ${}^2\gamma_{Aij}$ tracks the increased $\mathscr{P}_j(F_{hi}))$ so that Γ_{Aij} gradually increases. Since any keypoint's area can be the main place to conduct the task, the gradually increased workspace is more reliable for performing the task. On the other hand, when the slave robot is not at the key point and α_A converges to zero, the adaptive law is changed to be ${}^2\dot{\gamma}_{Aij} = \kappa_{A3}\mathscr{P}_j(F_{hi})$. This means that the gain Γ_{Aij} accelerates its increase rate to let the slave robot fast enlarge its workspace. Finally, By adding ${}^1\gamma_{Aij}$, the gain Γ_{Aij} will only increase or keep its current value, but never decrease. The adaptively tuned scaling gain Γ_A, the valocity boundary $\mathbf{B}_v$, and the dynamically regulated workspace created by ${}^t\underline{X_{os}}$ and ${}^t\overline{X_{os}}$ efficiently work together to guarantee a smooth movement.

When PSoT is switched to MSoT, the position control errors is changed to be

$$\begin{aligned} {}^te_s(t) &= {}^tX_s(t) + {}^tX_{rec} - {}^tX_m(t - T_f(t)) \\ {}^te_m(t) &= {}^tX_m(t) - ({}^tX_s + {}^t X_{rec})(t - T_b(t)) \end{aligned} \tag{27}$$

The velocity boundary in (26) will also prevent the sudden jump between (19) and (27).

The dynamics of the robots, including master and slave, can be estimated by the Type-2 fuzzy neural network proposed in [16], which can attain high accuracy and is robust against uncertainties, and can be expressed by a combination of multiple linear local models as:

$$M_i\ddot{q}_i + C_i\dot{q}_i + D_iq_i + E_i = \tau_i + J_i^TF_{h/e} \tag{28}$$

where $i = m/s$ stands for master/slave; M_i, C_i, D_i and E_i are weighted sums of the local models' coefficients with dynamic fuzzy membership grades as the weights. Thus, M_i, C_i, D_i and E_i are known time-varying parameters to describe the nonlinear robotic system; q_i, $\dot{q}_i$, and $\ddot{q}_i$ are the vectors of joint displacement, velocity and acceleration;

According to the above equations, we design the slave control law in PSoT as

$$\begin{aligned} &\tau_s = J_s^\top F_s, \\ &\mathbf{r}_{s1} = \Lambda_{s1}{}^te_s + \dot{X}_s, \\ &F_s = -\mathcal{B}_vk_{s1}\mathbf{r}_{s1} - M_s\Lambda_{s1}(\dot{X}_s - \mathcal{C}_s\dot{X}_r) - M_s\dot{J}_s\dot{q}_s + C_s\dot{X}_s \\ &\quad + D_sX_s + E_s - J_s^TF_e \end{aligned} \tag{29}$$

Fig. 4. Experiment setup

where Λ_{s1} is a positive diagonal matrix. k_{s1} is a constant gain. $\mathcal{B}_v$ is a diagonal matrix expressed as $\mathcal{B}_v = diag([\mathbf{B}_{vx}^2 - \dot{X}_{sx}^2, \mathbf{B}_{vy}^2 - \dot{X}_{sy}^2, \mathbf{B}_{vz}^2 - \dot{X}_{sz}^2, 1, 1, 1])$. $\dot{X}_r = [\dot{X}_m^T(t - T_f), {}^oX_r^T]^T$, $\mathcal{C}_s = diag([\Gamma_{Ax}(1 - \hat{T}_f), \Gamma_{Ay}(1 - \hat{T}_f), \Gamma_{Az}(1 - \hat{T}_f), 1, 1, 1])$, $\hat{T}_f$ and $\hat{T}_b$ are the estimated T_f and T_b using the time delay differential estimator in [39].

The position controller ${}^{t\nu}F_s$ used for Task 4* in MSoT as

$$\begin{aligned} {}^{t\nu}F_s &= -k_{s2}\mathbf{r}_{s2} \\ \mathbf{r}_{s2} &= \Lambda_{s2}({}^{t\nu}X_s - \mathcal{O}_2) + {}^{t\nu}\dot{X}_s \end{aligned} \tag{30}$$

where Λ_{s2} is a positive diagonal matrix and k_{s2} is a constant.

At the master side, the operator will feel a spring-like force feedback when the robot is constrained by the barriers of its workspace, which increases the human force F_h and makes the equations (24) work. Accordingly, we propose the hyperplane weighting function as (31) where $\mu_{hm} > 1$, ${}^ub_{hm} > 1 \gg {}^lb_{hm} \geq 0$ and

$$sat_2(x) = \begin{cases} {}^ub_{hm} & \text{if } x \geq {}^ub_{hm} \\ x & \text{if } {}^lb_{hm} < x < {}^ub_{hm} \\ {}^lb_{hm} & \text{if } x \leq {}^lb_{hm} \end{cases} \tag{32}$$

The master workspace is created by using (31). Its logic is that when the slave end effector is conducting free motion inside the defined workspace, κ_{hm} converges to ${}^ub_{hm}$. On the other hand, when the slave end effector reaches the defined barriers, κ_{hm} decreases to ${}^lb_{hm}$. Based on this, we define the master control law as

$$\begin{aligned} &\tau_m = J_m^\top F_m, \\ &\mathbf{r}_m = \Lambda_m {}^te_m + \dot{X}_m, \\ &\mathcal{K}_m = diag([|F_{hx} + \lambda_h|^{-\kappa_{hmx}}, |F_{hy} + \lambda_h|^{-\kappa_{hmy}}, \\ &|F_{hz} + \lambda_h|^{-\kappa_{hmz}}]), \\ &F_m = -\mathcal{K}_m\mathbf{r}_m - M_m\Lambda_m(\dot{X}_m - \Gamma_A(t - T_b)(1 - \hat{T}_b) \\ &\dot{X}_m(t - T_b)) - M_m\dot{J}_m\dot{q}_m + C_m\dot{X}_m + D_mX_m + E_m \\ &- J_m^TF_h \end{aligned} \tag{33}$$

where Λ_m is a positive diagonal matrix. λ_h is a small parameter close to zero. Based on (33), when κ_{hm} is ${}^ub_{hm}$, $\mathcal{K}_m$ is close to zero that makes the operator have little force perception. When κ_{hm} decreases to ${}^lb_{hm}$, $\mathcal{K}_m$ increases, which provides the operator with a large force feedback. The large force feedback increases F_h, which then influences (24) of the proposed workspace tuning approach.

The stability of the slave and master control laws (29) and (33) is proved in Appendix.

IV. Experimental Results

This section presents the experiment results of the proposed ABT control algorithm. The experiment setup is shown in Fig. 4. The master haptic device is a Geomagic Touch and the slave robot is a UR10 robot. A gripper (Robotiq-85) is used to perform the pick-and-place tasks. Two computers are utilized to drive the master haptic device and the slave robot. The time delays in the communication channels between the two computers are around 100 ± 10 ms.

A. DOG

First, we evaluate the proposed DOG. After the keypoints are roughly preset, DMP is used to create trajectories between every two keypoints and DOG is utilized to determine the reference orientation ${}^oX_r$ from the trajectories. Given two keypoints (Point 1 = [-0.6423, -0.06, 0.3277, 0.6678, -0.665, -0.221, 0.2509], Point 2 = [-0.6453, -0.06, 0.6339, 0.5167, -0.5178, -0.4863, 0.478]), we firstly let the slave robot autonomously follow the trajectory between these two keypoints created by DMP as shown in Fig. 5A, and then tele-operate the slave robot to move from Point 1 to Point 2 with its orientation following the reference orientation ${}^oX_r$ determined by DOG as shown in Fig. 5B. In Fig. 5A, the sampling rate and preset time interval in the DMP are 0.01 and 1s, respectively. Therefore, there are 101 groups of pose elements in this trajectory. When the robot moves up following the trajectory, small signal jumps occur. In order to smooth those signal jumps, the sampling rate needs to be tuned very high, which means that a large number of pose elements are be generated in one trajectory. if a large series of trajectories are required, such large number of pose elements may increase the computational complexity. Also, if two keypoints are close to each other, the generated large numbers of pose elements are unnecessary.

