Utilizing Operator Intent for Haptic Teleoperation Under High Latencies

H.J.C. Kroep, P. Makridis, J. Huidobro, K. Wösten, D. Choudhary, N. Gnani, T.V. Prabhakar, S. Coppens, K. Van Berlo, and R. Venkatesha Prasad

Abstract—**Haptic teleoperation is a promising technology with applications in telemaintenance and disaster management. However, it faces significant challenges when the application is subjected to a high network latency and environments with moving objects. This work aims to extend Model Mediated Teleoperation (MMT) to overcome challenges in supporting dynamic environments. Instead of striving for perfect model alignment, we acknowledge the inevitable mismatch between the remote environment and its model at the operator. We propose a set of design principles and an accompanying framework for designing MMT solutions that prioritize operator intent. Our approach is exemplified through an application where an operator, located 8000 km away (The Netherlands – India) and subjected to an average of 179 ms end-to-end latency, guides a robot arm to draw on a whiteboard whose position is actively altered. We evaluate the effectiveness of our approach through a user study. We show a 3-point improvement on a 7-point Likert scale when users utilize our approach to teleoperate over significant network latency of up to 1 s.**

Index Terms—**Digital twins, 5G mobile communication, human computer interaction, human in the loop, tactile internet, telepresence.**

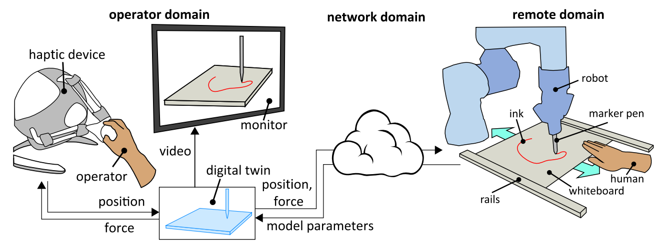

Fig. 1. We develop an approach allowing teleoperation over long distances. We demonstrate the effectiveness of our approaches by creating the application shown in the figure. An operator uses a haptic device (in TU Delft, the Netherlands) to guide a robot equipped with a marker to draw on a canvas situated on a different continent (IISc Bangalore, India). Simultaneously, another individual alters the canvas's position as the operator makes the drawing. See https://youtu.be/pJHoO2d8VVo for a video demonstration.

I. Introduction

TELEOPERATION is a compelling technology that is becoming increasingly feasible due to advancements in networking and robotics [1]. Some of the most celebrated teleoperation applications include telesurgery, telemaintenance, and remote disaster management [2], [3].

In a haptic teleoperation setup, the operator interacts with a haptic device to express their intent of actions that should be executed in the remote domain. Their movements and manipulations are transmitted to the remote side, where a robotic device imitates the operator's actions. The operator receives both visual and haptic feedback. In particular, haptic feedback allows the operator to refine their actions more precisely as if they were physically present in the remote environment. An example of such a system is illustrated in Fig 1.

Received 9 September 2024; revised 21 May 2025; accepted 3 July 2025. Date of publication 23 July 2025; date of current version 5 November 2025. Recommended for acceptance by S. Wang. *(Corresponding author: H.J.C. Kroep.)*

H.J.C. Kroep, P. Makridis, J. Huidobro, K. Wösten, S. Coppens, K. Van Berlo, and R. Venkatesha Prasad are with the Department of Networked Systems, Delft University of Technology, 2628 Delft, The Netherlands (e-mail: h.j.c.kroep@tudelft.nl; r.r.venkateshaprasad@tudelft.nl).

D. Choudhary, N. Gnani, and T.V. Prabhakar are with the Department of Electronic Systems Engineering, Indian Institute of Science, Bangalore 560012, India (e-mail: tvprabs@iisc.ac.in).

Digital Object Identifier 10.1109/TMC.2025.3591197

In practice, realizing teleoperation over a network is challenging due to the delay imposed by the network. As the distance between the operator and the remote environment increases, this problem becomes more significant. Moreover, the delay variation would also affect the stable teleoperation. In this work, we consider two approaches for mitigating the effects of this latency:

Tactile Internet (TI): is a recent approach that aims to develop ultra-low latency networks, with a latency of under 1ms and high reliability [4], [5]. Given such a network, we can realize teleoperation applications that deal with arbitrarily complex environments. The downside is that TI comes with an inherent maximum distance (≈150 km or 1 ms in terms of delay) supported due to fundamental physics-imposed limitations on network latencies. However, conceptualizing any network with such low latency is profoundly challenging even without the maximum distance.

Model-mediated teleoperation (MMT): differs from TI by assuming significant network delays from the outset rather than focusing on directly reducing latency. Instead, MMT aims to mitigate the impact of these delays on the system's transparency and stability [1], [6], [7]. An MMT system comprises two main components: (a) the operator and (b) the remote environment. On the operator's side, a detailed model replicating the characteristics of the remote environment is deployed. The operator then uses a haptic device to demonstrate their intended actions, which are sent to the remote robot, receiving instantaneous feedback based on the local model of the remote environment.

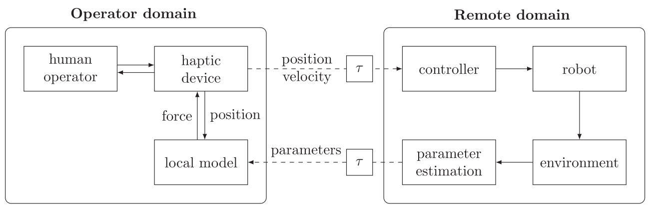

Fig. 2. General structure of a Model-Mediated Teleoperation system.

On the remote side, the robot executes the received commands while collecting sensor data such as force, position, and audio-visual information. This data is used for real-time estimation of the remote environment's model parameters. Instead of transferring all sensory data to the operator, only the model parameters are sent. The digital twin in the operator's domain is subsequently updated with these parameters. A high-level overview of this structure is illustrated in Fig. 2.

While MMT effectively addresses significant network delays, spanning several seconds [6], [8], using such methods also imposes three significant restrictions on the system. ① Performance relies on how well the local model tracks changes in the remote environment. In dynamic settings with a high-velocity motion, any latency can quickly lead to significant mismatches between the local simulation and the real world. Additionally, ② MMT methodologies lean on handcrafted models of the remote environment, making them less adaptable to increasingly complex scenarios. Lastly, ③ higher dynamic delay makes it difficult for the model to mimic the remote environment.

Thus, in this work, we aim to extend MMT to allow for complex and dynamic environments in the presence of considerable network latency. Instead of requiring the local model to match the remote environment accurately, we embrace that mismatch is unavoidable in dynamic environments and consider the operator intent as a way to navigate the mismatch. To the best of our knowledge, we are the first to suggest this approach. The contributions in this work are as follows:

❶ We introduce design principles for MMT solutions that prioritize operator intent.

❷ To enhance the scalability of MMT solutions, we propose leveraging physics engines that prioritize computational speed over absolute accuracy while integrating mechanisms to effectively manage the resulting inaccuracies.

❸ We present a comprehensive framework for MMT solutions tailored for complex, dynamic environments, incorporating an imitation controller.

❹ We demonstrate our framework by implementing a system that guides a robot arm to draw on a moving whiteboard over an 8000km distance (TU Delft in the Netherlands and IISc Bangalore, India). See video.[1]

❺ Our user study underscores the efficacy of our approach, with significant improvements in user experience under network latency of up to 1s. We show a 3-point improvement in user experience on a 7-point Likert scale.

[1]https://youtu.be/pJHoO2d8VVo

The rest of this work is structured as follows. Section II offers an overview of related literature. Section IV outlines design principles to enhance MMT. Section III introduces a mathematical framework for MMT applications. Section V applies the framework to a specific application. Section VI details the experimental setup of our user study. Section VII presents an analysis of the user study results. Finally, Section VIII summarizes and concludes our findings.

II. Related Works

A. Tactile Internet

Multiple empirical studies demonstrate that teleoperation applications where force feedback is transmitted over the network require sub 10ms latency to function [9], [10]. Kroep et al. showed that these requirements can be slightly relaxed when interacting with soft objects with low rigidity, but this is highly restrictive for applicability [10].

