Mixed Reality Teleoperation Assistance for Direct Control of Humanoids

Luigi Penco¹, Kazuhiko Momose², Stephen McCrory^{1,3}, Dexton Anderson¹, Nicholas Kitchel¹, Duncan Calvert^{1,3}, Robert J. Griffin^{1,3}

Abstract—Teleoperation plays a crucial role in enabling robot operations in challenging environments, yet existing limitations in effectiveness and accuracy necessitate the development of innovative strategies for improving teleoperated tasks. This article introduces a novel approach that utilizes mixed reality and assistive autonomy to enhance the efficiency and precision of humanoid robot teleoperation. By leveraging Probabilistic Movement Primitives, object detection, and Affordance Templates, the assistance combines user motion with autonomous capabilities, achieving task efficiency while maintaining humanlike robot motion. Experiments and feasibility studies on the Nadia robot confirm the effectiveness of the proposed framework. Supplementary video available at [https://youtu.be/oN-FD6YnF2c.](https://youtu.be/oN-FD6YnF2c)

Index Terms—Virtual Reality and Interfaces, Humanoid Robot Systems, Telerobotics and Teleoperation, Whole-Body Motion Planning and Control, Learning from Demonstration

I. INTRODUCTION

SEVERAL of the most compelling instances of humanoids
performing valuable tasks have been through teleoperation
[1] Hamman density were of appearable in this demain tele performing valuable tasks have been through teleoperation [\[1\]](#page-7-0). However, despite years of research in this domain, teleoperated humanoid robots continue to show substantial limitations when executing tasks under user control. Traditional direct control of robots through the user's own motions offers an intuitive approach for interacting with the remote environment [\[2\]](#page-7-1), [\[3\]](#page-7-2), [\[1\]](#page-7-0). However, differences in kinematics between humans and robots can lead to errors and multiple attempts for task completion, making this approach not very effective and time-consuming, and even lead to operator frustration.

Our work addresses the challenge of improving the direct control of humanoids in teleoperation by introducing autonomous assistance features in mixed reality. The goal is to enhance the efficiency and accuracy of teleoperated tasks. To achieve this, we employ Probabilistic Movement Primitives (ProMPs) [\[4\]](#page-7-3), which are widely used for modular movement representation and generation. These ProMPs

¹The authors are with the Florida Institute for Human and Machine Cognition, 40 S Alcaniz St, Pensacola, FL 32502, United States lpenco@ihmc.org

³The authors are with the University of West Florida, 11000 University Pkwy, Pensacola, FL 32514, United States rgriffin@uwf.edu

Digital Object Identifier (DOI): see top of this page.

Fig. 1. With the teleoperation assistance we propose, the user serves as a highlevel guide, directing the robot's autonomous task execution through a mixed reality interface (top left) and specifying the manner in which the task should be carried out. In the given example, the user merely initiates a portion of a punching motion, specifically indicating an uppercut technique. Recognizing this, the robot discerns that it is tasked with a punching action and, among the various techniques available, it should execute an uppercut. The user does not need to focus on the precision of the subsequent motion, as the initial input is sufficient for the robot to complete the task accurately.

are conditioned on the operator's input, allowing adaptation to the user's preferred way of performing the task while maintaining similarity to the demonstrations. Additionally, we utilize information from the robot's sensors to provide context and precision to the motion primitives. The physical interaction with objects is governed by Affordance Templates (ATs) [\[5\]](#page-7-4), ensuring accurate and controlled manipulation while reducing the risk of errors or unintended consequences during physical interaction. In order to observe, predict and direct the assistance, we developed a mixed reality interface, based on Coactive Design principles [\[6\]](#page-7-5). Using mixed reality has the benefit of providing an immersive experience for the operator as also shown in [\[7\]](#page-7-6).

The need of such teleoperation assistance was vividly highlighted during the finals of the ANA Avatar XPRIZE [\[8\]](#page-7-7), [\[9\]](#page-7-8), [\[10\]](#page-7-9), a four-year global competition on robot teleoperation that ended in November 2022. Operators struggled to effectively and accurately control the robots, even for seemingly simple tasks like grasping a bottle. Multiple attempts were often

Manuscript received: August, 14, 2023; Revised October, 12, 2023; Accepted December, 17, 2023.

This paper was recommended for publication by Editor Jee-Hwan Ryu upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by ONR Grant N00014-22-1-2593 and DAC Cooperative Agreement W911NF-22-2-0201.

²The author is with the Florida Institute of Technology, 150 W University Blvd, Melbourne, FL 32901, United States kmomose@my.fit.edu

required, showcasing the difficulties in achieving satisfactory outcomes solely through direct user control — especially when considering that robots can perform similar tasks autonomously with relative ease.

It is important to note that full direct control is typically achieved by projecting the RGB images from the robot camera to the Virtual Reality (VR) headset [\[9\]](#page-7-8), [\[10\]](#page-7-9), but this approach often fails to provide sufficient situational awareness [\[11\]](#page-7-10) to the user. This was also demonstrated in the ANA Avatar XPRIZE finals, where judge operators frequently encountered confusion regarding the robot's location within the environment. Hence, a combination of autonomy and mixed reality technology is crucial in bridging these gaps and enabling robots for real-world applications.