In comparison, the sampling rate in Fig. 5B is 0.1 (11 pose elements in one trajectory). By using DOG, the reference orientation ${}^oX_r$ and the slave orientation ${}^oX_s$ smoothly vary in continuous state without signals jump. Also, the required number of pose elements is also small.

B. Enlarge workspaces

When switching to bilateral teleoperation, the operator needs to firstly enlarge the workspace of the slave robot and the master scaling gain Γ_A in order to allow the workspace of the operator to be exactly fit for the specific task space as shown in Fig. 6. Fig. 7 and Fig. 8 demonstrate the process of operator enlarging the workspace using the proposed method. At beginning (0s - 30s), the master controls the slave robot to conduct free motion, with the master scaling gain Γ_A to be $\Gamma_{Ax1} = 1$, $\Gamma_{Ax2} = 1$, $\Gamma_{Ay1} = 1$, $\Gamma_{Ay2} = 1$, $\Gamma_{Az1} = 1$, $\Gamma_{Az2} = 1$. The initial position barriers in PSoT are $\underline{{}^tX_{osx}} = -0.6$, $\overline{{}^tX_{osx}} = 1$, $\underline{{}^tX_{osy}} = -0.1$, $\overline{{}^tX_{osy}} = 0.1$, $\underline{{}^tX_{osz}} = 0.26$, $\overline{{}^tX_{osz}} = 0.4$. the position offset ξ_{off} is $[-0.5, 0, 0.3]^T$, which means the origin position of the slave position is p_o is $[-0.5, 0, 0.3]^T$. During 0s - 30s, the slave robot is following the master's actual position. Then, during 30s - 170s, the slave robot reaches its barriers and conducts constraint motion. From 30s to 100s, the operator

$$\kappa_{hm} = \begin{bmatrix} \kappa_{hmx} \\ \kappa_{hmy} \\ \kappa_{hmz} \end{bmatrix} = \begin{bmatrix} sat_2(-\frac{{}^{u}b_{hm} - {}^{l}b_{hm}}{|\frac{1}{2}({}^{t}\overline{X_{osx}}(t-T_b) - {}^{t}\underline{X_{osx}}(t-T_b))^{\mu_{hm}}|}|{}^{t}X_{sx}(t-T_b) - \frac{{}^{t}\overline{X_{osx}}(t-T_b) + {}^{t}\underline{X_{osx}}(t-T_b)}{2}|^{\mu_{hm}} + {}^{u}b_{hm}) \\ sat_2(-\frac{{}^{u}b_{hm} - {}^{l}b_{hm}}{|\frac{1}{2}({}^{t}\overline{X_{osy}}(t-T_b) - {}^{t}\underline{X_{osy}}(t-T_b))^{\mu_{hm}}|}|{}^{t}X_{sy}(t-T_b) - \frac{{}^{t}\overline{X_{osy}}(t-T_b) + {}^{t}\underline{X_{osy}}(t-T_b)}{2}|^{\mu_{hm}} + {}^{u}b_{hm}) \\ sat_2(-\frac{{}^{u}b_{hm} - {}^{l}b_{hm}}{|\frac{1}{2}({}^{t}\overline{X_{osz}}(t-T_b) - {}^{t}\underline{X_{osz}}(t-T_b))^{\mu_{hm}}|}|{}^{t}X_{sz}(t-T_b) - \frac{{}^{t}\overline{X_{osz}}(t-T_b) + {}^{t}\underline{X_{osz}}(t-T_b)}{2}|^{\mu_{hm}} + {}^{u}b_{hm}) \end{bmatrix} \quad (31)$$

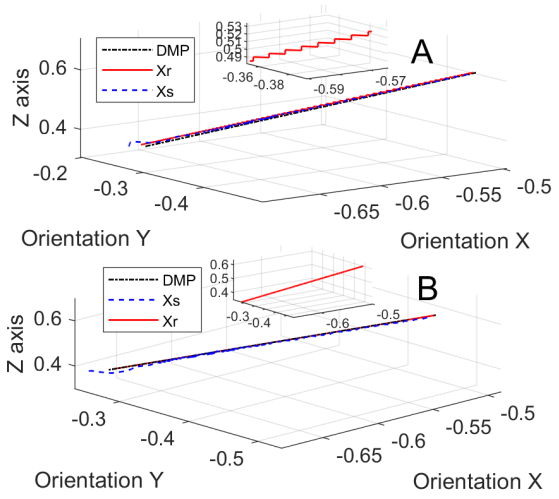


Fig. 5. Comparison between the trajectory created by DMP and DOG

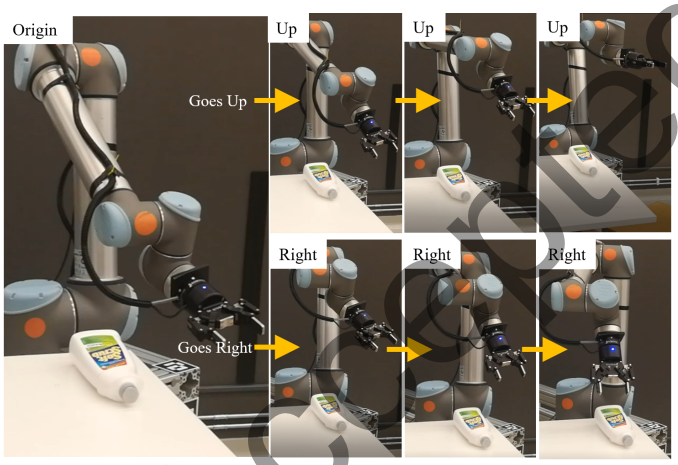


Fig. 6. Procedures of enlarging workspace

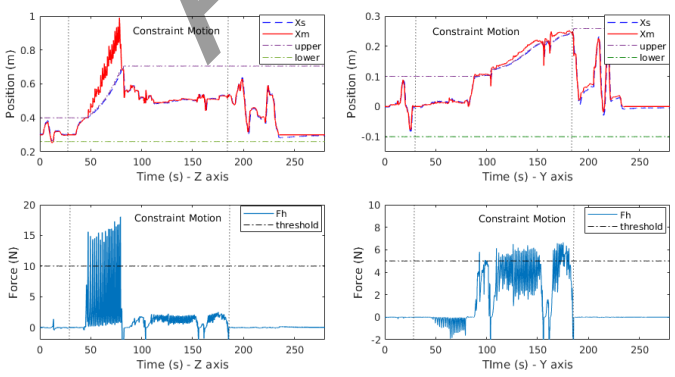


Fig. 7. Position tracking and human applied force when enlarging the workspace

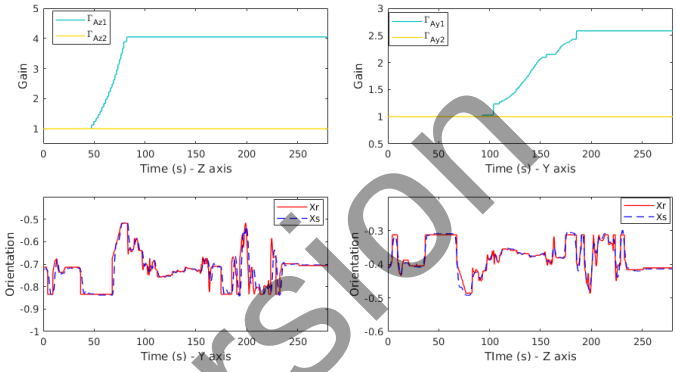


Fig. 8. The change of Γ_A and orientation tracking

drives the slave robot to move up; and then from 100s to 170s, the operator drives the slave robot to move right. when the transmitted master position exceed the barrier, the operator can feel a large spring-like force according to (31). Then, by using the workspace tuning approach (20)-(25), the master scaling gain Γ_A and the barriers ${}^{t}\underline{X_{os}}$, ${}^{t}\overline{X_{os}}$ also increase based on the human applied forces. By properly setting the adaptive controller in (21) ($\kappa_{A1} = 20$, $\kappa_{A2} = 5$, $\kappa_{A3} = 1$), the increasing rate is slow which provides the operator with enough time to decide whether the enlarged workspace is fit for the required task space. After 170s, the slave robot stops conducting constraint motion, and the operator then drives the slave to conduct similar free motion as that in the period (0s-30s). However, compared to that in 0s -30s, the slave robot now makes a much larger movement. The scaling gain Γ_A is also increased. The first two graphs in Fig. 8 illustrate the comparison between the increased gains and unraised gains, which makes the scales of the robot moving up and down, moving right and left are totally different. (e.g. When the robot moving downside its origin, the slave robot still closely follows the actual master position because of the unchanged scaling gain Γ_{Az1}). In addition, from Fig. 6, it clearly shows the gradual variation of the slave orientation from one keypoint to another, which proves the proposed DOG can effectively prevent sudden jumps of slave orientation. These gradually changing orientations benefit the robot to perform tasks.