A significant number of recent developments in networking have achieved a reduction in latency, which has advanced networks towards meeting Tactile Internet requirements. Recently, Promwongsa et al. provided a comprehensive study on advancements and research directions in Tactile Internet [11]. Advancements in 5G and future iterations will significantly improve network latency [12]. Of particular importance to TI are advancements in Software Defined Networks (SDN) and Network Function Virtualization (NFV) [13]. In particular, the kinematic data transmission in TI applications is a small but extremely low latency constraint, making SDN and NFV ideal solutions to provision for this type of traffic.

B. Model Mediated Teleoperation

MMT aims to enable teleoperation under significant network latency but introduces its own challenges. An important challenge in MMT is the model jump effect, which happens because of discrepancies between local model predictions and the real-time outcomes in remote environments. Updating the local model can cause jumps in the operator domain, resulting in a poor user experience and unintended control signals sent to the remote domain. [14]. Several methods have been explored to reduce the model jump effect, including delaying model updates and alerting operators about impending updates [8], [15], [16]. These solutions generally improve the user experience and should be actively considered for applications where the model jump effect is noticeable.

In MMT, another challenge emerges in designing the controller in the remote domain, particularly when attempting to execute actions demonstrated in the operator domain when there is a mismatch in states due to an inaccurate model. Song et al. provide a method that restricts the robot from applying destructively high forces or fast movements, thus limiting the operator's ability to cause unintentional damage to the environment. This is done by introducing an adaptive impedance controller [8]. Finally, MMT struggles with dynamic environments with moving objects. Xu et al. initiated the advancement of MMT to accommodate movable objects [17]. They adopted a model-based approach tailored to a particular scenario, limiting its broader applicability.

C. Discerning Human Intent

In our work, we add to the MMT design by considering the intent of the human operator. While this approach has seen limited investigation in the context of MMT, it has been studied actively in other fields. One such field is robot-human collaboration, where understanding operator intent is vital. Several studies aim to decipher human intent for synchronous robot collaboration [18], [19], [20]. Note that for collaboration, human intent is determined so that a robot can collaborate with a human, while in MMT, human intent should be determined to replicate it. Recently, Ly et al. used intent prediction to enhance haptic teleoperation by modulating robotic assistance via adaptive force guidance [21]. Their results show that intent-aware guidance reduces grasp attempts and improves trajectory smoothness, highlighting its benefits for teleoperation.

Another place where human intent is considered is when designing AI agents that are trained to adopt the skills of humans. Learning from Demonstration (LfD) is an intuitive way to transfer human skills to robots. Here a human demonstrates how to perform an action, which is then abstracted into skill models [19], [22], [23]. A robot can then perform similar actions in a new environment. In contrast with MMT, these approaches involve a form of training before deploying the robot. Similarly, imitation learning techniques aim to make AI agents behave as a human would when presented with the same scenario as the AI is currently in [24].

Another significant line of research focuses on modeling the human operator as a digital twin. Human Digital Twins (HDTs) aim to digitally represent human behavior or physiology to enable personalized and immersive interaction. Recent work explores HDTs through advances in networking, generative AI, edge computing, and multimodal feedback [25], [26], [27], [28]. These efforts aim to create real-time, responsive digital representations of users that can adapt to changing conditions and personalize system behavior. In many cases, HDTs are used to support remote monitoring, training, or assistance by maintaining a synchronized virtual model of the human. Such capabilities are especially relevant in healthcare, rehabilitation, and collaborative teleoperation settings.

Our work builds on advancements in MMT to enable active force feedback while incorporating operator intent. Our framework enables intuitive teleoperation in dynamic environments, allowing operators to manipulate a remote environment effectively.

III. A Mathematical Framework for Teleoperation With MMT

This section provides a mathematical framework for teleoperation that builds upon Model-Mediated Teleoperation (MMT). An overview of this framework is shown in Fig. 3. We divide the teleoperation system into three main parts:

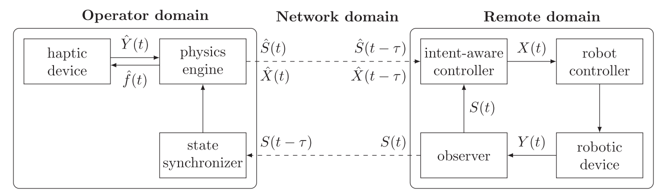

Fig. 3. Illustration of the proposed teleoperation framework, structured around an operator, remote, and network domain.

Operator domain: Houses the operator's haptic device and provides the user with local, real-time feedback based on a digital twin of the remote environment.

Remote domain: Contains the robotic device and environment in which real interactions occur.

Network domain: Transfers data between the operator and remote domains.

Throughout this work, any parameter in the remote domain is represented with a corresponding $\hat{}$ notation in the operator domain. Let

$$S(t) \; = \; \{\text{object states, environment properties}, \dots\}$$

be the observed state of the remote environment at time t. This may include object positions, orientations, shapes, mass, friction coefficients, and other relevant attributes. Some of these can be provided as accurate priors, while others might be approximate and refined over time. In the operator domain, we maintain a digital twin of the remote environment,

$$\hat{S}(t) \; = \; \{\text{virtual replicas of states in } S(t)\}.$$

A physics-based model or another simulation mechanism in the operator domain updates $\hat{S}(t)$ in real-time, allowing the user to interact locally with an environment that approximates $S(t)$. In this work, we consider the simulation mechanism to be a physics engine.

To capture raw sensor data independently of the interpreted control or state variables, we introduce $Y(t)$ and $\hat{Y}(t)$. Here, $Y(t)$ represents all raw measurements in the remote domain, such as robot encoder readings or camera data, while $\hat{Y}(t)$ collects the raw measurements in the operator domain, for example from a haptic device. These measurements do not directly define the state $S(t)$ or the action $X(t)$; rather, they serve as inputs to processes that update $S(t)$, $\hat{S}(t)$, $X(t)$, and $\hat{X}(t)$.

In particular, the operator manipulates a haptic device, providing a measurement $\hat{Y}(t)$ that may include the end-effector position or force sensors. Based on $\hat{Y}(t)$, the system computes a control signal $\hat{X}(t)$, which represents the operator's intent in the simulated environment. For instance, $\hat{X}(t)$ may describe a virtual end-effector pose or velocity command that the operator applies to $\hat{S}(t)$. On the remote side, a control input $X(t)$ is applied to the robotic device, which interacts with the real environment $S(t)$. All data exchanged between the operator and remote domains passes through a network with inherent latency τ. Consequently, any state update or control input generated at time t in one domain may only arrive at time $t + \tau$ in the other.

In this work, we assume that there is lossless communication of all states $S, \hat{S}, X, \hat{X}$.

With the mathematical framework established, the next step is to consider the effects caused by the presence of a human operator.

IV. Design Principles for Human-Centered MMT

In this section, we describe the core design principles for our human-centered Model-Mediated Teleoperation (MMT) system. Building on the framework introduced in Section III, we focus on how to handle inevitable mismatches between the local digital twin $\hat{S}(t)$ and the remote environment $S(t)$, the critical role of operator intent, and the importance of operator perception in ensuring a seamless user experience. We then conclude with a discussion on *scalability* considerations, specifically the choice of real-time physics engines. We validate these principles in Section V with a concrete application and user study.

A. Prioritizing Speed Over Accuracy in Local Simulation

In MMT, the local digital twin $\hat{S}(t)$ must approximate the remote environment $S(t)$ in real time. Since perfect accuracy is neither feasible nor necessary, we argue that prioritizing computational speed over precision is the better approach and that mechanisms should be in place to handle inevitable mismatches.

Physics engines simplify dynamics to ensure efficiency, omitting details like micro-collisions or precise friction modeling. These approximations introduce discrepancies, but mismatches are unavoidable regardless due to external disturbances, sensor noise, and latency.

Crucially, human perception favors smooth, responsive interactions over strict physical correctness. A fast, lightweight simulation improves usability and scalability, while robust mismatch-handling mechanisms ensure effective operation despite its inherent inaccuracies.