A recent and promising approach to addressing the direct control issues is shared-control, which integrates robot assistance with user input to facilitate the completion of tasks [\[12\]](#page-7-11). For instance, Rakita *et al.* [\[13\]](#page-7-12) implemented a shared-control approach in which they teleoperated the upper body of a humanoid robot. By maintaining a constant offset between the end effectors or limiting the motion of the endeffector that holds the object, the system provides on-the-fly assistance and helps the user perform bimanual tasks more easily. Similarly, in [\[14\]](#page-7-13), Rahal *et al.* designed a shared control approach to assist the human operator by enforcing different nonholonomic-like constraints representative of the cutting kinematics.

In many shared-control frameworks, the user provides an input that allows the robot to consider the human's intent and to assist in the task by adjusting its motion or executing a pre-optimized version of that motion. To determine the final reference for the robot's motion, a policy blending mechanism that governs the balance between the user input and the enhanced robot motion is employed [\[12\]](#page-7-11). The blending policy is usually the distance to the goal. The closer the robot gets to a predicted goal, the more likely it is that the predicted goal is the correct one, and thus, the robot input is favored over the user's.

Approaches have also been proposed that consider unknown goals. For instance, in the case of [\[15\]](#page-7-14), a POMDP-based strategy has been employed to provide assistance, aiming to minimize the expected cost-to-go. It is worth highlighting that this particular approach is predominantly designed for the control of goal-oriented components of the robot, such as its end-effectors. Consequently, its direct applicability to the control of other robot components, such as adjusting the orientation of the chest and forearms as needed for wholebody tasks may be limited. Alternatively, [\[16\]](#page-7-15) presents a novel approach leveraging the Koopman operator for learning the dynamics of a machine and enhancing human-machine interaction. However, the application and scalability of this system to more complex systems, such as humanoid robots, remain unvalidated and appear to be challenging due to the increased complexity and unpredictability inherent in such systems.

In haptic-based shared-control approaches, a different strategy is employed. Here, only the user's input actuates the robot, but haptic information is used to guide the user during task execution [\[14\]](#page-7-13), [\[17\]](#page-7-16). By incorporating haptic feedback, the system can provide the user with tactile cues or force feedback, improving the user's understanding of the task and making the teleoperation process more intuitive and effective.

In line with the efforts that have been made in humanoid teleoperation, our work continues to advance these endeavors. By integrating predictive modeling techniques, we are poised to proactively gauge the user's intended motion, tailoring the robot behavior to both fit the user input and the pose of manipulated objects in the environment. By leveraging the strengths of both the user and the robot, we do not just enhance task efficiency; we also maintain the human-likeness of the robot's motion. Our work is validated through experiments and feasibility studies on the Nadia humanoid robot.

II. ASSISTIVE AUTONOMY

Our assistive autonomy seeks to improve robot teleoperation by enabling more precise and adaptive robot movements. This approach ensures fidelity to the demonstrated actions while compensating for inaccuracies in user input, both during the demonstrations and during actual operation. The proposed method utilizes ProMPs along with ATs when physical interaction with the object is required.

The system follows a series of operations as shown in Fig. [2.](#page-2-0) The learning phase involves recording few demonstrations (less than 30) of the task performed in different ways by a human operator in a simulated environment. ProMPs are learned from these demonstrations. During teleoperation, the system recognizes the current task by identifying the corresponding ProMP, aided by semantic object information. The system then updates the posterior distribution of the ProMPs using the initial user input and the estimated object pose. The mean trajectory of the updated ProMP is used as a reference for the robot controller. If an affordance is available for that object, a blending mechanism is used to achieve smooth transitions between ProMP-generated motions and ATs. The generated trajectories are then tracked by the robot controller (Section [II-H\)](#page-3-0), which calculates the appropriate joint commands. Both the object pose conditioning and the AT integration make it possible to transfer the learned ProMPs to the real world without any real-world training data.

A. Probabilistic Movement Primitives (ProMPs)

ProMPs [\[4\]](#page-7-3) are a probabilistic model used to represent trajectory distributions. These models describe the time-varying mean and variance of trajectories using basis functions. Each trajectory is parameterized by a weight vector $w \in \mathbb{R}^m$. Given the weight vector, the probability $\boldsymbol{\xi}(t)$ of observing a trajectory *y* is modeled using a linear basis function model:

$$
\boldsymbol{\xi}_y(t) = \boldsymbol{\Phi}_t \boldsymbol{w} + \boldsymbol{\varepsilon}_\xi, \tag{1}
$$

where Σ_{ξ} represents the observation noise variance, and $\boldsymbol{\varepsilon}_{\xi} \sim$ $\mathcal{N}(0, \Sigma_{\xi})$ is the trajectory noise. $\Phi_t \in \mathbb{R}^{n \cdot m}$ is a block-diagonal matrix containing the *m* basis functions for each dimension *n*. The distribution $p(\mathbf{w}; \boldsymbol{\theta})$ over the weight vector **w** is Gaussian, with parameters $\boldsymbol{\theta} = \boldsymbol{\mu}_{w}$, $\boldsymbol{\Sigma}_{w}$ specifying the mean and variance of *w*.

