Utilizing Operator Intent for Haptic Teleoperation under high latencies

Journal:	*Transactions on Mobile Computing*
Manuscript ID	TMC-2024-09-1467
Manuscript Type:	Regular (S1)
Keywords:	Teleoperation, URLLC, L.2.0.a Kinesthetic devices < L.2 Haptics Technology < L Haptics

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.P. Prabhakar, S. Coppens, R.R. Venkatesha Prasad, K. Van Berlo

Abstract—Bilateral teleoperation is a promising technology with applications such as telemaintenance and disaster management. However, it faces significant challenges when the application is subjected to simultaneously a high network latency and a dynamic environment with moving objects. This work aims to extend Model Mediated Teleoperation (MMT) to overcome its 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 $8000\,\mathrm{km}$ away (the Netherlands – India) and subjected to an average of $179\,\mathrm{ms}$ 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 second.

1 Introduction

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

In a haptic teleoperation setup, an operator engages with a device at the operator's end. The movements and manipulations performed on this device are then transmitted to the remote side. The remote robotic device imitates the operator's actions. The operator who then experiences feedback from the remote interactions. The feedback can be in the form of audio-visual and haptic feedback. The addition of haptic feedback, in particular, gives the operator the experience as if they were physically present in the remote environment and allows the operator to express actions in a more refined and natural manner.

In practice, realizing teleoperation over a network is challenging due to the delay imposed by the network. As the distance between operator and 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 $1\,\mathrm{ms}$ 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 ($\approx$ 150 km or $1\,\mathrm{ms}$ in terms of delay) supported due to fundamental physics-imposed limitations on network latencies. However, even without the maximum distance, conceptualizing any network with such low latency is profoundly challenging.

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 communicate actions to the remote robot, receiving instantaneous feedback based on the local model of the remote environment.

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.

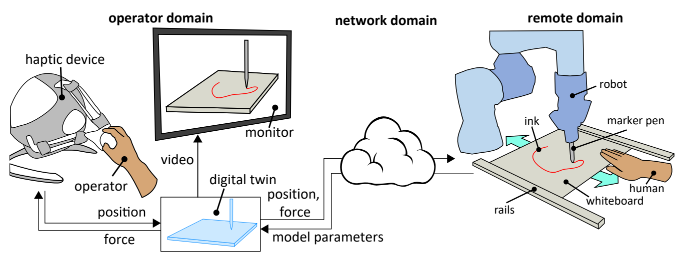


Fig. 1. We develop an approach allowing teleoperation over long distances. We demonstrate our effectiveness of approaches by creating the application shown in the figure. An operator uses a haptic device (in TUDelft the Netherlands) to guide a robot, equipped with a marker, to draw on a canvas situated on a different continent (IISc, Bangalore, India). Simultaneously, another individual alters the canvas's position as the operator makes the drawing. See https://drive.google.com/drive/folders/1kzseNW7iwhsAkVAYOEEOUp8AzoUtMxg9?usp=sharing for two videos.

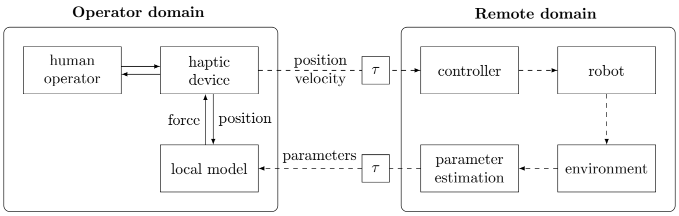


Fig. 2. General structure of a Model-Mediated Teleoperation system.

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 heavily depends on the local model being a faithful representation of the remote side. This is especially the case in dynamic environments since moving objects complicate updating the local model. 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 significantly improve the scalability of MMT solutions, we advocate utilizing physics engines that prioritize speed over accuracy and include methods to deal with the consequences of these 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 whiteboard over an 8000 km distance (TUdelft in the Ntherlands and IISc, Bangalore, India). See the video[1].

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

The rest of this work is structured as follows. Section 2 offers an overview of related literature. Section 3 outlines design principles aimed at enhancing MMT. Section 4 introduces a mathematical framework for MMT applications. Section 5 applies the framework to a specific application. Section 6 details the experimental setup of our user study. Section 7 presents an analysis of the user study results. Finally, Section 8 summarizes and concludes our findings.

2 Related Works

2.1 Tactile Internet

Multiple empirical studies demonstrate that teleoperation applications where force feedback is transmitted over the network requires sub 10 ms 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 achieve a reduction in latency, which advances 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 small, but extremely latency constraint, making SDN and NFV ideal solutions to provision for this type of traffic.

2.2 Model Mediated Teleoperation

MMT solves the problem of performing teleoperation with significant network latency. MMT, however, poses 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, leading to an undesirable experience, and an undesirable control signal being 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 unintentionally cause 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.

2.3 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 field of MMT, it has been

1. https://drive.google.com/drive/folders/1kzseNW7iwhsAkVAYOEEOUp8AzoUtMxg9?usp=drive_link

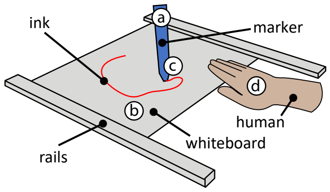


Fig. 3. Illustration of an application where a marker draws on a whiteboard while a human moves the whiteboard between two rails.

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.

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], [21], [22]. 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, there are imitation learning techniques that aim to make AI agents behave as a human would when presented with the same scenario as the AI is currently in [23].

3 Design Principles for human centered MMT

In this section, we present design principles to make a system for user centered teleoperation. It is important to first understand why an understanding of operator intent is needed in the first place. The reason is that the operator is operating under conditions that differ from the reality when the robot is mimicking the operators actions. This effect is exacerbated at increases in latency. When working with a local model, there can also be differences due to inaccurate measurements of the remote environment or limitations in the simulation accuracy.