B. Characterizing and Managing Mismatches

To quantify the degree of divergence between $\hat{S}(t)$ and $S(t)$, we introduce a mismatch function:

$$\mathbf{e}(t) \;=\; \boldsymbol{\delta}\big(S(t), \hat{S}(t)\big), \tag{1}$$

where $\boldsymbol{\delta}$ measures the difference between the two states. Our aim is not necessarily to make $\mathbf{e}(t)$ zero at all times but to manage it so that the operator's actual intent is still respected. Mismatches arise because it is infeasible for $\hat{S}(t)$ to perfectly replicate $S(t)$ under real-world constraints. We identify four main sources:

Measurement noise: The remote measurements $Y(t)$ can be inaccurate, and their delayed arrival may lead to outdated or erroneous updates of $S(t)$.

Simulation noise: The local physics engine used to update $\hat{S}(t)$ inevitably simplifies real-world dynamics (e.g., friction, fluid flow), causing it to diverge from $S(t)$.

External influences: Humans or other agents in the remote domain can alter $S(t)$ in ways the local simulation $\hat{S}(t)$ does not predict.

Latency: All of the above effects can be amplified by communication delays.

In principle, if $\boldsymbol{e}(t)$ could be kept near zero, the operator would essentially experience an environment that precisely mirrors the real remote one, making *operator intent* straightforward to interpret. However, maintaining $\boldsymbol{e}(t) \approx 0$ at all times would demand prohibitively low network latencies. Consequently, larger mismatches are unavoidable, and it becomes critical to carefully consider the operator's goals so that the remote robot still performs the actions the operator intends.

C. Inferring Operator Intent From Local Interaction

Operator intent: describes how an operator wants to interact with the remote environment, independent of latency, modeling errors, or other artifacts. Modeling operator intent allows the teleoperation system to look beyond the direct measurements of trajectories or forces and consider how and why the operator performs certain actions, which enables the system to mitigate or compensate for inevitable mismatches.

To illustrate, consider a simple teleoperation setup where only the position of the haptic device end-effector is transmitted. In that scenario, there is limited information about how strongly the operator wishes to interact with an object or whether the operator was attempting to press a button, pivot an object, or gently tap it. By contrast, if we exploit a local physics simulation and observe the additional data streams (e.g., forces, collisions, object responses), we can gain insight into the higher-level goals guiding the operator's behavior. We identify three key aspects:

Operator's Behavior: By directly engaging with the local physics engine, the measured trajectory $\hat{Y}(t)$ of the operator results in a trajectory and applied force $\hat{X}(t)$ in the virtual state $\hat{S}(t)$. The combination of $\hat{X}(t)$and $\hat{S}(t)$ enables the inference of contextual actions. For instance:

Relative motion: The trajectory the operator follows, measured relative to objects in the environment.

Applied force: The magnitude and direction of the force exerted on different objects or regions within $\hat{S}(t)$.

Object's behavior: Understanding operator intent involves not only analyzing their direct actions but also observing how objects in the environment respond. Instead of focusing solely on raw input signals, the system can infer higher-level goals by examining how objects in $\hat{S}(t)$ are manipulated. For example, if an object is being translated, rotated, or otherwise moved in a specific way, this likely reflects a purposeful manipulation. This also includes detecting state changes such as tipping, sliding, or collisions. By incorporating these object-centric observations, the system can refine the control signal $X(t)$ to better ensure that the physical environment $S(t)$ aligns with the operator's true intent.

Critical transitions: Small object position or operator force deviations may be acceptable under most circumstances. However, certain interactions, which we call critical transitions, can magnify tiny errors into major consequences. For instance, missing the exact position needed to press a button may fail to activate a mechanism altogether, or allowing a slight misalignment when an object is on the verge of falling can lead to irreversible damage. Recognizing and explicitly tracking such transitions lets the system apply high-fidelity updates where they matter most without requiring global precision at all times.

By modeling these different facets of operator intent—behavior of the user, effect on objects, and critical transitions—the teleoperation framework can reconcile mismatches before they become noticeable or impede task success.

D. Accounting for Operator Perception

Not all mismatches between $\hat{S}(t)$ and $S(t)$ are equally perceptible to the operator. For example, research shows that humans often tolerate slow or smoothly varying discrepancies more readily than abrupt, high-frequency changes, particularly those involving contact forces or sudden shifts in object positioning [9], [10], [29].

This observation is crucial for translating operator intent into the remote domain: even if the system must modify the control signal to compensate for mismatches, such trajectory adjustments may be counterproductive if they are noticeably discontinuous or create unexpected sensations on the operator's end. For instance, an operator might tolerate a subtle scaling of motion over a few seconds without sensing anything unusual but would immediately detect an abrupt "jump" in position or force feedback.

Consequently, effective design strategies prioritize minimally perceptible manipulations of $\hat{X}(t)$ to form $(X(t)$. By leveraging knowledge of how users perceive deviations, we can allow the system to correct mismatches and realize the operator's true objectives without undermining their sense of direct control or immersion.

To implement these principles, we rely on two key components: an *intent-aware controller* and a *state synchronizer*. The intent-aware controller modifies the teleoperation control signal according to the delayed local and remote states,

$$X(t) \leftarrow f\big(\hat{S}(t-\tau),\ \hat{X}(t-\tau),\ S(t)\big),$$

with the purpose of preserving the operator's intent despite mismatches. Meanwhile, the state synchronizer maintains the local digital twin by integrating new remote measurements into $\hat{S}(t)$,

$$\hat{S}(t) \leftarrow g\big(\hat{S}(t),\ \hat{X}(t),\ S(t-\tau)\big),$$

ensuring that local and remote environments remain closely aligned. Together, these building blocks—managing mismatches, capturing and preserving operator intent, and respecting operator perception—enable a user-centered MMT system. In the following section, we implement these concepts to create a functional teleoperation application.

V. Teleoperation Application of Remote Drawing on a Whiteboard

In this section, we apply our framework and the design principles outlined to a concrete teleoperation application. In the chosen application, a person draws on a whiteboard in a remote location. The whiteboard is locked between two rails, restricting it to a 1 DoF motion over a table. The whiteboard's position can be manipulated by people present in the remote environment.

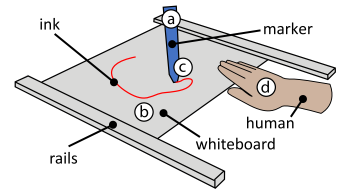

Fig. 4. Illustration of the experimental setup where a marker (a) draws on a whiteboard (b) while a human hand (d) moves the board between two parallel rails. The tip of the marker (c) deposits ink on the surface as the drawing is made.

The task is illustrated in Fig. 4. The task necessitates precise control over the marker's pressure on the whiteboard and the trajectory to give the operator control over the drawing being made despite the canvas being in motion. We go through the design principles outlined in Section IV.

A. Mismatch Analysis

The first step is to identify the mismatches that can occur in this application. This will be done as described in Section IV-B.

measurement noise: The measured whiteboard position $Y(t)$ in the remote domain is sent back and used to update $\hat{S}(t)$ in the operator domain. Significant measurement noise is generated by the way the whiteboard is tracked in the remote environment. Due to this, there will be considerable high frequency noise on the tracked position of the whiteboard.

simulation noise: In the chosen application, the simulation does not predict changes in the position of any of the objects present. Therefore, the only noise caused by the simulation will be inaccurate force calculations. However, the consequences of inaccurate calculations do not accumulate.

external influences: In this case, there is a human in the remote domain manipulating the position of the whiteboard. The remote humans behavior is considered to be unpredictable in this application.

latency: The network link has an average roundtrip latency of 179 ms. Because of this, the mismatch of the whiteboard position $\mathbf{e}(t)$ between its observed location in the remote domain and the operator's digital twin grows with both the latency and the speed of the whiteboard's motion.