Fig. 2. Flowchart of the assistive autonomy. During the training phase, the human operator teleoperates the robot in simulation, and performs the tasks in different ways. A ProMP is learned for every task. When teleoperating the robot in real-time, the ProMPs are used to generate the assistive robot motion: (1) the system recognizes the current task, with the help of context given by the object detection; (2) it updates the ProMP according to the initially observed user input and object pose; (3) if an affordance is available for that object, a blending mechanism is used to to achieve smooth transitions between ProMP-generated motions and ATs.

B. Learning ProMPs from Demonstrations

In the learning phase of ProMPs, demonstrations are recorded from a human operator teleoperating the robot in simulation to perform the task in different ways. To decouple the movement from the time signal and obtain a durationindependent representation, a phase variable $v \in [0,1]$ is introduced. The modulated trajectories $\xi_i(v)$ are then used to learn a ProMP for the task. The parameters $\boldsymbol{\theta} = \boldsymbol{\mu}_w, \boldsymbol{\Sigma}_w$ of the ProMP are estimated using a maximum likelihood estimation algorithm. The weight vectors w_i for each demonstration *i* are computed using linear ridge regression, as shown in [\(2\)](#page-2-1), where the ridge factor λ is typically set to a very small value $(\lambda = 10^{-12}$ in our case):

$$
\mathbf{w}_i = \left(\mathbf{\Phi}_v^\top \mathbf{\Phi}_v + \lambda\right)^{-1} \mathbf{\Phi}_v^\top \mathbf{\xi}_i(v),\tag{2}
$$

Assuming normal distributions $p(\mathbf{w}) \sim \mathcal{N}(\mathbf{\mu}_w, \mathbf{\Sigma}_w)$, the mean μ_w and covariance Σ_w can be computed from the samples *wⁱ* :

$$
\boldsymbol{\mu}_{\mathbf{w}} = \frac{1}{D} \sum_{i=1}^{D} \mathbf{w}_i, \quad \boldsymbol{\Sigma}_{\mathbf{w}} = \frac{1}{D} \sum_{i=1}^{D} (\mathbf{w}_i - \boldsymbol{\mu}_{\mathbf{w}}) (\mathbf{w}_i - \boldsymbol{\mu}_{\mathbf{w}})^\top, \quad (3)
$$

where *D* is the number of demonstrations. Each ProMP models the behavior of an individual body part of the robot and is learned in the object frame. This means that the movements are represented and learned relative to the object or task being performed, rather than being tied to a specific robot or coordinate system. This decoupling of the movement from the robot's frame of reference allows for greater flexibility in executing the learned movements in different environments.

C. Recognizing the Motion Primitive

Given the current context (detected object), we determine to which ProMP the present teleoperated motion belongs by minimizing the distance between the initial *nobs* observations (typically corresponding to a fourth or a third of the duration of a given motion) and the ProMP's mean:

$$
\hat{k} = \arg\min_{k \in [1:K]} \left[\sum_{t \in T_{obs}} ||\mathbf{y}(t) - \mathbf{\Phi}_t \boldsymbol{\mu}_{\mathbf{w}_k}|| \right],
$$
\n(4)

where K is the number of tasks in the dataset associated to the detected object (Section [II-E\)](#page-3-1) and $T_{obs} = \{t_1, \ldots, t_{n_{obs}}\}$ is the set of timesteps associated to the *nobs* early observations. While computing \hat{k} , the ProMP is modulated to have a duration equal to the mean duration of the demonstrations. The recognition [\(4\)](#page-2-2) starts whenever a motion is detected, i.e. the derivative of the observed end-effector trajectories exceeds a given threshold, after the user activates the assistance mode (Section [III\)](#page-3-2). This ensures that motion data is not processed as observations used for conditioning in instances where the user remains inactive.

D. ProMPs Conditioning

Once the right ProMP has been identified, we update their posterior distribution to take into account the initial observations from the user input. Conditioning is beneficial for tailoring the motion to accommodate the user's preferred method of task execution. If this step is omitted, the system would default to executing the average value of the learned data, which may not align with the user's intentions. Each ProMP has to be conditioned to reach a certain observed state y_t^* . The conditioning for a given observation $\mathbf{x}_t^* = {\mathbf{y}_t^*, \mathbf{\Sigma}_y^*}$ (with $\mathbf{\Sigma}_y^*$) being the noise in the desired observation) is performed by applying Bayes' theorem:

$$
p(\mathbf{w}_{\hat{k}}|\mathbf{x}_t^*) \propto \mathcal{N}(\mathbf{y}_t^*|\mathbf{\Phi}_t \mathbf{w}_{\hat{k}}, \mathbf{\Sigma}_{\mathbf{y}}^*)p(\mathbf{w}_{\hat{k}}).
$$
 (5)