It is important to minimize these differences, but it is unrealistic to require a design that does not have any mismatch between the operator and remote side. Therefore, we propose an approach where the mismatch is embraced, and solutions are placed to address the consequences of these mismatches. We propose to consider the operator intent and perception to provide the context to optimize the choices the robot makes. In Section 5, we validate these design principles with a concrete application and user study.

3.1 Mismatches

If the operator were perceiving the remote environment with high accuracy and without latency, there would be no need for the analysis presented here. Thus, the reason why the operator intent and perception need to be analyzed is because the perceived environment and real environment have mismatches. We first need to consider the nature of those mismatches we can expect.

Measurement noise: Feedback to the operator is based on measurements made in the remote domain. When using model-mediated feedback, the simulation requires a virtual replica based on observations from the remote environment. This replica will have inaccuracies due to both measurement noise and the inevitable simplifications used.

Simulation noise: The deployed simulation cannot replicate a real environment accurately enough to predict the future. Additionally, we advocate for using simpler, faster engines commonly found in video game design to optimize computational latency. As a result, objects will not respond exactly as simulated. This includes limitations in the complexity of objects; for example, if fluids are involved, accuracy can be expected to be very low.

External influences: The remote environment can feature influences not captured by the model. These could be events happening outside the scope of the constructed model or unpredictable active participants on the other side, such as humans.

Latency: Finally, there is latency. The key effect of latency is amplifying the negative effects of all other influences. An inaccurate simulation that predicts only a few milliseconds before being corrected will have a considerably smaller mismatch than a correction after 100 ms.

3.2 Capturing Operator intent

We introduce the notion of operator intent, which, within teleoperation, signifies how an operator would act if they were directly in the remote environment instead of the experience that was provided to the operator by the teleoperation system. It is not necessary to fully grasp operator intent to have an opportunity to improve the user experience. However, considerable improvements in user experience can be achieved by designing for operator intent.

In a simple setup, the only transmitted information would be the trajectory of the haptic device manipulated by the operator, without a clear way of extracting additional info. Because the operator is interacting with a local simulation, there is an opportunity to capture significantly more information. We separate out information about the operator's behavior, and the objects behavior separately.

Operator's behavior: By engaging with a local model, the operator can express their trajectory and the force they apply simultaneously. Thus, the operator's actions encompass information on both position and force. Not only that, but the operator's position and force can also be considered in the context of the local environment. The operator's trajectory can also be considered in relation to any object in the simulation environment. Similarly, the amount of applied force can be considered relative to the object's in the environment.

Object's behavior: The consequences of the operator's actions on the objects in the environment can be separately observed. Similar to the operator's behavior, these can be expressed as a global trajectory or as the trajectory relative to other objects in the environment. Furthermore, the applied forces to these objects can be captured too. Additional things can be considered, like whether an object falls, breaks, or is deformed.

Hard transitions In many situations, even a slight change in position or force can make a significant difference. For example, the difference of pressing a button or not, or the difference between an object lingering on the edge of a table or falling over. If these hard transitions are present and not directly tracked, a high level of accuracy is needed to make sure the operator experience and reality in the remote domain do not converge significantly.

Identifying hard transitions and separately ensuring that they are consistent can be a good way to avoid the need for high levels of accuracy for the entire application, while still maintaining a good user experience.

3.3 Operator Perception

Besides operator intent, another key concept to consider is operator perception. In the context of teleoperation, the operator perception is described as to what extent the operator can perceive a deviation of the imitation to the demonstration and how this affects the user experience.

The controller in the remote domain attempts to manipulate the robot to imitate the operator's intent. In order to do so, the controller requires some room for deviations. These deviations are used to improve the realization of operator intent. The more freedom the controller has to alter the trajectory of the robot, the easier it becomes to align the environments. However, it is important to ensure that the deviations are only marginally detectable by the human operator.

One must consider the operator's perception of these deviations. Recognizing differences between the operator's original actions and the imitated trajectory is not as simple as quantifying the deviation using metrics like the root mean square error (RMSE) between the original and imitated trajectories. This has been demonstrated in multiple prior works [9], [10], [24]. Humans perceive deviations in a more nuanced manner. For instance, while differences like low-frequency deviations in absolute position and variations in scale might be challenging to detect, the difference between hovering over and hitting an object or small high-frequency vibrations become immediately apparent. Furthermore, how deviations are perceived and impact the user experience can be subjective and vary between operators.

3.4 Scalability

Traditional model-mediated teleoperation (MMT) methods often use handcrafted solutions, mainly due to their aim to achieve precise alignment between the virtual and remote environments. However, as we have identified the potential to accommodate a certain degree of mismatch, this opens doors for alternative approaches to representing the remote environment.

We advocate for the use of real-time physics engines. These engines, already prevalent in the robotics and computer graphics fields, can model a diverse array of scenarios in real time. For operator interactions in an MMT system, the level of accuracy provided by a physics engine is sufficient. This is primarily because the operator's perception cannot detect minor inaccuracies, provided the simulated interaction is plausible.

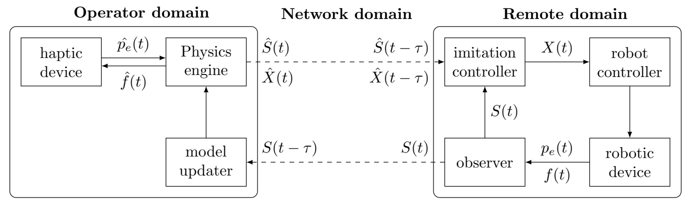


Fig. 4. Illustration of our proposed framework that aims to extend MMT to work with complex and dynamic environments by considering operator intent.

Furthermore, leveraging physics engines presents a set of pragmatic advantages. They are often supported by extensive communities, many are open-source, and they can be adapted to specific requirements.

In Section 5, we describe how we deploy a physics engine to construct the application used in our user study.

4 A Framework for MMT with Operator Intent

In this section, we outline a framework for designing a system for teleoperation over long distances. The system is built on the MMT system design and emphasizes the design principles given in the previous section. An overview of the framework is illustrated in Fig. 4. The system features three main parts (domains): The Operator Domain, Remote Domain, and Network Domain. Throughout this work, for any parameter θ in the remote domain, we use $\hat{\theta}$ to denote its counterpart in the operator's domain.