B. Capturing Operator Intent

The next step is to identify operator intent as described in Section IV-C.

operator's behavior: Considering that the purpose of the application is to make a drawing, the assumption is made that the relative position and relative pressure applied by the marker to the whiteboard are highly important when the marker is in close proximity to the whiteboard. However, it is important to consider that in order to capture this behavior, the noise on the tracking of the whiteboard needs to be considered.

Object's behavior: In this application, the operator is not actively manipulating the position of an object in the environment. The ink that the marker leaves on the whiteboard is not considered object behavior.

Critical transitions: In this case, the most important critical transitions are the transition between hovering a marker over the whiteboard, drawing on the whiteboard, and crushing the marker tip against the whiteboard. In each of these transitions, a small difference of 1mm can cause a significant difference.

C. Identifying Limits in Operator Perception

Next, limits in the operator perception need to be identified. Any alterations in trajectory while drawing on the whiteboard will be clearly noticeable by the operator. In this case, the ink will leave behind a permanent reminder of the difference in trajectory in the form of a mismatch between the actual and the intended drawing. Whenever the marker is not directly touching objects, small differences in trajectory are hard to distinguish. Therefore, there is an opportunity for the controller in the remote domain to manipulate the robot's trajectory while not in direct contact with the whiteboard.

D. Translation Between Absolute and Relative Trajectory

Based on the insights obtained previously, we identify the key mismatch to consider is the active manipulation of the whiteboard by a remote human. We suggest a method that maps the end-effector positions between the operator and remote domains. When close to an object, in this case, the whiteboard, the relative position from the end-effector to the object is preferred over its global position. This transition between global and relative positions should be minimally perceivable.

In this section, we describe how to design an imitation controller that adjusts the measured operator behavior to respect operator intent in our specific application. Let $\hat{X}$ be the end-effector position directly mapped from the haptic device in the operator domain, and let X_{relative} be the corresponding position in the remote domain that maintains the same relative placement with respect to the whiteboard. We introduce a transition factor α to smoothly blend these two positions:

$$X(t) = \alpha \hat{X}(t-\tau) + \big(1-\alpha\big) X_{\text{relative}}(t), \tag{2}$$

where $\hat{X}(t-\tau)$ incorporates the communication delay τ. Specific details on how α is calculated are given in (6).

In this application, a haptic rendering algorithm converts the end-effector position of the haptic device $\hat{\boldsymbol{p}}_e$ to a proxy position in the virtual environment $\hat{\boldsymbol{p}}_p$ and applied force $\hat{\boldsymbol{f}}$. The control signal in the operator domain is thus $\hat{X} = \{\hat{\boldsymbol{p}}_p, \hat{\boldsymbol{f}}\}$, where $\{\}$ indicates a set. We consider an application with only one moving object, in this case, the whiteboard. Therefore, we can state that $S = \boldsymbol{p}_o$ and $\hat{S} = \{\hat{\boldsymbol{p}}_o\}$, where $\boldsymbol{p}_o$ and $\hat{\boldsymbol{p}}_o$ is the position of the whiteboard in the remote environment and the digital twin, respectively. We specify the collection of points that comprise the object as P_o in such a way that if a position $\boldsymbol{p}$ is inside the object, then $(\hat{\boldsymbol{p}} - \hat{\boldsymbol{p}}_o) \in P_o$ in the operator domain and $(\boldsymbol{p} - \boldsymbol{p}_o) \in P_o$ in the remote domain.

The analysis of operator intent given at the beginning of this section suggests that near the whiteboard, the relative distance between the operator and the whiteboard is more significant than the absolute position. This leads to the following modification,

$$X_{\text{mod}}(t) = \{\hat{\boldsymbol{p}}_p(t-\tau) + \boldsymbol{p}_o(t) - \hat{\boldsymbol{p}}_o(t-\tau), \hat{\boldsymbol{f}}(t-\tau)\}, \tag{3}$$

with $\boldsymbol{p}_o(t) - \hat{\boldsymbol{p}}_o(t-\tau)$ denoting the vector from the object in the operator domain to its counterpart in the remote domain.

We first consider a transition region. We propose to use the distance between the operator proxy and the object in the operator domain as the transition region. We consider P as the collection of all points that fall inside of the object. We consider an object that cannot rotate. The closest vector from the operator to the object in the operator domain can be obtained with

$$\hat{\boldsymbol{p}}_{\min} = \arg\min_{\boldsymbol{p}\in P}(\boldsymbol{p} + \hat{\boldsymbol{p}}_o - \hat{\boldsymbol{p}}_p). \tag{4}$$

To create a smooth transition between $\hat{X}$ and X_{mod}, we introduce a transition function $g(x)$ that continuously varies from 0 to 1. We use the cubic polynomial

$$g(x) = \begin{cases} 0 & \text{if } x \leq 0, \\ 3x^2 - 2x^3 & \text{if } 0 < x < 1, \\ 1 & \text{if } x \geq 1, \end{cases} \tag{5}$$

where x is correlated to the distance between the operator's end-effector and the object. This particular function was chosen because it is the simplest cubic polynomial that satisfies the constraints $g(0) = 0$, $g(1) = 1$, and $\frac{d}{dx}g(0) = \frac{d}{dx}g(1) = 0$, ensuring both continuity and smoothness at the boundaries.

The critical case for maintaining monotonicity occurs when the operator is moving directly toward the object, i.e., when the direction of motion $\frac{d}{dt}\hat{\boldsymbol{p}}_p$ is aligned with the shortest vector to the object $\hat{\boldsymbol{p}}_{\min}$. If the slope of the transition function is equal to or greater than one, the correction term may become large enough that the robot stops moving or even moves in the opposite direction of the operator. This can be avoided by increasing the size of the transition region. However, to prevent unnecessary expansion in all directions, we only stretch the transition region along the direction of the correction.

In this work, we use a transition length of $|\boldsymbol{p}_o - \hat{\boldsymbol{p}}_o|$ in all directions except the critical one. In the critical direction, we apply a larger transition length of $2|\boldsymbol{p}_o - \hat{\boldsymbol{p}}_o|$. The resulting transition factor is given by:

$$\alpha = \begin{cases} g\left(\frac{|\hat{\boldsymbol{p}}_{\min} - \frac{1}{2}\text{proj}_{(\boldsymbol{p}_o - \hat{\boldsymbol{p}}_o)}\hat{\boldsymbol{p}}_{\min}|}{|\boldsymbol{p}_o - \hat{\boldsymbol{p}}_o|}\right) & \text{if } \hat{\boldsymbol{p}}_{\min} \cdot (\boldsymbol{p}_o - \hat{\boldsymbol{p}}_o) > 0, \\ g\left(\frac{|\hat{\boldsymbol{p}}_{\min}|}{|\boldsymbol{p}_o - \hat{\boldsymbol{p}}_o|}\right) & \text{otherwise,} \end{cases} \tag{6}$$

where $\text{proj}_{\boldsymbol{b}}\boldsymbol{a}$ denotes the projection of $\boldsymbol{a}$ onto $\boldsymbol{b}$. The chosen method demonstrated reliable performance under our application constraints, yielding smooth behavior and a positive user experience in empirical testing. Nonetheless, alternative transition functions could also be considered. Finally, we apply (3), (5), (4), (6) within (2) to compute the control signal used by the robot controller.

The method provides a formalized way to translate between the operator and remote domains, considering the spatial relationships of objects and end-effectors in both environments. The effects of this method are further illustrated in Fig. 5.

The result of deploying this method, is that whenever the end-effector is in close proximity to the object, it's relative

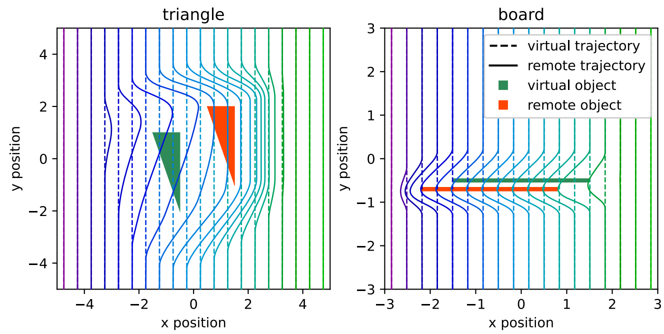

Fig. 5. Illustration of the behavior of the intent-aware controller design. For a given set of virtual trajectories of $\hat{\boldsymbol{p}}_p$ in the operator domain, the modified trajectory of $\boldsymbol{p}_p$ in the remote domain is shown.

position is prioritized. This means that the robot will never attempt to puncture the whiteboard due to latency, solving this hard transition. It will also ensure that the drawing is recreated faithfully on the remote whiteboard.