The conditional distribution of $p(w_k|\mathbf{x}_t^*)$ is Gaussian with mean and variance

$$
\hat{\boldsymbol{\mu}}_{\mathbf{w}_{\hat{k}}} = \boldsymbol{\mu}_{\mathbf{w}_{\hat{k}}} + \boldsymbol{L} (\mathbf{y}_t^* - \boldsymbol{\Phi}_t^\top \boldsymbol{\mu}_{\mathbf{w}_{\hat{k}}}),
$$
\n(6)

$$
\hat{\Sigma}_{w_{\hat{k}}} = \Sigma_{w_{\hat{k}}} - L\Phi_t^\top \Sigma_{w_{\hat{k}}},\tag{7}
$$

where

$$
\mathbf{L} = \mathbf{\Sigma}_{\mathbf{w}_{\hat{k}}} \mathbf{\Phi}_t \left(\mathbf{\Sigma}_{\mathbf{y}}^* + \mathbf{\Phi}_t^\top \mathbf{\Sigma}_{\mathbf{w}_{\hat{k}}} \mathbf{\Phi}_t \right)^{-1} . \tag{8}
$$

and $w_{\hat{k}}$, $\Sigma_{\hat{k}}$ are the weight and covariance of the identified \hat{k} ProMP. In situations where the motion capture system is highly accurate, we can consider Σ_{γ}^{*} to approach zero, signifying that there is very little uncertainty in the observed data.

E. Object Detection and Pose Estimation

In order to accurately perceive and interact with objects, the estimation of their pose is crucial. For object pose estimation, we utilized ArUco markers. This method serves as a temporary solution while the integration of state-of-the-art object detection algorithms in our framework is ongoing. The user interface (UI) displays the estimated pose as an overlaying virtual object, generated from CAD data.

In certain situations, manual override of object pose estimation may be necessary. We achieve this by utilizing the virtual objects in the interface (Section [III\)](#page-3-2). The user can interact with these virtual objects and manually adjust their pose to match the real-world objects based on the point cloud from the depth camera and LIDAR on the robot.

F. Affordance Templates (ATs)

An AT [\[5\]](#page-7-4) is an adjustable pairing of 3D object geometries and sequence of robot actions represented in object-centric coordinates. At run-time, virtual object frames are registered by the user, defining waypoint frames that are converted into robot specific end-effector frames. To facilitate the creation and management of ATs, we have developed our user interface that allows for easy editing, saving, and loading.

G. Blending ProMPs to ATs

The generation of training trajectories for ProMPs does not require high precision. The object pose information ensures accuracy because ProMPs are updated to reach the detected object pose. However, the object's centroid may not align with the desired grasping point encoded in the AT. Hence, transitioning to the AT needs to occur before the ProMP motion is completed. To achieve a smooth transition, a policy blending mechanism adjusts the ProMP-generated motion $y_{P\text{row}}$ by incorporating the first sample from the AT-generated trajectory y_{AT}^0 . This process determines the refined reference for the movement using the following blending equation:

$$
\mathbf{y}' = (1 - \alpha)\mathbf{y}_{ProMP} + \alpha \mathbf{y}_{AT}^0.
$$
 (9)

Here, α represents the blending coefficient, which is defined by the function:

$$
\alpha(x) = \frac{1}{1 + a \ e^{-b(x - c)}}, \quad \alpha \in [0, 1]
$$
 (10)

The value of *x* is given by $x = \frac{i}{N_b}$, where *i* ranges from 0 to N_b , and N_b is the number of blending samples. The values of *a*,*b*, and *c* ∈ R are chosen to ensure that $\alpha(0) = 0$, $\alpha(1) = 1$, and $0.8 < \alpha(0.7) < 0.9$. This allows for an early transition to the affordance, discarding any imprecision present in the final part of the recorded training trajectories.

H. Robot Controller

The robot uses a momentum-based whole-body controller that is framed as a quadratic program (QP) [\[18\]](#page-7-17). The controller's primary task is to track a desired rate of change of momentum, but it can simultaneously track a set of external motion objectives for the robot's pelvis height, chest

Fig. 3. Operator's view in mixed reality while remotely controlling the robot using the proposed teleoperation assistance. The virtual panel provides realtime guidance, presenting the user with essential instructions for each step of the operation. Pre-activation. An object has been detected (highlighted in blue and with an orange arrow) for which the autonomy is available, and the user can activate the teleoperation assistance. Generation. The user can start doing the task and provide initial input to the autonomy that will adapt accordingly. Validation. The user can preview the proposed motion via a ghost robot and spline trajectories, validate it or reject it. Execution. The user can execute the proposed motion via joystick control.

orientation and arm configurations. By solving the QP, the controller produces a joint acceleration vector and contact wrenches. These values are then used to calculate the desired actuator torques through inverse dynamics. In the context of humanoid teleoperation [\[1\]](#page-7-0), references for the tracked body segments are determined using an Inverse Kinematics (IK) QP. This IK QP ensures the feasibility and safety of the specified references [\[19\]](#page-7-18). The output of the Inverse Kinematics QP is then supplied to the momentum-based whole-body controller, further enhancing the overall performance and balance of the robot's movements during teleoperation and assistive autonomy scenarios. For a more in-depth understanding of the whole-body controller, additional detailed information can be found in [\[18\]](#page-7-17).