We denote S as the observed state of the environment in the remote domain. The state of the environment includes attributes like object location, orientation and shape, mass, friction coefficients, center of mass, and inertia. These properties are either accurately provided as priors or are provided with an initial guess which can be improved with subsequent observations in the remote domain at runtime.

In the operator domain, a physics engine enables operators to engage with a digital twin of the distant environment. The digital twin of the environment in the remote domain used by the physics engine in the operator domain is denoted as $\hat{S}$.

The operator manipulates a haptic device to interact with the local physics engine. The haptic device measures only the position of its end-effector, which is the endpoint on the robotic arm, and is denoted as $\hat{\boldsymbol{p}}_e$. A haptic rendering algorithm within the physics engine converts the position of the end-effector to a position in the virtual environment and an applied force. The description of the operator's state in the virtual environment is the control signal $\hat{X}$. The predicted applied force is indicated as $\hat{\boldsymbol{f}}$, which is fed back to the operator without network delay. With this, we can consider the physics engine as a function that modifies the state of the environment and the operator represented by

$$\left(\hat{S}(t), \hat{X}(t)\right) = \text{physicsEngine}\left(\hat{S}(t-e), \hat{X}(t-e), \hat{\boldsymbol{p}}_e(t)\right),$$

where $t-e$ indicates the previous discrete step of the physics engine.

Both a full description of the virtual state $\hat{S}$ and the operator $\hat{X}$ are transmitted to the remote domain. The data arrives with an added network latency of τ. The imitation controller considers the delayed state in the operator domain and the state of the local environment to modify the control signal. Then we get

$$X(t) = \text{imitationController}\left(\hat{X}(t-\tau), \hat{S}(t-\tau), S(t)\right).$$

Note that the imitation controller should be designed so that if there is no mismatch between the two states, the imitation controller should not modify the control signal provided by the operator. This means that when $S(t) = \hat{S}(t-\tau)$, one gets $X(t) = \hat{X}(t-\tau)$. However, When the two states have mismatches, the imitation controller is responsible for modifying the control signal to prioritize the realization of operator intent. The strategy for discerning operator intent and operator perception can be determined beforehand and used to design the imitation controller.

The control signal is used to drive the robot controller, which covers any intricacies related to the used robotic device. The robot controller is completely agnostic to the considerations of operator intent and only considers the output of the imitation controller. The robotic device measures the position of the end-effector $\boldsymbol{p}_e$ and optionally the force applied to it as $\boldsymbol{f}$. The observer is a collection of sensors in the remote domain that track the realtime position, orientation, and motion of every object in the environment. Combined with the measurements from the robotic device, the observer constructs an estimation of the state of the remote environment S. It is important that object tracking is done with high accuracy and low latency, especially when the tracking information is directly being used in the control strategy in the imitation controller.

The observed state of the remote environment is sent back to the operator domain, where it arrives with τ network delay. Audio, video, and force measurements that result from active remote interactions can be included in the feedback and should be immediately relayed back to the operator. The measured state of the remote environment S is used to update the digital twin in the operator domain $\hat{S}$. We get

$$\hat{S}(t) = \text{modelUpdater}\left(S(t-\tau), \hat{S}(t), \hat{X}(t)\right).$$

The update strategy should be designed so that it minimally disturbs the operator. The updating strategy can involve postponing model updates when the operator is actively interacting with an object, which has been shown to have potential [15], [16].

The network domain encompasses all system elements that synchronize data between the operator and remote domains. Teleoperation applications feature multiple modalities with highly varying requirements. Kinematic data, which captures object positions and orientations, has stringent latency requirements but is compact, taking up only a few bytes per object in the environment. Typically, this type of data is highly resistant to data loss because subsequent packets remove the need to retransmit prior ones. Kroep et al. demonstrated a teleoperation setup with satisfactory user experience using a network with 50% packet loss [9]. This is further amplified by the fact that the controller of the robot arm serves as a low pass filter both due to the acceleration limitations of the robot and the lower update rate (60 Hz against a 1 kHz packet rate). Conversely, data detailing objects' shapes, physical attributes, and audio-visual content have a larger payload but are significantly more tolerant of latency while requiring high reliability. Therefore, the network must proficiently manage diverse data types, ensuring high-volume transmission while adhering to the varying latency demands of each attribute. Especially in complicated environments, data generation may outpace available bandwidth, necessitating prioritization of vital data based on the operator's actions and vicinity to objects. For this reason, adept protocols, efficient bandwidth utilization, data compression, and priority management should be considered in the network design.

5 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. 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 3.

5.1 Mismatches

The first step is to identify the mismatches that can occur in this application. This will be done as described in Section 3.1

measurement noise: In this case, 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 latency of 179 ms. In particular the external influences will be communicated with this latency, causing challenges to make the application work. In a simple implementation the whiteboard position relative to the marker will be consistantly different between operator and remote domain, causing any drawing to be considerably deformed.

5.2 Capturing operator intent

The next step is to identify operator intent as described in Section 3.2.

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

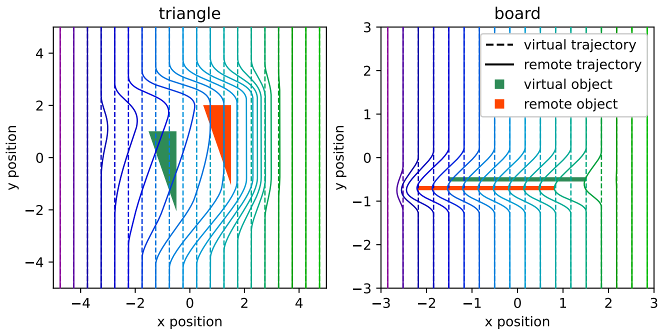


Fig. 5. Illustration of the behavior of the imitation controller design. For a given set of virtual trajectories of $\hat{\boldsymbol{p}}_p$ in the operator domain, the modified trajectory of $\boldsymbol{p}_p$ in the remote domain is shown.