E. Controller Design

It is common for a robotic device in these types of applications to be controlled with a compliance controller that makes use of an accurate force-torque sensor [30]. In this application, there are no remotely initiated interactions that need to be relayed to the operator, as the alteration of the whiteboard position does not lead to a force on the robot's end-effector. Therefore there is no requirement of using a force-torque sensor, as the measured force would not be relayed to the operator.

For this controller we make use of a Cartesian position controller [30]. The applied force $\boldsymbol{f}$ is converted into a positional offset. We can calculate the target position fed to the position controller as

$$\boldsymbol{p}_{\text{target}} = \boldsymbol{p}_p + \frac{1}{k_s}\boldsymbol{f}. \tag{7}$$

In this case the position of the end-effector p_p is the modified version obtained using (2), (6). Note that the spring constant k_s is a parameter that determines the corresponding amount of force that should be applied proportional to the penetration depth of the operators position into the surface of an object. The parameter can be set by the system's designer, where a higher k_s yields a stiffer interaction, but also an increased susceptibility to noise.

The position correlation between the operator and remote domain is calibrated so that when the haptic devices end-effector's proxy in the operator domain contacts the whiteboard without exerting force, its corresponding position in the remote domain remains 1 mm above the whiteboard. Consequently, drawing in the remote domain will only occur when the operator applies force. Only then is the displacement caused by $\frac{1}{k_s}\boldsymbol{f}$ enough for the marker to apply pressure to the whiteboard. To match this behavior in the virtual domain, if enough force is applied to the whiteboard, ink is deposited accordingly.

In the user study, two control strategies are investigated. In the first control strategy, only the operator's behavior is considered. The operator's trajectory and applied force directly lead to a target position using (7). In the second control strategy, the mismatch in the whiteboard position between the operator and remote domains is considered. Here, (2), (7) are used to create a target position that tracks the whiteboard when it is in its vicinity. Here, $\boldsymbol{p}_p$ is a part of $X(t)$.

VI. Experimental Setup

In this section, we describe the experimental setup used to implement the teleoperation application – remote drawing with a marker on a moving whiteboard. The operator domain is deployed at the TU Delft, The Netherlands and the remote domain in deployed in IISc Bangalore, India. An overview of the application is provided in Fig. 6.

In the operator domain, the Bullet-Physics engine provides the local simulation [31]. We have adapted the physics engine to support a haptic rendering algorithm and interface with a Novint Falcon as the haptic device. The haptic rendering algorithm features a virtual proxy of the Novint Falcon end-effector that can collide and interact with objects in the virtual environment. The Novint Falcon provides position measurements and enables 3D force feedback, both at 1 kHz. The physics engine is decoupled from the rendering engine so that the physics engine can run at 1 kHz while the OpenGL-based renderer runs at 60 Hz. This ensures a smooth and responsive haptic response from the physics engine with sub 1ms computational delay. A photo of the operator domain is shown in Fig. 7(a).

In the remote domain, we deploy a UR3 robot with a gripper mounted on its end-effector to hold a marker. Two rails secured to the table constrain the whiteboard to a one-degree-of-freedom motion towards the base of the UR3. Square sponge pieces and felt pads are attached to the bottom corners of the whiteboard, providing a suspension of four millimeters until fully compressed. A ROS2 environment communicates directly with the UR3 [32]. Fixed to the movable whiteboard is a small 3D-printed part, which is tracked using an Intel Realsense D415 camera capturing RGB-D images at 90 Hz.

Fail safes were implemented to prevent the robot from making dangerous motions. If triggered, the robot comes to a stop, and participants in the user study receive a repeated explanation of restricted actions before the test is restarted. These measures ensure that the robot's velocity never exceeds a predefined maximum, prevent motions that could cause the robot or its attachments to collide with itself or the table, and automatically shut off the robot arm if no communication is received for more than 0.5 s. A photo of the remote domain is shown in Fig. 7(b).

All static elements in the remote environment are replicated and fixated in the virtual environment. The only dynamic element is the whiteboard. Data relayed to the remote domain includes the proxy position of the end-effector, the applied force, and the location of the virtual whiteboard. With the 1 DoF constraint in mind, this data captures all dynamic information within the virtual environment. The remote domain sends feedback – the whiteboard position and live camera footage – to the operator.

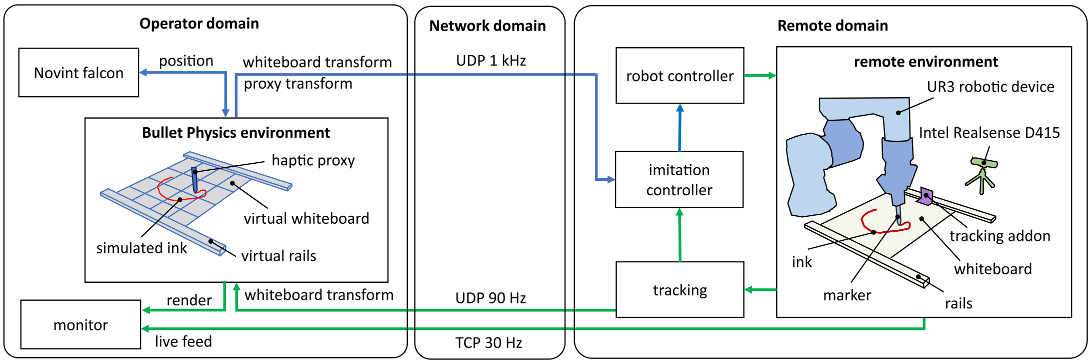

Fig. 6. Schematic overview of experimental setup. In the figure, we lay down the different components of our setup, showcasing how they relate to the proposed teleoperation framework and how data flows through the system. Blue arrows indicate high-frequency communication of 1 kHz, while green arrows indicate a medium-frequency communication of approximately 60 Hz.

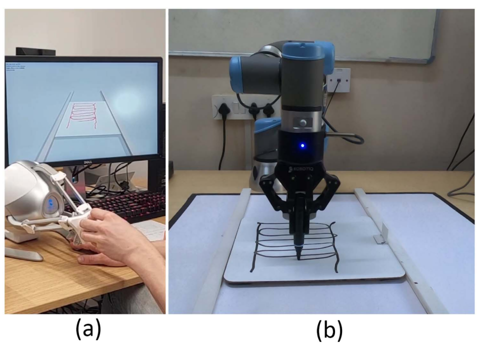

Fig. 7. The experimental setup used in the user study. (a) the operator domain and (b) remote domain. There is suspension fitted below the whiteboard in the form of sponge and felt pads. This also makes the whiteboard glide consistently over the surface.

A. Whiteboard Tracking and Updating

In this setup, the only tracking needed in the remote domain is the position of the whiteboard. The known 1 DoF movement limitation of the whiteboard enables a tailored and swift tracking solution. However, this can be extended for multiple DoF with more cameras and sensors. The captured point cloud is used to track the 3D-printed part that is affixed to the whiteboard and protrudes above the rails.

The whiteboard's movement is restricted to a 1 DoF motion along a rails on a surface. Due to this, the possible locations of the affixed 3D part are highly restricted. To refine the tracking, only a narrow section of the point cloud, where only the 3D printed part's points exist, is used. By averaging all the points within this section, we obtain the position of the affixed part, and thus the whiteboard's position. Consequently, a high-precision, 90 Hz sampling rate tracking solution with minimal computational lag was realized. There are a multitude of alternatives that can be used for tracking an object that is restricted to a specific 1 DoF motion, but a low latency measurement is highly beneficial when using the measurement to have the robot compensate for the whiteboard's movement.