III. MIXED REALITY INTERFACE

Throughout our work, we have formulated a set of robust principles that inform the design of human-robot operator interfaces. These principles have a central focus on three essential elements: observability, predictability, and directability [\[6\]](#page-7-5). Essentially, the user should be able to observe the robot's current state and behaviors, predict its actions in response to instructions, and direct the robot's next actions.

Our current interface includes a camera view to display the robot's perspective, alongside a created "digital twin" world view. This world view integrates all perception inputs from a depth camera and LIDAR, as well as a third-person view of the robot generated from proprioception sensors and state estimation. This interface provides the user with the option to operate the robot using a mouse and keyboard or via a

Fig. 4. Interdependence analysis of a teleoperated task using the assistive autonomy.

VR headset and controllers. The latter mode, which we call kinematics streaming, allows the operator to directly control the robot's hands (and other parts via VR trackers) and use the VR controller triggers to open and close the grippers. During kinematics streaming, a yellow virtual ghost robot appears, overlaying the robot visualizer model (Fig. [3\)](#page-3-3). This ghost robot serves as a reference, representing the solution found by our IK algorithm based on the current VR input.

Fig. [4](#page-4-0) depicts the interdependence analysis [\[6\]](#page-7-5) highlighting the relationships between the user and the robot during a teleoperation task facilitated by the assistive autonomy and showing how Coactive Design principles govern their interaction. While the user is directly controlling the robot and whenever an object is detected for which the assistive autonomy is available, a virtual object representing the detected object appears in the UI at the estimated location. An information panel also appears on top of the right VR controller (Fig. [3\)](#page-3-3). This panel informs the user that the assistive autonomy is available and can be activated by pressing a dedicated button on the left controller.

Once activated, the panel prompts the user to move and start performing the task (Fig. [3\)](#page-3-3). This allows the robot to analyze the user's initial inputs and determine trajectories for each controlled body part accordingly. At this stage, the robot can either be directly controlled by the user or remain in an idle state. The assistive autonomy processes the motion, identifies the task, and adapts to the user's preferred way of performing it. Following this initial processing, the system presents a preview of the computed motion to the user. Predictability over the robot actions is provided through a visualization of the robot's motion with a gray ghost preview and spline trajectories (Fig. [3\)](#page-3-3). At this point, the user has the option to either accept or reject the suggested motion by pressing a dedicated button on the VR controller. The preview of the planned trajectory determined by the assistive autonomy can be disabled if the user wants to speed up operation.

Upon validation, the user gains control over the robot and can direct the movements along the computed trajectories by tilting the joystick of a controller forward. This enables the user to control the robot's actions frame by frame. In fact, the motion can be paused by stopping the tilting motion at any time. To provide extra safety, a yellow ghost robot model is displayed as feedback on how well the robot is following the reference trajectories. If the operator observes significant deviations between the yellow ghost and the actual

Fig. 5. ProMP-based reference trajectories for reaching the door handle. Left column: learned ProMPs (light thick lines with transparent regions) alongside the corresponding 20 demonstrations (dark thin lines), expressed in the door handle frame. These demonstrations include reaching motions from different approach locations. Right column: updated ProMPs (dark lines) after observing the user input (lighter short lines followed by a vertical gray bar).

robot's movement, they can pause the motion, exit the assistive autonomy mode, or provide a different initial input. Thus, safety is guaranteed by the joystick control (directability) and the visualization of the yellow ghost (observability).

The information panel was designed with key usability heuristics in mind, such as visibility of system status, matching the system with the real world, and recognition over recall [\[20\]](#page-7-19), [\[21\]](#page-7-20). The assistive autonomy mode state is indicated by the color of a small robot icon (green for engaged, white for disengaged), and a horizontal progress bar shows the completion percentage of the task, helping the user observe robot's status (Fig. [3\)](#page-3-3). As the operator tilts the joystick forward, the blue bar expands, and when it reaches the edge, the task is complete, and the robot icon turns white. The panel utilizes representations of real-world VR controller icons for more intuitive control. Additionally, it guides the operator through the steps required to direct the assistive autonomy by highlighting the corresponding buttons or actions in green (Fig. [3\)](#page-3-3).

Fig. 6. ProMP-AT blending policy during teleoperation. The reference trajectories for the robot during the teleoperation process are illustrated. In the first part, the trajectories are the user's input with the robot following their motions. After the user has provided some input, they can stop. The ProMP trajectory is updated according the initial user input, allowing the robot to execute the updated motion. When the ProMP motion reaches its end, a blending mechanism smoothly transitions between the ProMP and the AT trajectories. Finally, the AT motion is executed.