For comparison, we validate the scaled control method that is used by the previous study (e.g. [34], [40]), where the master scaling gain is set as constant. Fig. 9 demonstrates the experiment, where the task is teleoperating a slave robot to grasp a bottle. Since the actual workspace is unknown, we set the master scaling gain to be $[5, 5, 10]$ that is large enough to cover the task space. The first two figures show the slave position tracking the amplified master position and the second

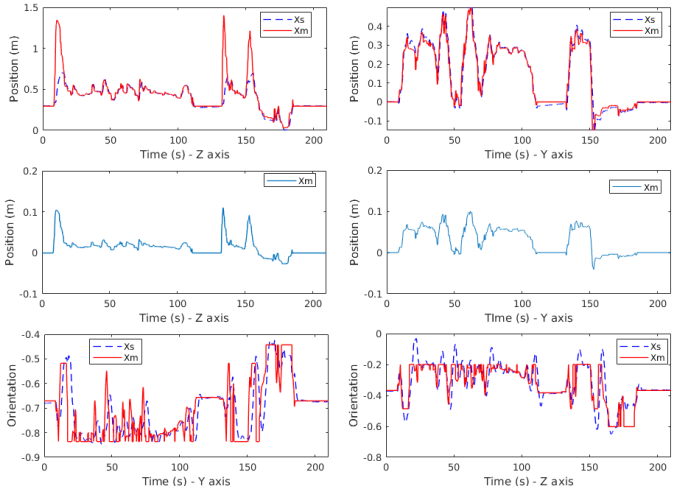


Fig. 9. Position and orientation tracking in the experiment with constant gain for comparison

two figures show the actual master position. From the four figures, we can see that any little movement of the operator can lead to a sharp movement of the slave robot. Moreover, since the slave orientation definition is based on the slave position in DOG, the slave orientation is drastically varying as shown in the final two figures. The operator can barely perform the required task.

C. Single task

After the slave workspace is enlarged to match the actual task space, the slave robot is ready to perform the pick-and-place task. In this subsection, the slave robot is controlled to pick one fallen bottle at one table (keypoint) and place it to stand upright on the other table (keypoint). Figs. 10 and 11 demonstrate position tracking, human applied force, orientation tracking and regulation, and the gripper applied force. When the slave robot moves to the desired position and is ready to grasp the bottle, PSoT is switched to MSoT (10s-50s), in which the slave workspace is changed to be a small rectangle centered by the current slave position ($\overline{{}^tX_{csx}} - \underline{{}^tX_{csx}} = 0.2$, $\overline{{}^tX_{csy}} - \underline{{}^tX_{csy}} = 0.2$, $\overline{{}^tX_{csz}} - \underline{{}^tX_{csz}} = 0.1$) in order to restrict the slave robot's movement. The equation (27) is also utilized to support the operator to conduct fine movements. After grasping the bottle, the robot starts to move down in order to place the bottle to a lower table. From 110s to 190s, the slave robot are in constraint motion in order to enlarge the lower boundary $\underline{{}^tX_{osz}}$ (from 0.26 to 0.01). During the constraint motion, the slave robot is slowly moving down, which allows the operator to have enough time to regulate the orientation and place the object (let the fallen bottle stand upright on the other table). Compared with Fig. 9, Fig. 11 shows the smoothly varying slave orientation, which validates the superiority of the proposed algorithm.

Fig. 12 demonstrates the orientation regulation when performing this task. **Task** 4^* in MSoT allows the tip of the slave robot rotating like a cone, and **Task** 5^* allows the gripper keep parallel to the table. Therefore, it is easy to regulate the robot orientations from the primitive orientations in the

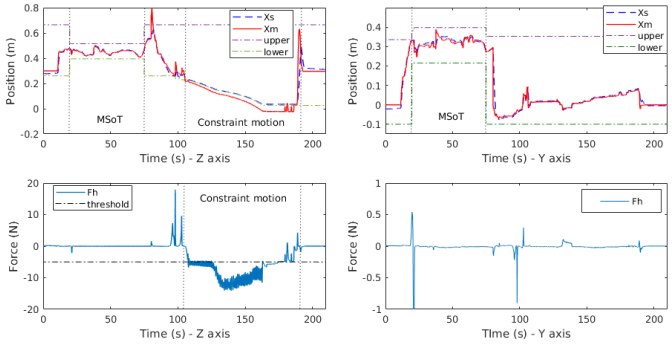


Fig. 10. Position tracking and human applied force in the pick-and-place task

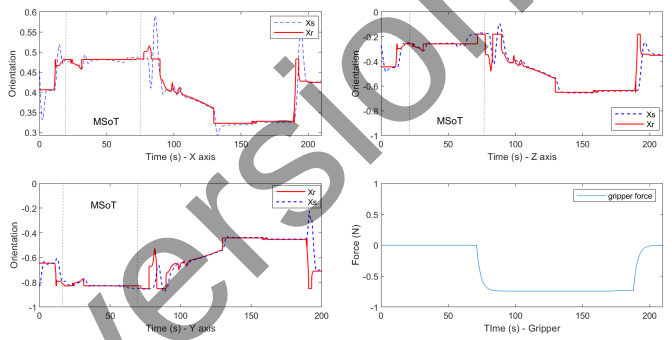


Fig. 11. Orientation tracking and the observed gripper force

keypoints generated by using admittance control to the optimal orientations for performing tasks.

D. Multiple tasks

In the final experiment, the operator derives the slave robot to pick and place three objects which are placed with different poses. The experiment procedure is shown in Fig. 13. At Procedure 1 (P1 in short), three keypoints are roughly determined (Fig. 13A-C), in which the primitive desired orientations (Table I) are included in the three keypoints. Then, at P2 (Fig.