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.

Hard transitions: In this case, the most important hard 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 $1\,\mathrm{mm}$ can cause a significant difference.

5.3 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.

5.4 Translation between absolute and relative trajectory

Based on the insights obtained previously, we identify the key mismatch to consider 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 seamless, ensuring that any movement in the operator domain corresponds to a monotonically increasing movement in the remote domain. In other words, any motion in the operator domain should not result in a contrary movement in the remote domain.

In this section, we describe how to design an imitation controller that adjusts the measured behavior of the operator to respect the operator intent in our specific application. $\hat{X}$ describes the position of the end effector that directly corresponds with the haptic device in the operator domain. We take X_{relative} as the position in the remote domain that has the same position related to the whiteboard as in the operator domain. Next, a transition factor α is chosen to smoothly transition between $\hat{X}$ and X_{relative}. We get

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

Note that $\hat{X}(t-\tau)$ includes the communication delay τ.

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,

$$X_{\text{mod}}(t) = \{\hat{\boldsymbol{p}}_p(t-\tau) + \boldsymbol{p}_o(t) - \hat{\boldsymbol{p}}_o(t-\tau), \hat{\boldsymbol{f}}(t-\tau)\}, \tag{2}$$

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 of 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

$$\hat{\boldsymbol{p}}_{\min} = \arg\min_{\boldsymbol{p} \in P} (\boldsymbol{p} + \hat{\boldsymbol{p}}_o - \hat{\boldsymbol{p}}_p). \tag{3}$$

To smoothly transition between $\hat{X}$ and X_{mod}, we design a smooth transition function that is used over the transition region to calculate the transition factor α. As a transition function, we introduce a cubic spline as

$$g(x) = \begin{cases} 0 & \text{if } x \leq 0, \\ 1 & \text{if } x \geq 1, \\ 3x^2 - 2x^3 & \text{otherwise,} \end{cases} \tag{4}$$

where x is any input to the function g, in this case correlated to the distance between the operator position and the object.

The transition region should be chosen as such that the monotonicity condition is preserved. This means that any movement in the operator domain will not lead to any movement in the opposite direction in the remote domain. This condition holds when $\frac{d}{dt}\hat{\boldsymbol{p}}_p(t-\tau) \cdot \frac{d}{dt}\boldsymbol{p}_p(t) > 0$.

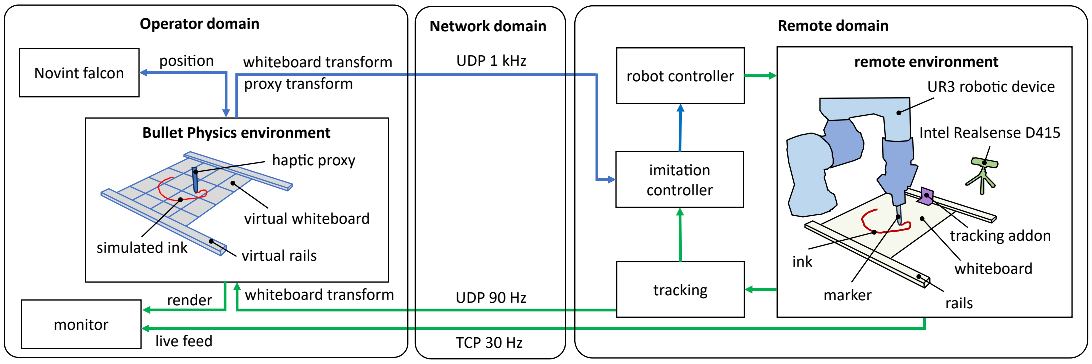


Fig. 6. Schematic overview of experimental setup. In the figure, we lay down the different components of our setup, showcasing how they relate to the proposed teleoperation framework and how data flows through the system. Blue arrows indicate high-frequency communication of $1\,\mathrm{kHz}$, while green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

The critical direction that determines whether the monotonicity condition is met is when $\frac{d}{dt}\hat{\boldsymbol{p}}_p$ and $\hat{\boldsymbol{p}}_{\min}$ are in the same direction. Because the maximum slope of the transition function in Eq. (4) is 1.5, the transition needs to be at least $1.5|\boldsymbol{p}_o - \hat{\boldsymbol{p}}_o|$ when $|\frac{d}{dt}\hat{\boldsymbol{p}}_p \times \hat{\boldsymbol{p}}_{\min}| = 0$ and $\frac{d}{dt}\hat{\boldsymbol{p}}_p \cdot \hat{\boldsymbol{p}}_{\min} > 0$ to guarantee monotonicity. Here, we denote $|\cdot|$ as the l^2 norm of a vector.

In this work we use a transition length of $|\boldsymbol{p}_o - \hat{\boldsymbol{p}}_o|$ in all but the critical direction and $2|\boldsymbol{p}_o - \hat{\boldsymbol{p}}_o|$ in the critical direction. Thus, we can get the transition factor as

$$\alpha = \begin{cases} g\left(\frac{|\hat{\boldsymbol{p}}_{\min} - \frac{1}{2}\mathrm{proj}_{(\boldsymbol{p}_o - \hat{\boldsymbol{p}}_o)}\hat{\boldsymbol{p}}_{\min}|}{|\boldsymbol{p}_o - \hat{\boldsymbol{p}}_o|}\right) & \text{if } \hat{\boldsymbol{p}}_{\min} \cdot (\boldsymbol{p}_o - \hat{\boldsymbol{p}}_o) > 0, \\ g\left(\frac{|\hat{\boldsymbol{p}}_{\min}|}{|\boldsymbol{p}_o - \hat{\boldsymbol{p}}_o|}\right) & \text{otherwise,} \end{cases} \tag{5}$$

where $\mathrm{proj}_{\boldsymbol{b}}\boldsymbol{a}$ is the projection of $\boldsymbol{a}$ onto $\boldsymbol{b}$. Finally, we can use Eq. (2), (4), and (5) in (1) to obtain the control signal for 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 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.