IV. TELEOPERATION ASSISTANCE IN ACTION

A. Experiments

We conducted experiments with Nadia^{[1](#page-5-0)}, our advanced humanoid robot, provided with 31 degrees of freedom and partially powered by hydraulics. Users were equipped with a Valve Index VR headset and controllers to operate the robot. To evaluate the effectiveness of our approach, we conducted a door opening task. The training phase consists in having a single expert user perform a given task several times and in different ways. Hence for the door opening task, we collected data by attempting to reach the door handle from various angles and positions, using the VR controllers to control the robot's hand poses. This resulted in a diverse set of training data, which allows the system to learn and adapt to different scenarios that may be encountered during teleoperation, including variations in environmental conditions.

The training data is recorded in the door handle frame, which provides additional flexibility and adaptability to the detected pose of the door handle. This adaptive approach ensures that the robot can successfully complete the door opening task while compensating for variations in the environment and any inaccuracies in the user's input, especially when approaching the real object due to poor situational awareness. As shown in Fig. [5,](#page-4-1) the updated ProMPs also align with the initial user's intention. This enables the user to effectively direct the robot's motion as they desire, while maintaining fidelity to the demonstrations. When approaching the object, a blending policy gradually shifts the input from the ProMP-calculated trajectory to the AT, as illustrated in Fig. [6.](#page-5-1) This blending mechanism ensures smooth transitions between the autonomous motion generated by the ProMP and the controlled manipulation governed by the AT. By seamlessly

integrating these two components, the robot can effectively and accurately interact with the object, reducing the risk of errors or unintended consequences during physical interaction.

To further emphasize the adaptability to the user intention of our system, we extended our evaluation to include a punching task. The unique aspect of this task lies in its multifaceted execution: a single action, such as a punch, can be manifested in diverse techniques, each so distinct that they're individually labeled, like hook punches, uppercuts, and jabs. Instead of devising separate ProMPs for each distinct punching technique and target, we opted for a more integrated strategy. A singular ProMP was developed to cover all techniques, adapting to different targets within the task. This approach aligns with the concept of having a limited number of movement primitives associated with each context.

By employing this unified ProMP for the punching task, we aimed to assess the system's ability to generalize and adapt its punching actions to the different ways the user wants to per-form the motion^{[2](#page-5-2)}. Differently from the door opening task, the punching motion required the coordination of multiple body parts, necessitating additional ProMPs for the forearm orientations, and chest orientation. In this case the user was also equipped with three Vive trackers 3.0 that tracked the forearms and chest motion. Fig. [7](#page-6-0) shows the ProMPs adaptation during a left punching motion. The left column displays the learned ProMPs alongside the corresponding demonstrations, while the remaining columns show the updated ProMPs after observing a portion of the test trajectories. The updated ProMPs align with the user's intended punching motion, allowing the robot to punch different targets with the user's chosen punching technique.

Table [I](#page-6-1) reports the mean Root Mean Square (RMS) comparison – with associated standard deviation (*SD*) – between the trajectories generated by the user with those generated by the assistive autonomy after using a portion (about a third) of the same user directed trajectories as observation. The mean is calculated based on 10 test trajectories for the punching motion^{[3](#page-5-3)}. The RMS errors across different motion components indicate a remarkably small deviation between the trajectories generated by the assistive autonomy system and the user-directed robot motion (2.35*cm* with a standard deviation of 0.87*cm* for position and no more than 0.05*rad* with a standard deviation of 0.02*rad* for orientation). The similarity between the adapted trajectories and the user's intended motion demonstrates the system's ability to perform the task as intended by the user. Videos and data of the experiments are available online^{[4](#page-5-4)}.

B. Feasibility Studies

The IHMC Robotics Team conducted feasibility studies to test whether the humanoid robot can achieve the door opening

²Note that the punching task does not incorporate an AT; hence no blending is involved.

³The RMS error for a single test is calculated by evaluating the errors between each sample of the two compared trajectories. Next, the overall RMS in position or orientation is computed, followed by calculating the mean and *SD* over all the tests.

⁴Dataset: [https://doi.org/10.5281/zenodo.8215964,](https://doi.org/10.5281/zenodo.8215964) video: [https://youtu.be/](https://youtu.be/oN-FD6YnF2c) [oN-FD6YnF2c](https://youtu.be/oN-FD6YnF2c)

Fig. 7. ProMP adaptation during punching motion. The ProMPs for the most relevant body parts involved in a left punching motion are shown. Left column: learned ProMPs (light thick lines with transparent regions) alongside the corresponding 27 demonstrations (dark thin lines), expressed in the target frame. These demonstrations include 9 jabs, 9 hook punches, and 9 uppercuts. Remaining columns: updated ProMPs (dark thick lines) after observing a portion of the test trajectories (portion of the lighter thinner lines up to the vertical gray bar). We compare these updated ProMPs with the previously learned ProMPs. From left to right, the test trajectory represents a jab, a hook punch, and an uppercut, respectively.