In this setup, the state synchronizer is only responsible for synchronizing the whiteboard's position. Because the whiteboard's movement is only influenced by the remote domain, complications that could arise from synchronizing actively manipulated objects are avoided. Thus, remote domain measurements directly inform the virtual domain's whiteboard positioning. The model update works in the form of a teleport, so the update does not cause a spike in frictional force for the operator.

During the experiment, the whiteboard is moved at an average speed of approximately 0.8 m/s. Due to the latency in tracking and the responsiveness of the robot arm, increasing the movement speed reduces the accuracy of the drawing. Additionally, at higher speeds, it becomes very challenging for any operator to produce an accurate drawing on such a moving surface.

B. Network

The operator and remote domains are separated by an approximate distance of 8000 km. Kinematic and force data from the operator, as well as kinematic data from the whiteboard, are relayed over a UDP channel at 1 kHz and 90 Hz, respectively. Both feedforward and feedback packets carry a 100 byte payload, with a total packet size of approximately 142 bytes, including headers. Additionally, the packets contain a sequence number so that only the most recent packets are used, while out-of-order packets are ignored. A video stream from the remote domain is forwarded to a secondary computer in the operator domain at a maximum rate of 30 Hz.

The connection between the operator and remote domains is established over the public internet using a standard Ethernet link, with necessary ports opened on both sides. Since we do not impose restrictions on routing beyond our local networks, the specific ISP routing details were not a controlled factor in our experiments.

The Round-Trip Time (RTT) for the UDP link was assessed over 10hours, and the results are depicted in Fig. 8. The RTT reveals that 80% of packets arrive between 172 ms and 177 ms, with a stable average RTT of approximately 179 ms. Additionally, network jitter, defined as the variation in packet inter-arrival times, remains low, ranging from 0.014 ms to 0.022 ms. Last, to push the network further in our experiments, we used NetEm

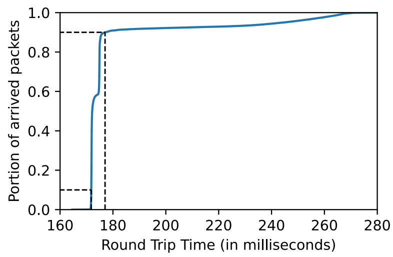

Fig. 8. Cumulative distribution of end-to-end network latency measured over 15 hours. 80% of the packets have latencies between 172 ms and 177 ms. The average latency is 179 ms.

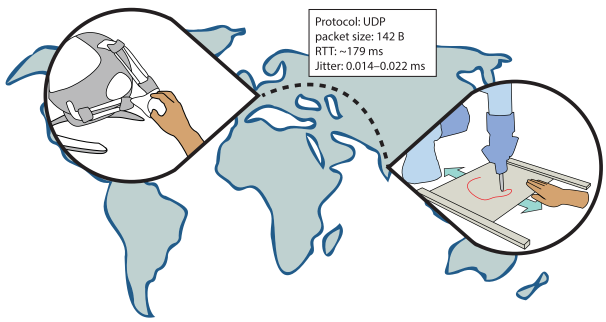

Fig. 9. Network link properties used for the user study.

to increase the latency by 1 s. The network setup has been illustrated in Fig. 9.

C. Objective Drawing Quality

Assessing the quality of a drawing objectively poses significant challenges, particularly when trying to distinguish between the skill of the human operator and the negative effects caused by the system. The core issue is the inability to accurately gauge the operator's intended drawing and use it as a ground truth. Nonetheless, we can establish an objective correlation between the drawings produced in the virtual environment of the operator domain and their reproductions on the whiteboard in the remote domain without requiring a user study.

We analyze the impacts of two variables: the tracking speed of the whiteboard and the network delay. We employ a hard-coded sequence representing a drawing, serving as a repeatable benchmark. The whiteboard is driven by a motor following a predictable path. The whiteboard is fitted onto a lead screw, which allows for a predictable sideways motion that can imitate the behavior of the remote human. In this case the system is configured to make repetitive oscillating motions. This setup enables precise predictions of the whiteboard's position, facilitating the examination of delays in capturing the whiteboard's position in real-time. During each experiment, we record the positions of the robot's end effector and the whiteboard in the remote domain, alongside the virtual end effector and virtual whiteboard positions in the operator domain.

Comparing these sequences is complex, as they do not synchronize in time. To perform this comparison, we utilize the ETVO algorithm as proposed by Kroep et al. [9]. This algorithm is an adaptation of the Dynamic Time Warping algorithm and

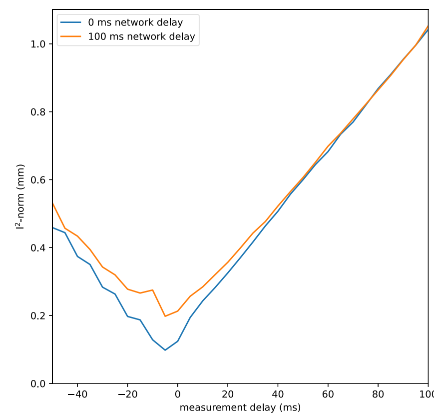

Fig. 10. This image shows the effect of network delay and measurement delay of the whiteboard on the error of the reproduced image. Negative measurement delay means prediction is used. If there is measurement delay, the robot cannot compensate for the movement in the whiteboard, resulting in an incorrect drawing and increased error. Likewise, excessively predicting the movement of the whiteboard means the robot is compensating for movements that have not yet taken place. The optimal point is at -5 ms of prediction, because that compensates for the delay in the sensing method and thus improves the drawing.

effectively separates the effects of time mismatches and position mismatches.[2] It quantifies image quality using an ℓ^2-norm.

The results are shown in Fig. 10. Here, one can see that the presence of the network has a minor impact on the quality of the drawing, resulting in minor deviations from the originally drawn path. One can also see that the ability to measure the whiteboard's position with low latency is of high importance, as this directly influences the ability of the robotic device to adjust based on the whiteboard's movement. In this plot, predictions have been included to show that an accurate prediction can compensate for the inevitable delay accumulated due to hardware, communication, and computation.

D. User Study

While the objective result demonstrates the system's ability to replicate a drawing accurately, it does not directly indicate a good performance. For that, a user study is needed. In the user study, participants use the experimental setup to draw pictures in a remote environment under varying conditions. Firpst, the participant practices drawing in the virtual environment without any connection to the remote domain until they are comfortable. This usually takes 5 minutes. The participants are challenged to recreate a specific drawing that involves two vertical lines and a zigzag pattern in between them. An example of such a drawing being made is shown in Fig. 7. The participants are presented with four scenarios.

① *Render with stationary whiteboard, natural latency:* The participant only observes the local render in the operator domain.

② *Live feed with stationary whiteboard, natural latency:* The participant only observes live video of the remote domain.

[2] Directly using the DTW algorithm is possible when addressing the challenge of long sequences leading to extremely long computation times.

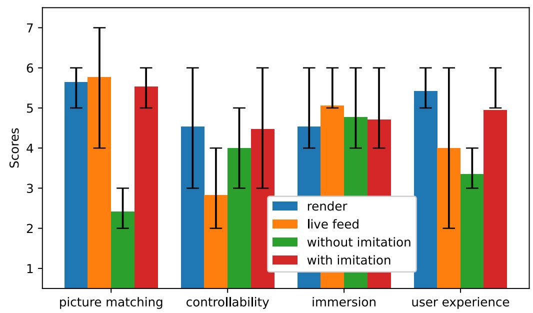

Fig. 11. The results of the user study. Four scenarios are rated on four categories using a 7-point Likert scale.

③ *Render with moving whiteboard, without intent-aware controller, increased latency:* The participant only observes the local render in the operator domain. The whiteboard is in constant motion. The controller does not consider the movement of the whiteboard.

④ *Render with moving whiteboard, with intent-aware controller, increased latency:* The participant only observes the local render in the operator domain. The whiteboard is in constant motion. The intent-aware controller considers the movement of the whiteboard. For scenario ③ and ④ an artificial latency of 1S is added using NetEm. Participants are tasked with rating each scenario in the following four aspects. Each aspect is rated on a Likert scale with seven points.