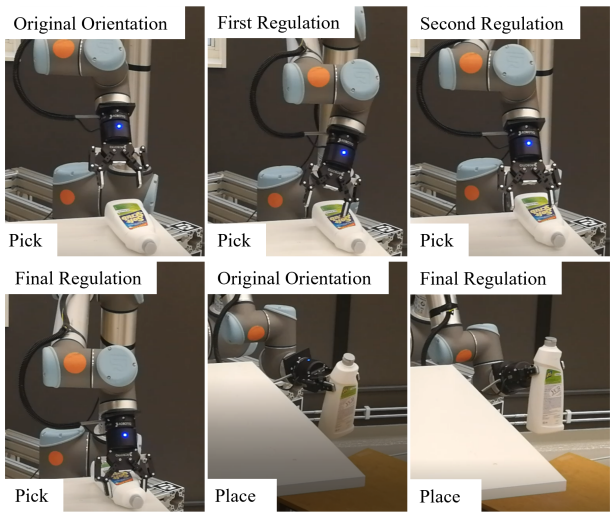


Fig. 12. Regulating the slave orientation when picking and placing a bottle

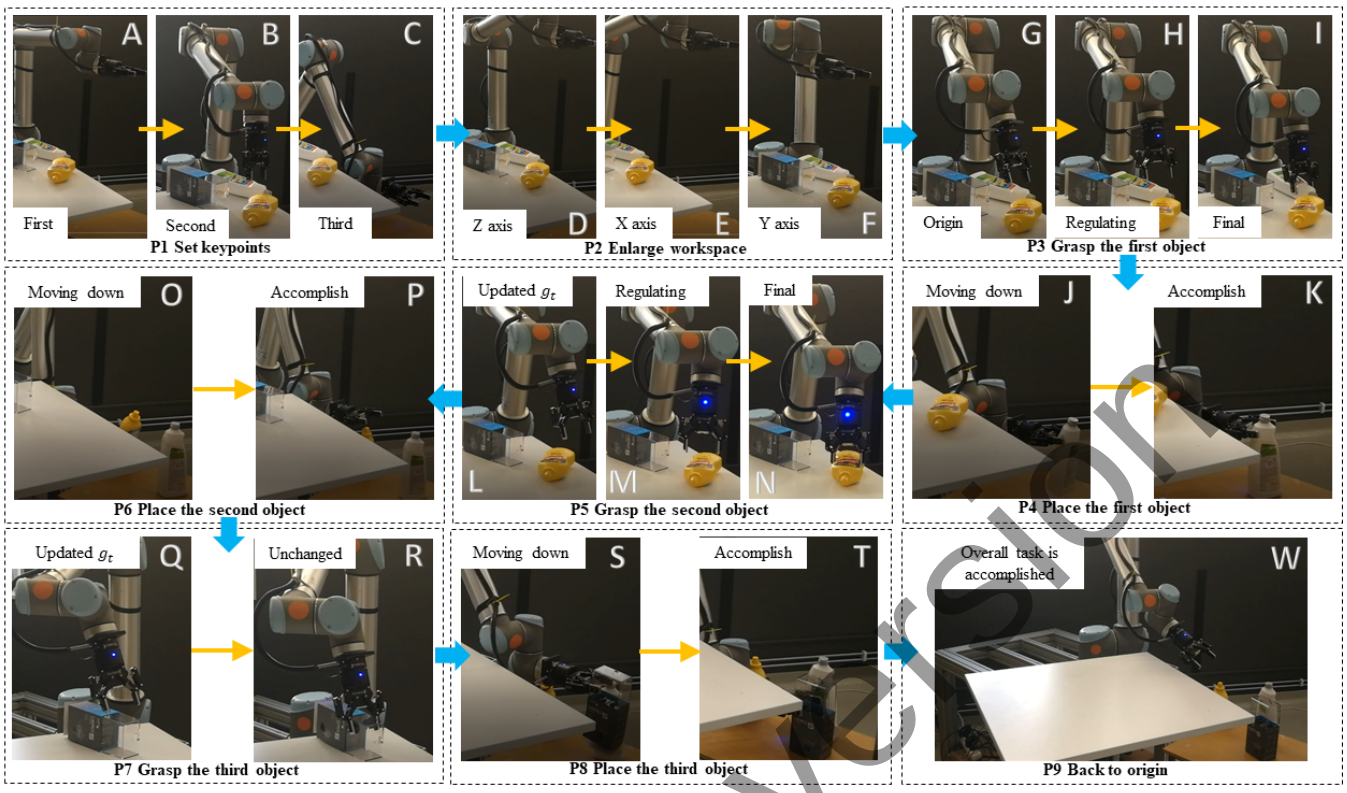


Fig. 13. Overall procedure of picking and placing three objects (a white bottle, a yellow bottle and a box) with different orientations.

TABLE I
KEY ORIENTATIONS

	${}^{o}X_{sx}$	${}^{o}X_{sy}$	${}^{o}X_{sz}$	${}^{o}X_{sw}$
1^{st} keypoint	0.5166	-0.5174	-0.486	0.478
2^{nd} keypoint	0.7544	-0.6457	-0.1184	0.001
3^{rd} keypoint	0.3891	-0.4325	-0.5815	0.5687
1^{st} grasp	0.5864	-0.7939	0.1353	0.08377
2^{nd} origin	0.5867	-0.7901	-0.1361	0.08137
2^{nd} grasp	0.2718	-0.9251	-0.2401	0.1101
3^{rd} origin/grasp	0.5842	-0.7881	-0.1283	0.09073

TABLE II
FINALLY ENLARGED Γ_A, $\underline{{}^{t}X_{os}}$ AND $\overline{{}^{t}X_{os}}$

Γ_A	$\underline{{}^{t}X_{os}}$ and $\overline{{}^{t}X_{os}}$
$\Gamma_{Ax1} = 1.053$	$\overline{{}^{t}X_{osx}} = 1$
$\Gamma_{Ax2} = 3.167$	$\underline{{}^{t}X_{osx}} = -0.8167$
$\Gamma_{Ay1} = 4.134$	$\overline{{}^{t}X_{osy}} = 0.4134$
$\Gamma_{Ay2} = 1.766$	$\underline{{}^{t}X_{osy}} = -0.1766$
$\Gamma_{Az1} = 5.321$	$\overline{{}^{t}X_{osz}} = 0.8321$
$\Gamma_{Az2} = 9.673$	$\underline{{}^{t}X_{osz}} = -0.0869$

13D-F), the operator teleoperates the slave robot to enlarge its workspace. The final enlarged gain Γ_A and barriers $\underline{{}^{t}X_{os}}$ and $\overline{{}^{t}X_{os}}$ are shown in Table II. At P3 (Fig. 13G-I), the slave robot is controlled to grasp the first white bottle. After re-regulating the orientation using MSoT, where Task 4^* helps the robot's tip point down to the ground. The robot can then grasp the white bottle, where the current orientation is recorded in Table I (1^{st} grasp). Then, at P4 (Fig. 13J-K), the slave robot stably places the white bottle onto the other table in another angle to let the fallen bottle stand upright. At P5 (Fig. 13L-N), the slave robot starts to pick the second yellow bottle. Note that because of the last successful pick, the goal orientation in the range of Keypoint 2 is updated to be the robot orientation of grasping the first bottle (1^{st} grasp) by using GUR with its cost-to-go value being 5.8461. Therefore, original orientation for the second grasp (2^{nd} origin) is same as the orientation of grasping the first bottle (1^{st} grasp). However, due to the large difference between the orientation of the second yellow bottle and that of the first white bottle, the robot with its current orientation is unable to grasp the second bottle. Therefore, MSoT is used to re-regulate the robot orientation. Note that by using MSoT, Task 5^* makes the robot gripper always parallel to the table and the operator can adjust the orientation to be optimal in a short time interval. Also, the current robot orientation of picking the second bottle is recorded and its cost-to-go value is 8.2735. The reason that the cost-to-go value is larger is that the robot needs to twist more if the robot moves from Keypoint 1 or 3 to Keypoint 2 (the immediate cost ${}^{h}r_{t,i}$ is larger). At P6 (Fig. 13O-P), the robot also stably places the yellow bottle to allow it to stand upright. At P7 (Fig. 13Q-R), the slave robot starts to grasp the third object, a box. Since the cost-to-go value of the orientation of grasping the first bottle (1^{st} grasp) is lower than that of the orientation of grasping the second bottle 2^{nd}, the original orientation at the third grasp (3^{rd} origin/grasp) is still same as the orientation of the first grasp. Also, since the pose of the box is similar as that of the first bottle, the robot can grasp the box in an optimal

orientation without changing its orientation. Later, at P8 (Fig. 13S-T), the slave robot stably places the box onto the other table and keeps it stand upright. Finally, at P9 (Fig. 13W), the operator drives the slave robot back to its origin and completes the overall task.

V. Conclusion

This paper proposes a novel shared autonomy control strategy for ABT, which allows the operator to remotely control the slave robot with optimal orientation regulation in an adaptive workspace. In this control strategy, the orientation definition algorithm DOG is proposed. Combined with DMP, the slave robot can have a smooth and reasonable orientation change when the slave robot is moved to arbitrary positions in the overall worksapce. A new stack of task MSoT is proposed that allows the operator to drive the slave robot to conduct fine movement and moreover, provides the slave robot with the ability to regulate its orientation. In master-slave control laws, a workspace tuning approach is proposed to update the scaling gains and the barriers of workspace to the extent that the robot's workspace can adapt to different task spaces. Based on the shared autonomy control supported by the above new algorithms, the human operator can solely use position command to perform tasks with various orientation regulation, which can effectively alleviate the operator's burden. Experiments for different scenarios and multiple tasks are conducted by using an experiment platform which consist of a 3-DoF haptic device and a 6-DoF UR10 robot. The experiment results show the feasibility of the proposed strategy.