5.5 Controller design

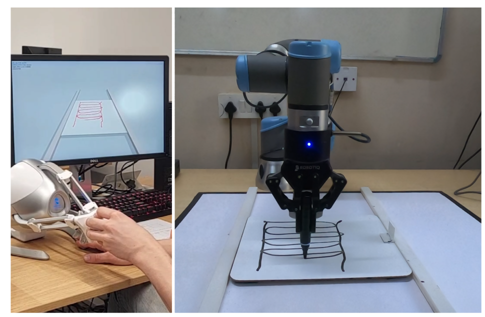
It is common for a robotic device in these types of applications to be controlled with a compliance controller that makes use of an accurate force-torque sensor [25]. In this application, there are no remotely initiated interactions that need to be relayed to the operator, as the alteration of the whiteboard 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 [25]. The applied force $\boldsymbol{f}$ is converted into a positional offset. We can calculate the target position fed to the position controller as

$$\boldsymbol{p}_{\mathrm{target}} = \boldsymbol{p}_p + \frac{1}{k_s}\boldsymbol{f}. \tag{6}$$

In this case the position of the end-effector p_p is the modified version obtained using (1) and (5). Note that for safety, $\boldsymbol{p}_{\mathrm{target}}$ is restricted within an operating range to avoid undesirable control signals when someone accidentally hits the Novint Falcon. Also note the spring constant k_s. The value of 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 Novint Falcon end-effector's proxy in the operator domain contacts the whiteboard without exerting force, its corresponding position in the remote domain remains $1\,\mathrm{mm}$ 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.



(a) (b)

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.

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 Eq. (6). In the second control strategy, the mismatch in the whiteboard position between the operator and remote domains is considered. Here, Eq. (1) and Eq. (6) 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 $X(t)$.

6 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 in a Western European institution and the remote domain in an Asian institution[2]. An overview of the application is provided in Fig. 6.

In the operator domain, the Bullet-Physics engine provides the local simulation [26]. We have adapted the physics engine to support a haptic rendering algorithm and interface with the 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 $1\,\mathrm{kHz}$. The physics engine is decoupled from the rendering engine so that the physics engine can run at $1\,\mathrm{kHz}$ while the OpenGL-based renderer runs at $60\,\mathrm{Hz}$. This ensures a smooth and responsive haptic response from the physics engine with sub $1\,\mathrm{ms}$ computational delay. A photo of the operator domain is shown in Fig. 7(a).

In the remote domain, we deploy a UR3 robot. Mounted on the end-effector of the UR3 is a gripper that holds a marker. Two rails secured to the table lock a whiteboard in a 1 DoF motion towards the base of the UR3. Pieces of square sponge and felt pads are attached on the bottom of the corners of the whiteboard to serve as a suspension of $4\,\mathrm{mm}$ until fully compressed. A ROS2 environment communicates directly with the UR3 [27]. Fixed to the movable whiteboard is a small 3D-printed part. An Intel Realsense D415 camera captures RGB-D images from the side and is used to track the 3D-printed part attached to the whiteboard. Tracking happens at $90\,\mathrm{Hz}$. 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.

6.1 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 easily 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.

2. More specifics on the institutions will be revealed on publication.

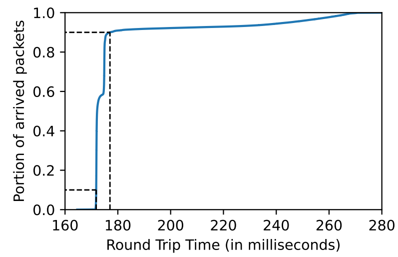


Fig. 8. Cumulative distribution of end-to-end network latency measured over 15 hours. $80\,\%$ of the packets have latencies between $172\,\mathrm{ms}$ and $177\,\mathrm{ms}$. The average latency is $179\,\mathrm{ms}$.

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, $90\,\mathrm{Hz}$ 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 model updater's only task is 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 $0.8\,\mathrm{m/s}$. 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.

6.2 Network

The operator and remote domains are separated by an approximate distance of $8000\,\mathrm{km}$. Kinematic and force data from the operator, as well as kinematic data from the whiteboard, are relayed over a UDP channel at $1\,\mathrm{kHz}$ and $90\,\mathrm{Hz}$, respectively. Both feedforward and feedback packets carry a 100-byte payload. 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

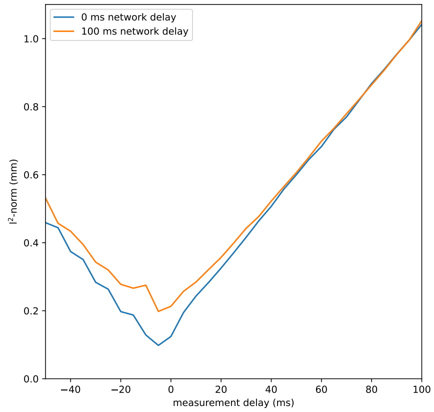


Fig. 9. This image shows the effect of network delay and measurement delay of the whiteboard on the error of the reproduced image. Negative measurement delay means prediction is used. If there is measurement delay, the robot cannot compensate for the movement in the whiteboard, resulting in an incorrect 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.

computer in the operator domain at a maximum rate of 30 Hz. The Round-Trip Time (RTT) for the UDP link was assessed over 10 hours, and the results are depicted in Fig. 8. The RTT reveals that 80 % of packets arrive between 172 ms and 177 ms. Last, to push the network further in our experiments, we used NetEM to increase the latency by 1 s.

6.3 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 hardcoded 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 effectively separates the effects of time mismatches and position mismatches [3]. It quantifies image quality using an ℓ^2-norm.

The results are shown in Fig. 9. 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 in a 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. While this 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.

6.4 User Study

In the user study, participants use the experimental setup to draw pictures in a remote environment under varying conditions. First, 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.