❶ *Picture matching:* How well does the final image in the remote domain match what you looked at while drawing?

❷ *Controllability:* How much control do you have over the drawing in the remote domain?

❸ *Immersion:* Do you feel like you are present at the remote location and all the things happening there are experienced by you?

❹ *Overall experience:* This rating reflects the user's overall experience, taking into account factors such as picture matching, controllability, and immersion. These aspects are prioritized based on the user's personal preferences. The user study was performed with 20 participants. The participants were aged between 20 and 60 years old, with an average age of 31 years. approximately half of the participants had never interacted with the haptic device before with the other half having at least 30 minutes of experience. No participant suffered from known neurological disorders.

VII. Performance Analysis

In the user study, participants were asked to draw on a remotely located whiteboard under varying teleoperation conditions. Fig. 11 summarizes their ratings across four distinct scenarios, each evaluated along four different axes on a 7-point Likert scale.

The first two scenarios differ only in the source of video feedback:

Render: Participants observe a locally rendered simulation with immediate force feedback but approximate graphics.

Live Feed: Participants receive real-time video of the remote domain, albeit with visible latency.

In both cases, the local simulation provides instantaneous force feedback. In the live feed scenario, the force feedback precedes visual confirmation that the marker has contacted the whiteboard.

Picture matching: and *immersion* both favored the *Live Feed*. As expected, witnessing the real environment improved accuracy and depth of immersion: rather than relying on the simulated view, participants could directly observe the pen's motion on the remote whiteboard. However, participants found the *Render* scenario offered better controllability. They reported feeling more confident and able to perform tasks swiftly without compensating for video latency. When using the live video feed, operators often deliberately slowed their motions to mitigate the delay.

Inference 1: Real-time video of the remote environment improves immersion and reflects the true state more accurately.

Inference 2: Locally rendered simulation offers greater controllability, which translates into a more seamless user experience.

The next two scenarios introduced motion to the whiteboard, resulting in a dynamic environment and an added 1s of network delay. Participants tested:

Absolute control: The system ignores any relative offset from the moving whiteboard.

Intent-aware control: The system adjusts the end-effector's trajectory to account for the whiteboard's changing position.

Despite no subjective difference in controllability or immersion during the task, participants observed a stark contrast in the final drawings. With absolute control, the resulting sketches deviated significantly from the operator's intentions, ranking it lowest overall. Even though participants valued a smooth user experience, the mismatch in the final drawing was noticeable and undesirable.

Inference 3: A straightforward Model-Mediated Teleoperation (MMT) approach fails to handle dynamic environments, causing large discrepancies between operator intent and real outcomes.

Conversely, with the intent-aware controller, most participants found that the drawn result on the remote whiteboard closely matched their local sketches. This capability to account for an actively moving surface while preserving the user's intended trajectory significantly boosted overall satisfaction, outperforming even the live-feed scenario with a stationary board.

Inference 4: Capturing and preserving operator intent via an intent-aware controller substantially improves MMT performance under dynamic conditions.

VIII. Conclusion

Teleoperation is a promising technology with applications such as telemaintenance and disaster management. However, it faces significant challenges when the application is subjected to a high network latency and dynamic environments. This work set out to extend Model Mediated Teleoperation (MMT) to overcome challenges in supporting dynamic environments with moving objects.

We propose to embrace the existence of mismatches between the local model and the remote environment and navigate the challenge by considering operator intent. To significantly enhance the scalability of MMT solutions, we advocate the use of available physics engines over handcrafted models. We have provided design principles and an accompanying framework for MMT solutions focusing on the human operator. We have applied our design principles and framework to the concrete application of guiding a robot arm to draw on a whiteboard, whose position is actively altered. We demonstrated the system's ability to replicate actions produced by the operator through objective measurements. We built this application on a system where the operator and remote domain are 8000km apart with an average end-to-end network latency of 165ms. Our user study underscores the efficacy of our approach by demonstrating a 3-point improvement on a 7-point Likert scale over network latencies of up to 1 s.

As we move forward, we believe that the future of teleoperation lies not just in refining models or decreasing latencies but in better understanding and designing for the human operator. This work marks a step in that direction, and it opens avenues for more adaptive, human-centric teleoperation solutions in the future.

References

[1] S. N. F. Nahri, S. Du, and B. J. Van Wyk, "A review on haptic bilateral teleoperation systems," *J. Intell. Robotic Syst.*, vol. 104, pp. 1–23, 2022.

[2] M. Simsek, A. Aijaz, M. Dohler, J. Sachs, and G. P. Fettweis, "5G-enabled tactile internet," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 3, pp. 460–473, Mar. 2016.

[3] F. Dressler, F. Klingler, M. Segata, and R. Lo Cigno, "Cooperative driving and the tactile internet," in *Proc. IEEE*, vol. 107, no. 2, pp. 436–446, Feb. 2019.

[4] G. P. Fettweis, "The tactile internet: Applications and challenges," *IEEE Veh. Technol. Mag.*, vol. 9, no. 1, pp. 64–70, Mar. 2014.

[5] A. A. Ateya et al., "Model mediation to overcome light limitations—toward a secure tactile internet system," *J. Sensor Actuator Netw.*, vol. 8, no. 1, 2019, Art. no. 6.

[6] P. Mitra and G. Niemeyer, "Model-mediated telemanipulation," *Int. J. Robot. Res.*, vol. 27, no. 2, pp. 253–262, 2008.

[7] X. Xu, B. Cizmeci, C. Schuwerk, and E. Steinbach, "Model-mediated teleoperation: Toward stable and transparent teleoperation systems," *IEEE Access*, vol. 4, pp. 425–449, 2016.

[8] J. Song, Y. Ding, Z. Shang, and J. Liang, "Model-mediated teleoperation with improved stability," *Int. J. Adv. Robotic Syst.*, vol. 15, no. 2, 2018, Art. no. 1729881418761136.

[9] K. Kroep, V. Gokhale, J. Verburg, and R. V. Prasad, "ETVO: Effectively measuring tactile internet with experimental validation," *IEEE Trans. Mobile Comput.*, vol. 23, no. 3, pp. 2054–2065, Mar. 2024.

[10] K. Kroep, V. Gokhale, A. Simha, R. V. Prasad, and V. S. Rao, "TIM: A novel quality of service metric for tactile internet," in *Proc. ACM/IEEE 14th Int. Conf. Cyber-Phys. Syst.*, 2023, pp. 199–208.

[11] N. Promwongsa et al., "A comprehensive survey of the tactile internet: State-of-the-art and research directions," *IEEE Commun. Surv. Tut.*, vol. 23, no. 1, pp. 472–523, First Quarter, 2021.

[12] S. K. Sharma, I. Woungang, A. Anpalagan, and S. Chatzinotas, "Toward tactile internet in beyond 5G era: Recent advances, current issues, and future directions," *IEEE Access*, vol. 8, pp. 56 948–56 991, 2020.

[13] Z. S. Bojkovic, B. M. Bakmaz, and M. R. Bakmaz, "Vision and enabling technologies of tactile internet realization," in *Proc. IEEE 13th Int. Conf. Adv. Technol. Syst. Serv. Telecommun.*, 2017, pp. 113–118.

[14] B. Willaert, H. Van Brussel, and G. Niemeyer, "Stability of model-mediated teleoperation: Discussion and experiments," in *Proc. Int. Conf. Haptics Percep. Devices Mobility Commun.*, Tampere, Finland, Springer, Jun. 13-15, 2012, pp. 625–636.

[15] P. Mitra, D. Gentry, and G. Niemeyer, "User perception and preference in model mediated telemanipulation," in *Proc. IEEE 2nd Joint EuroHaptics Conf. Symp. Haptic Interfaces Virtual Environ. Teleoperator Syst.*, 2007, pp. 268–273.

[16] X. Xu, C. Schuwerk, and E. Steinbach, "Passivity-based model updating for model-mediated teleoperation," in *Proc. 2015 IEEE Int. Conf. Multimedia Expo Workshops*, 2015, pp. 1–6.