Appendix

A. Stability of the Bilateral Teleoperation System

Before proving the system stability, some lemmas is provided as follows

Lamma 1: (*Schur complement*) Let M, P, Q be the given matrices such that $Q > 0$. Then

$$\begin{bmatrix} P & M^T \\ M & -Q \end{bmatrix} < 0 \Leftrightarrow P + M^T Q^{-1} M < 0 \tag{34}$$

Lamma 2:[41] For any constant matrix $M \in \mathbb{R}^{n*n}$, $m = m^t > 0$, and $\beta \leq \eta \leq \alpha$, the following inequalities hold:

$$(\alpha - \beta) \int_\beta^\alpha \dot{x}^T(\eta) M \dot{x}(\eta) d\eta \geq (\int_\beta^\alpha \dot{x}(\eta) d\eta)^T M \int_\beta^\alpha \dot{x}(\eta) d\eta \tag{35}$$

Adding the control laws (29) and (33) into (28) the following equation is derived.

$$\dot{\mathbf{r}} = \mathscr{A}_1 \mathbf{r} + \mathscr{A}_2 \dot{X}(t-T) + \mathscr{A}_3 X(t-T) + \mathscr{A}_4 X + \mathscr{A}_5 \mathcal{F} \tag{36}$$

where $\mathbf{r} = [\mathbf{r}_s^T, \mathbf{r}_m^T]^T$, $X(t-T) = [X_s^T(t-T_b), X_m^T(t-T_f)]^T$, $\dot{X}(t-T) = [\dot{X}_s^T(t-T_b), \dot{X}_m^T(t-T_f)]^T$, $\mathcal{F} = [X_s^T, (X_s(t-T_b)]^T$. $\mathscr{A}_1 = \begin{bmatrix} -\mathcal{B}_v M_s^{-1} k_{s1} & 0 \\ 0 & -M_m^{-1} \mathcal{K}_m \end{bmatrix}$, $\mathscr{A}_2 = \begin{bmatrix} 0 & \Lambda_s \mathcal{H}_{s1} \\ \Lambda_m \mathcal{H}_{m1} & 0 \end{bmatrix}$, $\mathscr{A}_3 = \begin{bmatrix} 0 & \Lambda_s \mathcal{H}_{s2} \\ \Lambda_m & 0 \end{bmatrix}$, $\mathscr{A}_4 = \begin{bmatrix} \Lambda_s & 0 \\ 0 & \Lambda_m \mathcal{H}_{m2} \end{bmatrix}$, $\mathscr{A}_5 = \begin{bmatrix} -\Lambda_s & 0 \\ 0 & -\Lambda_s \end{bmatrix}$. $\mathcal{H}_{s1} = diag([\dot{T}_f - \dot{\hat{T}}_f, 0])$,$\mathcal{H}_{s2} = diag([-\dot{\Gamma}_A, 0])$, $\mathcal{H}_{m1} = \dot{T}_b - \dot{\hat{T}}_b$, $\mathcal{H}_{m2} = (1 - \dot{T}_b)\dot{\Gamma}_A(t - T_b)$.

We also define an output $\mathbf{e}$.

$$\begin{aligned} \mathbf{e} &= \begin{bmatrix} 1 & -1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} X + \begin{bmatrix} 0 & 0 \\ 1 & -1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \dot{X} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & -1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \ddot{X} \\ &+ \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & -1 \\ 0 & 0 \end{bmatrix} \mathbf{r} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & -1 \end{bmatrix} \mathcal{F} \\ &= W_1 X + W_2 \dot{X} + W_3 \ddot{X} + W_4 \mathbf{r} + W_5 \mathcal{F} \end{aligned} \tag{37}$$

Our goal is to minimize $\mathbf{e}$ by finding proper control gains such that the overall system is stable. The following H_∞ performance requirement is needed.

$$\int_0^\infty \mathbf{e}^T(\eta)\mathbf{e}(\eta) < \Upsilon^2 \int_0^\infty \mathcal{F}^T(\eta)\mathcal{F}(\eta) + \ddot{X}^T(\eta)\ddot{X}(\eta) d\eta \tag{38}$$

Consider the following Lyapunov funcations as $V = V_a + V_b + V_c + V_d$, where

$$V_a = \mathbf{r}_m^T \mathbb{P}_m \mathbf{r}_m + \mathbf{r}_s^T \mathbb{P}_s \mathbf{r}_s \tag{39}$$

$$V_b = \int_{t-\underline{T}_f}^t \mathbf{r}_m^T(\eta) \mathbb{Q}_m \mathbf{r}_m(\eta) d\eta + \int_{t-\underline{T}_b}^t \mathbf{r}_s^T(\eta) \mathbb{Q}_s \mathbf{r}_s(\eta) d\eta \tag{40}$$

$$\begin{aligned} V_c = &\int_{-\overline{T}_f}^0 \int_{t+\theta}^t \dot{X}_m^T(\eta) \mathbb{O}_m \dot{X}_m(\eta) d\eta \\ &+ \int_{-\overline{T}_b}^0 \int_{t+\theta}^t \dot{X}_s^T(\eta) \mathbb{O}_s \dot{X}_s(\eta) d\eta \end{aligned} \tag{41}$$

$$\begin{aligned} V_d = &\int_{-\overline{T}_f}^0 \int_{t+\theta}^t \ddot{X}_m^T(\eta) \mathbb{B}_m \ddot{X}_m(\eta) d\eta \\ &+ \int_{-\overline{T}_b}^0 \int_{t+\theta}^t \ddot{X}_s^T(\eta) \mathbb{B}_s \ddot{X}_s(\eta) d\eta \end{aligned} \tag{42}$$

where $\mathbb{P}_m > 0$,$\mathbb{P}_s > 0$,$\mathbb{Q}_m > 0$,$\mathbb{Q}_s > 0$,$\mathbb{B}_m > 0$,$\mathbb{B}_s > 0$. By applying the above two lemmas, we can achieve the follows

$$\begin{aligned} \dot{V}_a \leq 2\mathbf{r}^T \mathbb{P} (&\overline{\mathscr{A}}_1 \mathbf{r} + \overline{\mathscr{A}}_2 \dot{X}(t-T) + \overline{\mathscr{A}}_3 X(t-T) + \overline{\mathscr{A}}_4 X \\ &+ \mathscr{A}_5 \mathcal{F}) \end{aligned} \tag{43}$$

$$\dot{V}_b \leq \mathbf{r}^T \mathbb{Q} \mathbf{r} - \mathbf{r}^T(t-T) \mathcal{H} \mathbb{Q} \mathbf{r}(t-T) \tag{44}$$

$$\begin{aligned} &\dot{V}_c \leq \overline{T} \dot{X}^T \mathbb{O} \dot{X} - (X - X(t-T))^T U_1 \mathbb{O} (X - X(t-T)) \\ &(X(t-T) - X(t-\overline{T}))^T U_2 \mathbb{O} (X(t-T) - X(t-\overline{T}) \end{aligned} \tag{45}$$

$$\dot{V}_d \leq \overline{T}\ddot{X}^T\mathbb{B}\ddot{X} - (\dot{X} - \dot{X}(t-T))^T U_1\mathbb{B}(\dot{X} - \dot{X}(t-T)) \\ (\dot{X}(t-T) - \dot{X}(t-\overline{T}))^T U_2\mathbb{B}(\dot{X}(t-T) - \dot{X}(t-\overline{T}) \tag{46}$$