③ *Render with moving whiteboard, without imitation 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 imitation controller, increased latency*: The participant only observes the local render in the operator domain. The whiteboard is in constant motion. The imitation controller considers the movement of the whiteboard. For scenario ③ and ④ an artificial latency of 1 S 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

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

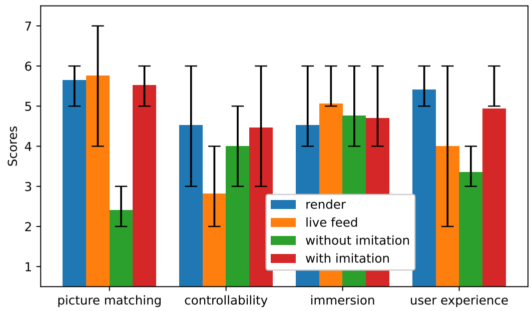


Fig. 10. The results of the user study. Four scenarios are rated on four categories using a 7-point Likert scale.

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.

7 Performance analysis

Fig. 10 showcases the overall results of the user study. The first two scenarios indicated as render and live feed, are identical except for the type of video feedback provided. In the first case, the local render is used, while in the second case, the live feed of the remote environment is used. Note that in both cases, the virtual environment provides instantaneous force feedback. In the scenario with the live feed, the participant experiences the force feedback of pressing on the whiteboard before he can visually see the marker in the remote domain touching the whiteboard.

There are two areas in which the live feed outperforms the local render. The first one is in picture matching. This outcome is anticipated since the direct footage of the remote environment is utilized. The only thing limiting the picture matching is the quality of the video feed. Conversely, the local render is only an approximation of the drawing in the remote domain. Furthermore, the mechanics of the robot controller filter out some of the lower frequencies in the operator trajectory.

Compared to observing a locally rendered approximation, a direct video feed from the remote environment was perceived to be more immersive. While realistic matching between the local renderings was not a focus of this work and can be significantly improved in the future, we expect the immersion of the real footage to not be exceeded by a local render in the near future.

Inference 1. The live feed offers a stronger immersion and perception of the true state of the remote environment.

Feedback from participants revealed a significant increase in perceived control when operating with the local render for visual feedback. Participants also remarked on the need to operate at a significantly slower pace when using the live feed to execute tasks. This behavior of using slow, methodical motions can help counteract the latency in the video feed. The difference in controllability translated to a strong preference for the local render experience, which was rated as markedly more desirable than its live feed counterpart.

Inference 2. The local render offers significantly better controllability resulting in a superior user experience.

For the next set of experiments, the whiteboard was continuously in motion, fully exposing the challenges of teleoperating in a dynamic environment with MMT. We also added 1 second of additional end-to-end network latency. Two control methods were assessed: (i) disregarding the whiteboard's relative position and (ii) taking the relative position into account. No discernible difference in controllability or immersion during task execution was observed between the two methods. However, after the task, there was a clear difference when observing the final drawing in the remote domain. Participants noted that when using the absolute control strategy, the final drawing significantly deviated from the drawing made in the operator domain. Participants indicated that it was more desirable to have a smooth experience during the task and have a mismatched result than to face the challenges of drawing with delayed visual feedback and getting an accurate outcome. Consequently, the absolute control strategy with the moving whiteboard was rated the worst overall experience.

Inference 3: As expected, a rudimentary implementation of MMT fails to handle dynamic environments, leading to large mismatch between what the operator was trying to achieve and what occurred in the remote side.

In stark contrast, when the imitation controller was deployed, participants observed hardly any difference between their drawing in the operator domain and the resulting drawing in the remote domain. Consequently, this scenario with a moving board was rated with a high overall experience, above the experience with live feed where the board was stationary. This demonstrates the efficacy of our imitation controller.

Inference 4: The performance of MMT approaches in dynamic conditions can be significantly improved by enhancing it with the capability to capture and preserve operator intent.

8 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 simultaneously a high network latency and a dynamic environment with moving objects. This work set out to extend Model Mediated Teleoperation (MMT) to overcome its 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 that focus 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 through objective measurements the system's ability to replicate actions produced by the operator. We built this application on a system where the operator and remote domain are $8000\,\mathrm{km}$ apart with an average end-to-end network latency of $165\,\mathrm{ms}$. 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 $1\,\mathrm{s}$.

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," *Journal of Intelligent & Robotic Systems*, vol. 104, pp. 1–23, 2022.

[2] M. Simsek, A. Aijaz, M. Dohler, J. Sachs, and G. P. Fettweis, "5G-Enabled Tactile Internet," *IEEE Journal on Selected Areas in Communications (JSAC)*, vol. 34, no. 3, pp. 460–473, 3 2016.

[3] F. Dressler, F. Klingler, M. Segata, and R. Lo Cigno, "Cooperative Driving and the Tactile Internet," *Proceedings of the IEEE*, vol. 107, no. 2, pp. 436–446, 2 2019.

[4] G. P. Fettweis, "The tactile internet: Applications and challenges," *IEEE Vehicular Technology Magazine*, vol. 9, no. 1, pp. 64–70, 2014.

[5] A. A. Ateya, A. Muthanna, A. Vybornova, I. Gudkova, Y. Gaidamaka, A. Abuarqoub, A. D. Algarni, and A. Koucheryavy, "Model mediation to overcome light limitations—toward a secure tactile internet system," *Journal of Sensor and Actuator Networks*, vol. 8, no. 1, p. 6, 2019.

[6] P. Mitra and G. Niemeyer, "Model-mediated telemanipulation," *The International Journal of Robotics Research*, 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," *International Journal of Advanced Robotic Systems*, vol. 15, no. 2, p. 1729881418761136, 2018.

[9] H. Kroep, V. Gokhale, J. Verburg, and R. V. Prasad, "Etvo: Effectively measuring tactile internet with experimental validation," *IEEE Transactions on Mobile Computing*, 2023.

[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 *Proceedings of the ACM/IEEE 14th International Conference on Cyber-Physical Systems (with CPS-IoT Week 2023)*, 2023, pp. 199–208.

[11] N. Promwongsa, A. Ebrahimzadeh, D. Naboulsi, S. Kianpisheh, F. Belqasmi, R. Glitho, N. Crespi, and O. Alfandi, "A comprehensive survey of the tactile internet: State-of-the-art and research directions," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 1, pp. 472–523, 2020.