[17] X. Xu, S. Chen, and E. Steinbach, "Model-mediated teleoperation for movable objects: Dynamics modeling and packet rate reduction," in *Proc. 2015 IEEE Int. Symp. Haptic Audio Vis. Environ. Games*, 2015, pp. 1–6.

[18] F. Ficuciello, L. Villani, and B. Siciliano, "Variable impedance control of redundant manipulators for intuitive human–robot physical interaction," *IEEE Trans. Robot.*, vol. 31, no. 4, pp. 850–863, Aug. 2015.

[19] L. Rozo, S. Calinon, D. G. Caldwell, P. Jimenez, and C. Torras, "Learning physical collaborative robot behaviors from human demonstrations," *IEEE Trans. Robot.*, vol. 32, no. 3, pp. 513–527, Jun. 2016.

[20] Y. Li and S. S. Ge, "Human–robot collaboration based on motion intention estimation," *IEEE/ASME Trans. Mechatronics*, vol. 19, no. 3, pp. 1007–1014, Jun. 2014.

[21] K. T. Ly, M. Poozhiyil, H. Pandya, G. Neumann, and A. Kucukyilmaz, "Intent-aware predictive haptic guidance and its application to shared control teleoperation," in *Proc. 30th IEEE Int. Conf. Robot Hum. Interactive Commun.*, 2021, pp. 565–572.

[22] A. T. Le et al., "Learning forceful manipulation skills from multi-modal human demonstrations," in *Proc. 2021 IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2021, pp. 7770–7777.

[23] Y. Michel, R. Rahal, C. Pacchierotti, P. R. Giordano, and D. Lee, "Bilateral teleoperation with adaptive impedance control for contact tasks," *IEEE Trans. Robot. Autom.*, vol. 6, no. 3, pp. 5429–5436, Jun. 2021.

[24] A. Hussein, M. M. Gaber, E. Elyan, and C. Jayne, "Imitation learning: A survey of learning methods," *ACM Comput. Surv.*, vol. 50, no. 2, pp. 1–35, 2017.

[25] J. Chen, C. Yi, S. D. Okegbile, J. Cai, and X. Shen, "Networking architecture and key supporting technologies for human digital twin in personalized healthcare: A comprehensive survey," *IEEE Commun. Surv. Tut.*, vol. 26, no. 1, pp. 706–746, First Quarter, 2024.

[26] J. Chen, Y. Shi, C. Yi, H. Du, J. Kang, and D. Niyato, "Generative AI-driven human digital twin in IoT-healthcare: A comprehensive survey. arxiv," 2024, *arXiv:2401.13699*.

[27] H. Xiang et al., "Realizing immersive communications in human digital twin by edge computing empowered tactile internet: Visions and case study," 2023, *arXiv:2304.07454*.

[28] K. Wu et al., "QoE-aware joint visual and haptic signal transmission with adaptive data compression for immersive interactions in human digital twin," *IEEE Trans. Netw. Service Manag.*, vol. 22, no. 3, pp. 2780–2794, Jun. 2025.

[29] R. Chaudhari, E. Steinbach, and S. Hirche, "Towards an objective quality evaluation framework for haptic data reduction," in *Proc. 2011 IEEE World Haptics Conf.*, 2011, pp. 539–544.

[30] S. Scherzinger, A. Roennau, and R. Dillmann, "Forward dynamics compliance control (FDCC): A new approach to cartesian compliance for robotic manipulators," in *Proc. 2017 IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 4568–4575.

[31] E. Coumans and Y. Bai, "Pybullet, a Python module for physics simulation for games, robotics and machine learning," 2016–2021. [Online]. Available: http://pybullet.org

[32] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, "Robot operating system 2: Design, architecture, and uses in the wild," *Sci. Robot.*, vol. 7, no. 66, 2022, Art. no. eabm6074. [Online]. Available: https://www.science.org/doi/abs/10.1126/scirobotics.abm6074

H.J.C. Kroep is currently working toward the PhD degree with the Networked Systems group, TU Delft. For the past seven years, he has worked on haptic bilateral teleoperation and tactile internet, focusing on predictive force feedback, ultra-low-latency communication, and edge computing. His contributions include novel performance metrics for Tactile applications, model-mediated teleoperation methods, and practical demonstrations of long-distance teleoperation over 5G networks.

P. Makridis received the MSc degree in computer science from the Delft University of Technology in 2024. He is a Machine Learning engineer with the Visual Computing Group, Sony Interactive Entertainment (PlayStation) in London. His interests lie in the intersection of computer graphics and machine learning, including applications in 3D reconstruction and rendering.

J. Huidobro received the MSc degree in computer science from TU Delft. Currently, he is a 3D Machine Learning researcher with ALLSIDES. His research interests span computer graphics and vision, with a focus on 3D generation and reconstruction.

K. Wösten is currently working toward the PhD degree with the Cognitive Robotics department, TU Delft. During the MSc in engineering, he contributed to research on haptic bilateral teleoperation, developing predictive control strategies and low-latency system architectures for immersive remote interaction. His broader interests include human–robot interaction, real-time control, and tactile internet technologies.

D. Choudhary is currently working toward the MSc degree and research assistant with Aalto University, Finland. His research interests primarily lie in systems and networking, specifically in software-defined networking and programmable data planes. Previously, he spent 2.5 years as a research assistant with the Centre for Networked Intelligence (CNI), Indian Institute of Science (IISc). His work there focused on leveraging programmable data planes to implement a Time-Sensitive Networking (TSN) switch.

N. Gnani is currently working toward the dual MSc degree in autonomous systems and intelligent robots with Université Côte d'Azur (Polytech Nice Sophia) and KTH Royal Institute of Technology. Before this, he spent more than five years with the Indian Institute of Science (IISc), Bangalore, as a research associate and technical associate with the Centre for Networked Intelligence, where he contributed to tactileinternet testbeds, ultralowlatency cyberphysical systems, IoT data management, and indoor localisation. His recent work includes developing an EdgeP4 innetwork edgeintelligence switch for tactile cyberphysical systems.

T.V. Prabhakar is a chief research scientist and leads the Zero Energy Networks Laboratory, the Indian Institute of Science in Bangalore. Over his career, he has architected networked embedded systems and microenergyharvesting platforms, applying them to IoT, indoor positioning, healthcare and smartcabin domains. His team has designed lowpower MAC and network protocols (including publish–subscribe and REST on 6LoWPAN), enabled subtwofoot RFID localisation, and built silicon-based RF energy harvesters. He has recently extended this expertise into tactileInternet applications, targeting submillisecond reliability and integrating mechanisms such as dynamic network slicing, edgeintelligent communication architectures, and deployment over emerging technologies like 5G and IEEE 802.11ay. He has authored over fifty refereed conference and journal articles, and his group's prototypes have earned bestpaper awards and national recognition.

S. Coppens received the MSc degree in embedded systems from TU Delft, where he contributed to research on haptic bilateral teleoperation over 5G as part of the Networked Systems group. He is an embedded engineer with Perciv AI in Delft, working at the intersection of hardware and software for robotics and radar perception.

K. Van Berlo received the MSc degree in embedded systems from TU Delft in 2022. During his master's thesis, he developed a realtime 6DoF object tracking system for the tactile internet, thereby advancing dynamic environment capture for modelmediated teleoperation. His current research interests are spiking neural networks (SNN), wireless communications, and control systems.

R. Venkatesha Prasad is an associate professor with the Networked Systems Group of TUDelft. His research interest is in the area of Tactile Internet, IoT, and 60 GHz mmWave networks. He has supervised 21 PhD students and more than 80 MSc students. He has more than 300 publications in peer-reviewed international journals, conferences, and standards, and book chapters. He is an associate editor on the editorial board of *IEEE Transactions on Mobile Computing*, *IEEE Transactions on Sustainable Computing*, and *IEEE Transactions on Cognitive Communications and Networking*. He was the vice-chair of the IEEE Tactile Internet standardisation workgroup and is now a mentor. For more information, please refer to http://st.ewi.tudelft.nl/rvprasad