where $\overline{\mathscr{A}}_1 = \begin{bmatrix} -\overline{\mathcal{B}}_v M_s^{-1} k_{s1} & 0 \\ 0 & -M_m^{-1}\mathcal{K}_m \end{bmatrix}$. $\overline{\mathcal{B}} \leq \mathcal{B}$ is a constant matrix. $\overline{\mathscr{A}}_2 = \begin{bmatrix} 0 & \Lambda_s\overline{\mathcal{H}}_{s1} \\ \Lambda_m\overline{\mathcal{H}}_{m1} & 0 \end{bmatrix}$ with $\overline{\mathcal{H}}_{s1} \geq \mathcal{H}_{s1}$, and $\overline{\mathcal{H}}_m \geq \mathcal{H}_m$, $\overline{\mathscr{A}}_3 = \begin{bmatrix} 0 & \Lambda_s\overline{\mathcal{H}}_{s2} \\ \Lambda_m & 0 \end{bmatrix}$ with $\overline{\mathcal{H}}_{s2} \geq \mathcal{H}_{s2}$, $\overline{\mathscr{A}}_4 = \begin{bmatrix} \Lambda_s & 0 \\ 0 & \Lambda_m\overline{\mathcal{H}}_{m2} \end{bmatrix}$ with $\overline{\mathcal{H}}_{m2} \geq \mathcal{H}_{m2}$. $\mathbb{P} = diag([\mathbb{P}_m, \mathbb{P}_s])$, $\mathbb{O} = diag([\mathbb{O}_m, \mathbb{O}_s])$, $\mathbb{Q} = diag([\mathbb{Q}_m, \mathbb{Q}_s])$, $\mathbb{B} = diag([\mathbb{B}_m, \mathbb{B}_s])$, $\mathcal{H} = diag([1-\overline{d}_f, 1-\overline{d}_b])$, $U_1 = diag([\frac{2\overline{T}_f - \underline{T}_f}{T_f^2}, \frac{2\overline{T}_b - \underline{T}_b}{T_b^2}])$, $U_2 = diag([\frac{\overline{T}_f + \underline{T}_f}{T_f^2}, \frac{\overline{T}_b + \underline{T}_b}{T_b^2}])$.

From (36), we can derive $\mathscr{A}_1\mathbf{r} + \mathscr{A}_2\dot{X}(t-T) + \mathscr{A}_3X(t-T) + \mathscr{A}_4X + \mathscr{A}_5\mathcal{F} - \dot{\mathbf{r}} = 0$.

Then set $\mathscr{N} = [\mathcal{N}_1, 0, 0, 0, 0, 0, 0, 0, 0, \mathcal{N}_2]^T$. The following equation can be derived

$$\chi_1^T\mathscr{N}\mathscr{L}\chi + \chi_1^T\mathscr{N}\mathscr{A}_2\dot{X}(t-T) + \chi_1^T\mathscr{N}\mathscr{A}_3X(t-T) \\ + \chi_1^T\mathscr{N}\mathscr{A}_4X + \chi_1^T\mathscr{N}\mathscr{A}_5\mathcal{F} = 0 \tag{47}$$

where $\chi_1 = [\mathbf{r}, \mathbf{r}(t-T), X, X(t-T), X(t-\overline{T}), \dot{q}, \dot{q}(t-T), \ddot{q}, \dot{\mathbf{r}}]^T$ and $\mathscr{L} = [\mathscr{A}_1, 0, 0, 0, 0, 0, 0, 0, 0, -I]$. Then, $\dot{V}$ can be rewritten as

$$\dot{V} \leq \dot{V}_a + \dot{V}_b + \dot{V}_c + \dot{V}_d + 2\chi_1^T\mathscr{N}\mathscr{L}\chi + (\overline{\mathscr{A}}_2 + \overline{\mathscr{A}}_3 + \overline{\mathscr{A}}_4) \\ \chi_1^T\mathscr{N}\mathscr{N}^T\chi_1 + \overline{\mathscr{A}}_2\dot{X}^T(t-T)\dot{X}(t-T) \\ + \overline{\mathscr{A}}_3X^T(t-T)X(t-T) + \overline{\mathscr{A}}_4X^TX \\ \leq \chi_1^T\Xi_1\chi_1 + 2\mathbf{r}^T\mathbb{P}\mathscr{A}_5\mathcal{F} + 2\chi_1^T\mathscr{N}\mathscr{A}_5\mathcal{F} \tag{48}$$

with the matrix Ξ_1 written as (49)

$$\Xi_1 = \begin{bmatrix} \Xi_{11} & * & * & * & * & * & * & * & * & * \\ 0 & \Xi_{22} & * & * & * & * & * & * & * & * \\ \Xi_{31} & 0 & \Xi_{33} & * & * & * & * & * & * & * \\ \Xi_{41} & 0 & \Xi_{43} & \Xi_{44} & * & * & * & * & * & * \\ 0 & 0 & 0 & \Xi_{54} & \Xi_{55} & * & * & * & * & * \\ 0 & 0 & 0 & 0 & 0 & \Xi_{66} & * & * & * & * \\ \Xi_{71} & 0 & 0 & 0 & 0 & \Xi_{76} & \Xi_{77} & * & * & * \\ 0 & 0 & 0 & 0 & 0 & 0 & \Xi_{87} & \Xi_{88} & * & * \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \Xi_{99} & * \\ \Xi_{101} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \Xi_{1010} \end{bmatrix} \tag{49}$$

where $\Xi_{11} = 2\mathbb{P}\overline{\mathscr{A}}_1 + \mathbb{Q} + 2\mathcal{N}_1\mathscr{A}_1 + (\overline{\mathscr{A}}_2 + \overline{\mathscr{A}}_3 + \overline{\mathscr{A}}_4)\mathcal{N}_1^T\mathcal{N}_1$, $\Xi_{31} = \mathbb{P}\mathscr{A}_4$, $\Xi_{41} = \mathbb{P}\mathscr{A}_3$, $\Xi_{71} = \mathbb{P}\overline{\mathscr{A}}_2$, $\Xi_{101} = \mathcal{N}_2^T\overline{\mathscr{A}}_1 - \mathcal{N}_1^T + (\overline{\mathscr{A}}_2 + \overline{\mathscr{A}}_3 + \overline{\mathscr{A}}_4)\mathcal{N}_1^T\mathcal{N}_2$, $\Xi_{22} = -\mathcal{H}\mathbb{Q}$, $\Xi_{33} = -U_1\mathbb{O}$, $\Xi_{43} = U_1\mathbb{O}$, $\Xi_{44} = -U_1\mathbb{O} - U_2\mathbb{O}$, $\Xi_{54} = U_2\mathbb{O}$, $\Xi_{55} = -U_2\mathbb{O}$, $\Xi_{66} = \overline{T}\mathbb{O} - U_1\mathbb{B}$, $\Xi_{76} = -U_1\mathbb{B}$, $\Xi_{77} = -U_1\mathbb{B} - U_2\mathbb{B}$, $\Xi_{87} = U_2\mathbb{B}$, $\Xi_{88} = -U_2\mathbb{B}$, $\Xi_{99} = \overline{T}\mathbb{B}$, $\Xi_{1010} = -\mathcal{N}_2^T - \mathcal{N}_2 + (\overline{\mathscr{A}}_2 + \overline{\mathscr{A}}_3 + \overline{\mathscr{A}}_4)\mathcal{N}_2^T\mathcal{N}_2$.