[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 *2017 13th International Conference on Advanced Technologies, Systems and Services in Telecommunications (TELSIKS)*. IEEE, 2017, pp. 113–118.

[14] B. Willaert, H. Van Brussel, and G. Niemeyer, "Stability of model-mediated teleoperation: Discussion and experiments," in *Haptics: Perception, Devices, Mobility, and Communication: International Conference, EuroHaptics 2012, Tampere, Finland, June 13-15, 2012. Proceedings, Part I*. Springer, 2012, pp. 625–636.

[15] P. Mitra, D. Gentry, and G. Niemeyer, "User perception and preference in model mediated telemanipulation," in *Second Joint EuroHaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (WHC'07)*. IEEE, 2007, pp. 268–273.

[16] X. Xu, C. Schuwerk, and E. Steinbach, "Passivity-based model updating for model-mediated teleoperation," in *2015 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*. IEEE, 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 *2015 IEEE International Symposium on Haptic, Audio and Visual Environments and Games (HAVE)*. IEEE, 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 Transactions on Robotics*, vol. 31, no. 4, pp. 850–863, 2015.

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

[20] Y. Li and S. S. Ge, "Human–robot collaboration based on motion intention estimation," *IEEE/ASME Transactions on Mechatronics*, vol. 19, no. 3, pp. 1007–1014, 2013.

[21] A. T. Le, M. Guo, N. van Duijkeren, L. Rozo, R. Krug, A. G. Kupcsik, and M. Bürger, "Learning forceful manipulation skills from multi-modal human demonstrations," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 7770–7777.

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

[23] A. Hussein, M. M. Gaber, E. Elyan, and C. Jayne, "Imitation learning: A survey of learning methods," *ACM Computing Surveys (CSUR)*, vol. 50, no. 2, pp. 1–35, 2017.

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

[25] S. Scherzinger, A. Roennau, and R. Dillmann, "Forward dynamics compliance control (fdcc): A new approach to cartesian compliance for robotic manipulators," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 4568–4575.

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

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

Biographies

Kees Kroep received an MSc degree in electrical engineering at Delft University of Technology in 2018. Currently he is working toward his Ph.D. degree in Embedded and Networked Systems group at EEMCS, TU Delft. His Ph.D.work focuses on edge computing for tactile internet. His research interests include tactile internet and haptics technology.

R. Venkatesha Prasad is an associate professor at 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 63 MSc students. He has over 300 publications in the peer-reviewed international journals and 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 Transaction on Cognitive Communication and Networking. He was the vice-chair of IEEE Tactile Internet standardization workgroup and now a mentor. For more information, please refer to http://homepage.tudelft.nl/w5p50

Title: Utilizing Operator Intent for Intercontinental Bilateral Teleoperation

To the Editor in chief,
Trans on Mobile Computing

We thank the reviewers for their constructive in-depth reviews on the submission TMC-2024-04-0599 titled "Utilizing Operator Intent for Intercontinental Bilateral Teleoperation". We have used the feedback to create a new manuscript that, in our opinion, is a considerable improvement to the original submission. This document indicates what & how we have incorporated the comments from the reviewers in the revised manuscript. In black, we give the reviewers' comments; in blue is our response. In the revised article, we have provided a version that has text marked in blue to indicate areas of significant changes from the original. However, we have also thoroughly polished the paper, and many isolated changes have not been marked for readability.

Reviewer: 1

Comments:
In the manuscript, the authors are presenting an extended Model Mediated Teleoperation method that utilizes the operator's intention to allow for complex and dynamic environments in the presence of mismatches between the local model and the remote environment caused by network latency.
The idea in this work is interesting for the readers, and the topic of this work is relevant to the IEEE TMC. However, the presentation and structure of the work need to be improved since some parts lack details and some are difficult to follow.

Comment 1.1: As major comments:
In section 3.1.0.1, the authors listed the operator’s behavior as their design principles. Similarly, in their contribution, the authors stated that physics engines are optimized for speed. Hence, in Section 5, which part is related to the operator’s behavior for quick and purposeful movements. Additionally, the authors list five contributions in the manuscript. I cannot identify the optimized contemporary physics engines and the imitation controller are described in which section (Maybe Section 5 as a combination). So I suggest a subdivision of Section 5 or a summary of the contribution.

Response 1.1: We thank the reviewer for this insightful comment. We advocate using contemporary physics engines designed to compute complex scenes with plausible but not necessarily accurate outcomes. These systems are found in current-generation engines that are used to make video games. We are not building a physics engine from scratch; instead, we are modifying an existing physics engine, in this case, the *bullet physics engine*, to support haptic interactions.

One should pursue significant increases in the accuracy of the physics engines in use. However, we specifically advocate that fast and inaccurate physics engines be used and that the consequences of inaccurate predictions are to be handled with methods dedicated to this purpose.

In order to address the comment in the paper, the relevant contribution in Section 1 has been reformulated. Additionally, the first few sentences of Section 5.1 are reformulated to more clearly state what the imitation controller does.

The contribution in Section 1 now states: To significantly improve the scalability of MMT solutions, we advocate utilizing physics engines that prioritize speed over accuracy and include methods to deal with the consequences of these inaccuracies.

Comment 1.2: Section 5, on page 5, right column, lines 20-25. The author mentioned that the hard transitions are important in the remote drawing task. However, I can't find any algorithm to realize the hard transitions. Maybe the controller design, Section 6.1, contains the method of hard transitions, but I'm not sure. If so, please add the phrase "hard transitions" into the corresponding subsections to make it clearer. If not, please add the algorithm or references to realize the hard transitions.

Response 1.2: The reviewer has made some excellent comments about clarity in Section 5. We agree with the reviewer's concerns, and thus, both Section 3 and Section 5 have been rewritten to incorporate a better structure. This includes a more elaborate mention of intricate transitions.

Among the improvements we made is the final paragraph of Section 5.4, which now reads: The result of deploying this method is that whenever the end-effector is in close proximity to the object, its relative position is prioritized. This means 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.