TABLE I ASSISTIVE AUTONOMY MOTION VS USER DIRECTED ROBOT MOTION FOR THE LEFT PUNCHING TASK.

	RMS error adapted ProMP vs test
left hand position [cm]	2.35(0.87)
left hand orientation [rad]	0.03(0.01)
left forearm orientation [rad]	0.05(0.02)
chest orientation [rad]	0.04(0.01)

TABLE II RESULTS FROM FEASIBILITY STUDIES

DC: Direct Control, AA: Assistive Autonomy

task via teleoperation^{[5](#page-6-2)}. A total of eight members of IHMC participated in the feasibility studies. The group of participants included four expert users who were well acquainted with the system and its operation, and four individuals who were not

⁵The feasibility studies were classified as Non-Human-Subject Research (NHSR) and were conducted under approval from IHMC's IRB. No human research participants were involved or evaluated.

familiar with the system and that have declared not having used VR nor directly teleoperated a robot before. Each user was tasked with performing the door opening activity using two different methods: (i) direct control, and (ii) direct control with assistive autonomy. For each method, the users had to repeat the teleoperated task three times.

During each repetition, the robot's initial position and approach angle were varied relative to the door. Specifically, one trial was conducted with the robot positioned on the left side of the door, another at the center, and the third one on the right side of the door^{[6](#page-6-3)}. As a result, each user performed a total of six door opening trials - three for each of the robot positions, using each of the teleoperation methods. The following measures were collected: task completion rate, time on task, and the number of failed attempts. A trial time limit of 120 seconds was set, based on observations made during the development phase, and multiple door opening attempts were allowed within the time limit. Within the trial, we recorded the number of failed attempts as the number of times the robot was unable to successfully grasp the door handle, unlock the door after turning the handle, or both. Table [II](#page-6-4) presents the summarized results of the feasibility studies, displaying the mean values with their corresponding *SD*.

⁶<https://youtu.be/KeQwt9a9wV8>

V. LIMITATIONS AND FUTURE WORK

The system does not dynamically adjust to the user's pace in real-time. It is capable of registering the user's movement and can estimate or be pre-set to a custom speed. Nonetheless, it fails to respond to variations in the user's speed during live input, as showcased in the supplementary online video^{[7](#page-7-21)}. This issue is particularly significant in the context of human-robot interactions and can be resolved by utilizing Interaction Primitives [\[22\]](#page-7-22), which are an extension of ProMPs that facilitates real-time speed adaptation.

Another drawback of the current approach is that attempting to adapt the model based on observations that deviate significantly from the trained standard deviation can lead to improper conditioning of the model, resulting in undesirable behaviors⁶. During the door-opening task within our feasibility study, such issues did not arise, as the demonstrations encompassed a wide range of possible methods and angles of approach. Nevertheless, in scenarios such as punching tasks, users might execute the action in unforeseen and alternative manners. To mitigate this in future developments, we plan to introduce a validation step for such trajectories, which will identify and rectify outliers by mapping them to the nearest point within the confines of the learned model.

In our future work, we also plan to conduct extensive user studies to thoroughly analyze the benefits of our approach in terms of user workload and situational awareness. This will provide valuable insights into how effectively the system meets user needs and expectations.

VI. CONCLUSIONS

Our work addresses the inefficiencies observed in humanoid robot teleoperation by introducing autonomous assistance features. We have identified challenges in traditional direct control methods and proposed a shared-control framework that combines user input as high-level guidance for the robot's autonomous capabilities. By leveraging ProMPs and ATs we facilitate robot operation. Further enriched by mixed reality technology, we provide users with an immersive platform that integrates the assistive autonomy. This unique framework equips users with interface elements designed for effective supervision, prediction, and management of the robot's assistance.