Adding $\mathbf{e}^T\mathbf{e} - \Upsilon^2\mathcal{F}^T\mathcal{F} - \ddot{X}^T\ddot{X}$ to both sides of (48) yields

$$\dot{V} + \mathbf{e}^T\mathbf{e} - \Upsilon^2\mathcal{F}^T\mathcal{F} - \ddot{X}^T\ddot{X} \leq \chi_1^T\Xi_1\chi_1 + 2\mathbf{r}^T\mathbb{P}\mathscr{A}_5\mathcal{F} \\ + 2\chi_1^T\mathscr{N}\mathscr{A}_5\mathcal{F} + \mathbf{e}^T\mathbf{e} - \Upsilon^2\mathcal{F}^T\mathcal{F} - \ddot{X}^T\ddot{X} \\ \leq \chi_2^T\Xi_2\chi_2 \tag{50}$$

where $\chi_2 = [\chi_1^T, \mathcal{F}]$, Ξ_2 is derived by using Schur compliments as (51), where $\Xi_{11} = 2\mathbb{P}\overline{\mathscr{A}}_1 + \mathbb{Q} + 2\mathcal{N}_1\mathscr{A}_1$, $\Xi_{99} = \overline{T}\mathbb{B} - \Upsilon^2 I$, $\Xi_{101} = \mathcal{N}_2^T\overline{\mathscr{A}}_1 - \mathcal{N}_1^T$ $\Xi_{1010} = -\mathcal{N}_2^T - \mathcal{N}_2$, $\Xi_{111} = \mathbb{P}\mathscr{A}_5 + \mathcal{N}_1\mathscr{A}_5$, $\Xi_{1110} = \mathcal{N}_2\mathscr{A}_5$ $\Xi_{1111} = -\Upsilon^2 I$, $\Xi_{121} = W_4$, $\Xi_{123} = W_1$, $\Xi_{126} = W_2$, $\Xi_{129} = W_3$, $\Xi_{1211} = W_5$, $\Xi_{1212} = -I$, $\Xi_{131} = \mathcal{N}_1$ $\Xi_{1310} = \mathcal{N}_2$, $\Xi_{1313} = -(\overline{\mathscr{A}}_2 + \overline{\mathscr{A}}_3 + \overline{\mathscr{A}}_4)I$. Accordingly, if there exist Matrices that $\mathbb{P} > 0$, $\mathbb{Q} > 0$, $\mathbb{O} > 0$, $\mathbb{B} > 0$, $\mathcal{N}_1 > 0$, $\mathcal{N}_2 > 0$ such that Linear Matrix Inequality (LMI) holds, the overall bilateral teleoperation system is asymptotically stable. By using the LMI toolbox in Matlab to solve the inequality (51), γ is calculated to be $3.7190e-4$, which is small enough to guanratee the system stability.

B. Method for the Slave Tool Effector Parallel to the Ground

Create a vector that is aligned with the coordinate of the slave tip as $\mathcal{V}_{tool} = [x_{tool}, 0, 0]^T$. Then transfer this vector to the base coordinate by using the transformation matrix of the robot kinematics as

$$\begin{bmatrix} X_{base} \\ Y_{base} \\ Z_{base} \\ 1 \end{bmatrix} = \begin{bmatrix} R_{rot} & {}^tX_s \\ 000 & 1 \end{bmatrix} \begin{bmatrix} x_{tool} \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{52}$$

where R_{rot} is the rotation matrix.

Since the X axis of the slave tip coordinate is required to be parallel to the surface constituted by the X axis and Y axis of the slave base coordinate, Z_{base} needs to equal to ${}^tX_{sz}$. Accordingly, we can further derive

$$\begin{aligned} & r_{31}x_{tool} + 0 + 0 + {}^tX_{sz} = {}^tX_{sz} \\ \Rightarrow \quad & r_{31} = 0 \end{aligned} \tag{53}$$

where r_{31} is the first element of the third row of R_{rot}. Inside r_{31}, we want all the first five joints of the slave robot keep their current position and the only one that needs to be regulated is the sixth joints (the joint for the end effector). Based on (53), we can derive the reference joint position q_{r6}. Finally, letting the joint position of the sixth joint q_{s6} closely track q_{s6} can guarantee the slave end effector to be always parallel to the ground.

REFERENCES

[1] Q. Liao, D. Sun, and H. Andreasson, "Point set registration for 3d range scans using fuzzy cluster-based metric and efficient global optimization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, early access, 2020, DOI: 10.1109/TPAMI.2020.2978477.

[2] H. Dong, E. Asadi, G. Sun, D. K. Prasad, and I.-M. Chen, "Real-time robotic manipulation of cylindrical objects in dynamic scenarios through elliptic shape primitives," *IEEE Transactions on Robotics*, vol. 35, no. 1, pp. 95–113, 2018.

$$\Xi_2 = \begin{bmatrix} \Xi_{11} & * & * & * & * & * & * & * & * & * & 8 & * & * \\ 0 & \Xi_{22} & * & * & * & * & * & * & * & * & * & * & * \\ \Xi_{31} & 0 & \Xi_{33} & * & * & * & * & * & * & * & * & * & * \\ \Xi_{41} & 0 & \Xi_{43} & \Xi_{44} & * & * & * & * & * & * & * & * & * \\ 0 & 0 & 0 & \Xi_{54} & \Xi_{55} & * & * & * & * & * & * & * & \\ 0 & 0 & 0 & 0 & 0 & \Xi_{66} & * & * & * & * & * & * & * \\ \Xi_{71} & 0 & 0 & 0 & 0 & \Xi_{76} & \Xi_{77} & * & * & * & * & * & * \\ 0 & 0 & 0 & 0 & 0 & 0 & \Xi_{87} & \Xi_{88} & * & * & * & * & * \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \Xi_{99} & * & * & * & * \\ \Xi_{101} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \Xi_{1010} & * & * & * \\ \Xi_{111} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \Xi_{1110} & \Xi_{1111} & * & * \\ \Xi_{121} & 0 & \Xi_{123} & 0 & 0 & \Xi_{126} & 0 & 0 & \Xi_{129} & 0 & \Xi_{1211} & \Xi_{1212} & * \\ \Xi_{131} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \Xi_{1310} & 0 & 0 & \Xi_{1313} \end{bmatrix} < 0 \quad (51)$$

[3] A. Zeng, S. Song, K.-T. Yu, E. Donlon, F. R. Hogan, M. Bauza, D. Ma, O. Taylor, M. Liu, E. Romo, *et al.*, "Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching," in *2018 IEEE international conference on robotics and automation (ICRA)*, IEEE, 2018, pp. 1–8.

[4] O. Koç, G. Maeda, and J. Peters, "Optimizing the execution of dynamic robot movements with learning control," *IEEE Transactions on Robotics*, vol. 35, no. 4, pp. 909–924, 2019.

[5] D. Nicolis, M. Palumbo, A. M. Zanchettin, and P. Rocco, "Occlusion-free visual servoing for the shared autonomy teleoperation of dual-arm robots," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 796–803, 2018.

[6] A. Dietrich and C. Ott, "Hierarchical impedance-based tracking control of kinematically redundant robots," *IEEE Transactions on Robotics*, 2019.

[7] D. Heck, A. Saccon, R. Beerens, H. Nijmeijer, *et al.*, "Direct force-reflecting two-layer approach for passive bilateral teleoperation with time delays," *IEEE Transactions on Robotics*, vol. 34, no. 1, pp. 194–206, 2018.

[8] S. A. Deka, D. M. Stipanović, and T. Kesavadas, "Stable bilateral teleoperation with bounded control," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 6, pp. 2351–2360, 2018.

[9] D. Sun, Q. Liao, and H. Ren, "Type-2 fuzzy logic based time-delayed shared control in online-switching tele-operated and autonomous systems," *Robotics and Autonomous Systems*, vol. 101, pp. 138–152, 2018.

[10] Y. Li, K. Liu, W. He, Y. Yin, R. Johansson, and K. Zhang, "Bilateral teleoperation of multiple robots under scheduling communication," *IEEE Transactions on Control Systems Technology*, early access, 2019, DOI: 10.1109/TCST.2019.2923788.

[11] D. Sun, F. Naghdy, and H. Du, "Enhancing flexibility of the dual-master-dual-slave multilateral teleoperation system," in *2015 IEEE Conference on Control Applications (CCA)*, IEEE, 2015, pp. 300–305.

[12] Y. Yang, C. Hua, and X. Guan, "Multi-manipulators coordination for bilateral teleoperation system using fixed-time control approach," *International Journal of Robust and Nonlinear Control*, vol. 28, no. 18, pp. 5667–5687, 2018.

[13] D. Sun, F. Naghdy, and H. Du, "Time domain passivity control of time-delayed bilateral telerobotics with prescribed performance," *Nonlinear dynamics*, vol. 87, no. 2, pp. 1253–1270, 2017.

[14] M. Laghi, A. Ajoudani, M. G. Catalano, and A. Bicchi, "Unifying bilateral teleoperation and tele-impedance for enhanced user experience," *The International Journal of Robotics Research*, vol. 39, no. 4, pp. 514–539, 2020.

[15] A. Dong, Z. Du, and Z. Yan, "A sensorless interaction forces estimator for bilateral teleoperation system based on online sparse gaussian process regression," *Mechanism and Machine Theory*, vol. 143, p. 103 620, 2020.

[16] D. Sun, Q. Liao, T. Stoyanov, A. Kiselev, and A. Loutfi, "Bilateral telerobotic system using type-2 fuzzy neural network based moving horizon estimation force observer for enhancement of environmental force compliance and human perception," *Automatica*, vol. 106, pp. 358–373, 2019.

[17] D. Sun, Q. Liao, and H. Ren, "Type-2 fuzzy modeling and control for bilateral teleoperation system with dynamic uncertainties and time-varying delays," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 1, pp. 447–459, 2018.

[18] Z. Wang, H.-K. Lam, B. Xiao, Z. Chen, B. Liang, and T. Zhang, "Event-triggered prescribed-time fuzzy control for space teleoperation systems subject to multiple constraints and uncertainties," *IEEE Transactions on Fuzzy Systems*, early access, 2020, DOI: 10.1109/TFUZZ.2020.3007438.

[19] D. Rakita, B. Mutlu, M. Gleicher, and L. M. Hiatt, "Shared control–based bimanual robot manipulation," *Science Robotics*, vol. 4, no. 30, eaaw0955, 2019.

[20] K. Watanabe, T. Kanno, K. Ito, and K. Kawashima, "Single-master dual-slave surgical robot with automated relay of suture needle," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 8, pp. 6343–6351, 2017.

[21] C. Yang, J. Luo, C. Liu, M. Li, and S.-L. Dai, "Haptics electromyography perception and learning enhanced intelligence for teleoperated robot," *IEEE Transactions on Automation Science and Engineering*, vol. 16, no. 4, pp. 1512–1521, 2018.

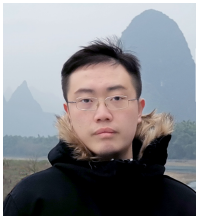
[22] C. Yang, C. Zeng, Y. Cong, N. Wang, and M. Wang, "A learning framework of adaptive manipulative skills from human to robot," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 2, pp. 1153–1161, 2018.

[23] Z. Lu, P. Huang, and Z. Liu, "Relative impedance-based internal force control for bimanual robot teleoperation with varying time delay," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 1, pp. 778–789, 2019.

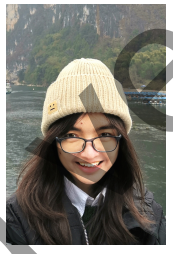
[24] D. Feth, A. Peer, and M. Buss, "Enhancement of multi-user teleoperation systems by prediction of dyadic haptic interaction," in *Experimental Robotics*, Springer, 2014, pp. 855–869.

[25] F. Abi-Farraj, C. Pacchierotti, O. Arenz, G. Neumann, and P. R. Giordano, "A haptic shared-control architecture for guided multi-target robotic grasping," *IEEE transactions on haptics*, 2019.

[26] R. Rahal, G. Matarese, M. Gabiccini, A. Artoni, D. Prattichizzo, P. R. Giordano, and C. Pacchierotti, "Caring about the human operator: Haptic shared control for enhanced user comfort in robotic telemanipulation," *IEEE Transactions on Haptics*, vol. 13, no. 1, pp. 197–203, 2020.
[27] D. Sun, Q. Liao, and A. Loutfi, "Single master bimanual teleoperation system with efficient regulation," *IEEE Transactions on Robotics*, early access, 2020, DOI: 10.1109/TRO.2020.2973099.
[28] T. Mizoguchi, T. Nozaki, and K. Ohnishi, "Stiffness transmission of scaling bilateral control system by gyrator element integration," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 2, pp. 1033–1043, 2013.
[29] S. Sakaino, T. Sato, and K. Ohnishi, "Multi-dof micro-macro bilateral controller using oblique coordinate control," *IEEE Transactions on Industrial Informatics*, vol. 7, no. 3, pp. 446–454, 2011.
[30] J. Guo, C. Liu, and P. Poignet, "A scaled bilateral teleoperation system for robotic-assisted surgery with time delay," *Journal of Intelligent & Robotic Systems*, pp. 1–28, 2017.
[31] M. Mehrtash, N. Tsuda, and M. B. Khamesee, "Bilateral macro–micro teleoperation using magnetic levitation," *IEEE/ASME Transactions on Mechatronics*, vol. 16, no. 3, pp. 459–469, 2011.
[32] A. Hace and M. Franc, "Fpga implementation of sliding-mode-control algorithm for scaled bilateral teleoperation," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 3, pp. 1291–1300, 2013.
[33] P. Yi, A. Song, J. Guo, Y. Liu, and G. Jiang, "Delay-dependent stabilization control for asymmetric bilateral teleportation systems with time-varying delays," in *Robotics and Automation Engineering (ICRAE), International Conference on*, IEEE, 2016, pp. 10–16.
[34] U. Ahmad and Y.-J. Pan, "A time domain passivity approach for asymmetric multilateral teleoperation system," *IEEE Access*, vol. 6, pp. 519–531, 2018.
[35] T. Hilliard and Y.-J. Pan, "Bilateral control and stabilization of asymmetric teleoperators with bounded time-varying delays," *Journal of Dynamic Systems, Measurement, and Control*, vol. 136, no. 5, p. 051 001, 2014.
[36] A. Talasaz, A. L. Trejos, and R. V. Patel, "The role of direct and visual force feedback in suturing using a 7-dof dual-arm teleoperated system," *IEEE transactions on haptics*, vol. 10, no. 2, pp. 276–287, 2016.
[37] I. Nisky, Y. Che, Z. F. Quek, M. Weber, M. H. Hsieh, and A. M. Okamura, "Teleoperated versus open needle driving: Kinematic analysis of experienced surgeons and novice users," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2015, pp. 5371–5377.
[38] F. Stulp, E. A. Theodorou, and S. Schaal, "Reinforcement learning with sequences of motion primitives for robust manipulation," *IEEE Transactions on robotics*, vol. 28, no. 6, pp. 1360–1370, 2012.
[39] D. Sun, F. Naghdy, and H. Du, "Neural network-based passivity control of teleoperation system under time-varying delays," *IEEE transactions on cybernetics*, vol. 47, no. 7, pp. 1666–1680, 2016.
[40] Y.-C. Liu and M.-H. Khong, "Adaptive control for nonlinear teleoperators with uncertain kinematics and dynamics," *IEEE/ASME Transactions on Mechatronics*, vol. 20, no. 5, pp. 2550–2562, 2015.
[41] O. Kwon, M.-J. Park, S.-M. Lee, and J. H. Park, "Augmented lyapunov–krasovskii functional approaches to robust stability criteria for uncertain takagi–sugeno fuzzy systems with time-varying delays," *Fuzzy Sets and Systems*, vol. 201, pp. 1–19, 2012.



Da Sun received B.Eng. and Ph.D. degrees in mechatronics from the University of Wollongong, Wollongong, NSW, Australia, in 2012 and 2016, respectively. From 2016 to 2017, he was a Research Fellow with the National University of Singapore, Singapore. He is currently a researcher at the Center for Applied Autonomous Sensor Systems (AASS) at Örebro University, Örebro, Sweden. His research interests include robotics learning, modelling, and control theory.



Qianfang Liao received a B.Eng. degree in automation from Wuhan University, Wuhan, China, in 2006, an M.Eng. degree in automation from Shanghai Jiao Tong University, Shanghai, China, in 2009, and a Ph.D. from the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, in 2015. Currently, she is a researcher with Örebro University, Örebro, Sweden. Her research interests include fuzzy modeling, control theory, and computer vision.

Amy Loutfi is a professor in information technology at Örebro University with the Center for Applied Autonomous Sensor Systems (AASS), Sweden. Her general interests are in the integration of artificial intelligence with autonomous systems, and over the years, she has looked into applications where robots closely interact with humans in both industrial and domestic environments. She received her Ph.D. degree in computer science in 2006; her thesis focused on the integration of artificial olfaction into robotic and intelligent systems to enable good human-robot interactions. Since then, she has examined various cases of human-robot interactions, including applications of telepresence.