REFERENCES

- [1] K. Darvish, L. Penco, J. Ramos, R. Cisneros, J. Pratt, E. Yoshida, S. Ivaldi, and D. Pucci, "Teleoperation of humanoid robots: A survey," *IEEE Transactions on Robotics*, vol. 39, no. 3, pp. 1706–1727, 2023.
- [2] Y. Ishiguro, K. Kojima, F. Sugai, S. Nozawa, Y. Kakiuchi, K. Okada, and M. Inaba, "High speed whole body dynamic motion experiment with real time master-slave humanoid robot system," in *IEEE Int. Conf. on Robotics and Automation*, May 2018, pp. 5835–5841.
- [3] L. Penco, N. Scianca, V. Modugno, L. Lanari, G. Oriolo, and S. Ivaldi, "A multimode teleoperation framework for humanoid loco-manipulation: An application for the iCub robot," *IEEE Robotics and Automation Magazine*, vol. 26, no. 4, pp. 73–82, 2019.
- [4] A. Paraschos, C. Daniel, J. Peters, and G. Neumann, "Using probabilistic movement primitives in robotics," *Autonomous Robots*, vol. 42, p. 529–551, 07 2018.
- [5] A. Pettinger, C. Elliott, P. Fan, and M. Pryor, "Reducing the teleoperator's cognitive burden for complex contact tasks using affordance primitives," in *IROS*, 2020, pp. 11 513–11 518.
- [6] M. Johnson, J. M. Bradshaw, P. J. Feltovich, C. M. Jonker, M. B. Van Riemsdijk, and M. Sierhuis, "Coactive design: Designing support for interdependence in joint activity," *Journal of Human-Robot Interaction*, vol. 3, no. 1, pp. 43–69, 2014.
- [7] S. Jorgensen, M. Wonsick, M. Paterson, A. Watson, I. Chase, and J. S. Mehling, "Cockpit interface for locomotion and manipulation control of the NASA Valkyrie humanoid in Virtual Reality," *NTRS - NASA Technical Reports Server. ntrs.nasa.gov/citations/20220007587*, 2022.
- [8] J. C. Vaz, A. Dave, N. Kassai, N. Kosanovic, and P. Y. Oh, "Immersive auditory-visual real-time avatar system of ana avatar xprize finalist avatar-hubo," in *2022 IEEE International Conference on Advanced Robotics and Its Social Impacts (ARSO)*, 2022, pp. 1–6.
- [9] R. Luo, C. Wang, E. Schwarm, C. Keil, E. Mendoza, P. Kaveti, S. Alt, H. Singh, T. Padir, and J. P. Whitney, "Towards robot avatars: Systems and methods for teleinteraction at avatar xprize semi-finals," in *IROS*, 2022, pp. 7726–7733.
- [10] M. Schwarz, C. Lenz, R. Memmesheimer, B. Pätzold, A. Rochow, M. Schreiber, and S. Behnke, "Robust immersive telepresence and mobile telemanipulation: Nimbro wins ana avatar xprize finals," *Int. J. of Soc. Robotics*, 2023.
- [11] M. R. Endsley, "Toward a theory of situation awareness in dynamic systems," *Human factors*, vol. 37, no. 1, pp. 32–64, 1995.
- [12] A. D. Dragan, K. C. Lee, and S. S. Srinivasa, "Legibility and predictability of robot motion," in *HRI*, 2013, pp. 301–308.
- [13] D. Rakita, B. Mutlu, M. Gleicher, and L. M. Hiatt, "Shared control–based bimanual robot manipulation," *Science Robotics*, vol. 4, no. 30, 2019.
- [14] R. Rahal, F. Abi-Farraj, P. R. Giordano, and C. Pacchierotti, "Haptic shared-control methods for robotic cutting under nonholonomic constraints," in *IROS)*, 2019, pp. 8151–8157.
- [15] S. Javdani, H. Admoni, S. Pellegrinelli, S. S. Srinivasa, and J. A. Bagnell, "Shared autonomy via hindsight optimization for teleoperation and teaming," *The International Journal of Robotics Research*, vol. 37, no. 7, pp. 717–742, 2018.
- [16] A. Broad, I. Abraham, T. Murphey, and B. Argall, "Data-driven koopman operators for model-based shared control of human–machine systems," *IJRR*, vol. 39, no. 9, pp. 1178–1195, 2020.
- [17] M. Selvaggio, J. Cacace, C. Pacchierotti, F. Ruggiero, and P. R. Giordano, "A shared-control teleoperation architecture for nonprehensile object transportation," *IEEE Transactions on Robotics*, vol. 38, no. 1, pp. 569–583, 2022.
- [18] T. Koolen, S. Bertrand, G. Thomas, T. De Boer, T. Wu, J. Smith, J. Englsberger, and J. Pratt, "Design of a momentum-based control framework and application to the humanoid robot atlas," *Int. Journal of Humanoid Robotics*, vol. 13, pp. 1 650 007–1, 03 2016.
- [19] S. J. Jorgensen, M. W. Lanighan, S. S. Bertrand, A. Watson, J. S. Altemus, R. S. Askew, L. Bridgwater, B. Domingue, C. Kendrick, J. Lee, and et al., "Deploying the NASA Valkyrie humanoid for IED response: An initial approach and evaluation summary," in *IEEE/RAS Int. Conf. on Humanoid Robots*, Oct 2019, p. 1–8.
- [20] J. Nielsen, "Enhancing the explanatory power of usability heuristics," in *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, 1994, pp. 152–158.
- [21] G. Adamides, G. Christou, C. Katsanos, M. Xenos, and T. Hadzilacos, "Usability guidelines for the design of robot teleoperation: A taxonomy," *IEEE Transactions on Human-Machine Systems*, vol. 45, no. 2, pp. 256– 262, 2014.
- [22] J. Campbell and H. Ben Amor, "Bayesian interaction primitives: A slam approach to human-robot interaction," in *Proceedings of the 1st Annual Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, vol. 78. PMLR, 13–15 Nov 2017, pp. 379–387.