Comment 1.3: The author should improve the structure of the manuscript. Especially for Section 5, which only contains one subsection. For the beginning part of Section 5 (on page 5, right column, and page 6, left column, lines 16-28), I think the description is rather redundant. I suggest a simplification of this part (at most within 10-15 lines) and put some useful information in Section 3. For the subsections, I suggest giving subsections in Section 5 according to the design principles mentioned in Section 3. Furthermore, I wonder if it will be better to put the controller design (Section 6.1) into Section 5.

Response 1.3: The reviewer has made some excellent suggestions about clarity in Section 5. To address this, Section 3 has been rewritten to have a considerably more general purpose and a more explicit set of design principles. Following the reviewer's suggestion, Section 5 has been rewritten with a new structure to follow the design principles described in Section 3.

Comment 1.4: The author should give more details on the experiments and results. In the objective drawing quality estimation experiment (on page 9, left column, lines 1-15), the author uses the l2-norm to quantify image quality. I suggest adding the details on the qualitative description of the l2-norm (e.g., a high value of norm means low/high image quality). Additionally, the author should add the trends and the analysis for the reason for the trends in Figure 9. For example, why does the norm become larger when the measurement delay (negative) becomes larger (from -5 ms to -40 ms)? Vital information can be marked in Figure 9 to make it clearer (e.g., 5 ms of prediction). Furthermore, there is no information showing how the data are collected in Figure 9. For example, for the horizontal axis (measurement delay), do you calculate the norm every 5 ms delay? In the user study (Section 6.5), some details of the participants should be given. For example, the number, age, gender, etc..

Response 1.4: The reviewer requests additional details for the trends in Fig. 9. In order to address this concern, captions are rewritten to provide considerably more information and now read:
“This image shows the effect of network delay and measurement delay of the whiteboard on the error of the reproduced image. A negative measurement delay means prediction is being used. If there is a 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.”

The reviewer requests additional information on the participants in the study. We agree more information should be provided. The added text to Section 6.4 reads:
“The user study was performed with 20 participants. The participants were between 20 and 60 years old, with an average age of 31. Approximately half of the participants had never interacted with the haptic device before, and the other half had at least 30 minutes of experience. No participant suffered from known neurological disorders.”

Comment 1.5: I suggest an improvement of the video. The video does not contain the dynamic movement of the whiteboard. The videos are suggested to include more details, such as experimental setups.

Response 1.5: We have added an additional video to further illustrate how our system performs. https://drive.google.com/drive/folders/1kzseNW7iwhsAkVAYOEEOUp8AzoUtMxg9?usp=sharing

Comment 1.6: I guess the moving speed of the whiteboard has a certain impact on the drawing quality and user performance. Have you investigated it? What’s the moving speed of the whiteboard when doing the experiments?

Response 1.6: The moving speed of the whiteboard during the experiment is 0.8 m/s on average. The fourth paragraph of Section 6.1 now reads: *“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, it becomes very challenging for any operator to produce an accurate drawing on such a moving surface at higher speeds.”*

Comment 1.7: As minor comments:
The video provided in Fig. 1 is not consistent with the text in Fig. 1’s caption. The video does not contain another individual altering the canvas’s position as the

operator makes the drawing.

Response 1.7: We have added an additional video to more clearly show the system in action. The videos can be found here: https://drive.google.com/drive/folders/1kzseNW7iwhsAkVAYOEEOUp8AzoUtMxg9?usp=sharing

Comment 1.8: Some paragraph divisions in the text are confusing. For example, on page 1, right column, lines 49-50. I suggest explaining both the operator's end and the remote side in the same paragraph.

Response 1.8: The paragraph mentioned has been rewritten for increased clarity. Furthermore, a thorough check to improve the writing quality has been performed; we hope the paper reads well now.

The part mentioned as an example now reads:
"Model Mediated Teleoperation differs from TI by assuming significant network delays from the beginning rather than focusing on reducing latency. Instead, MMT aims to mitigate the impact of these delays on the (control) system's transparency and stability Here transparancy refers to the ability of a system to be invisible to the operator, allowing them to faithfully experience the remote side. 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 communicate actions to the remote robot, receiving instantaneous feedback based on the local model of the remote environment."

"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."

Comment 1.9: Please standardize the font throughout the main text. For example, the phrase "remote domain" is in italics on page 4, right column, line 44, but in regular font on page 4, right column, line 24.

Response 1.9: This comment made it clear that the text emphasis with italics did not produce the intended effect. To address the comment, all uses of emphasis in the way described here have been removed from the manuscript.

Comment 1.10: Some notations of the parameters are confusing. The following lists some examples.
On page 6, left column, lines 37-39. The transition factor a is chosen to smoothly transition between X^ and X_mod. In my understanding, the equation should be given to show the relationship between the input X_mod(t) and the output X^(t). I can't fully understand why equation (1) gives the relationship between X_mod(t) and X(t).
On page 6, left column, lines 47-48. S = p_0 and S^ = {p^_0}, what's the meaning of

{}? In my understanding, the notations of S and S^ are only different in their domains, so why are they expressed in different ways? Additionally, {} usually means a set of data/points. Hence, for the expressions S^ = {p^_0} (on page 6, left column, lines 47-48) and X^ = {p^_p, f^} (on page 6, left column, lines 45-46), is the meaning of {} the same?
On page 6, right column, equation (3). Please give the definition of the parameter x. After seeing equation (5), I realized the parameter x actually refers to the expressions in equation (5). So I suggest an introduction to equation (3) after equation (5). Similarly, the definition of norm | * | (on page 6, right column, lines 14-15) can be introduced near page 6, right column, lines 30-31. Otherwise, it can make the reader confused about the parameter l.

Response 1.10: The reviewer made several comments that helped us improve the quality of the mathematical descriptions manuscript. In particular, Section 5.4 has been modified to address the reviewer's concerns.

Comment 1.11: I suggest an improvement of Fig. 7, which can include more details (can be shown in subfigure, notation, or partially enlarged view) as described in Section 6, on page 7, left column, lines 32-36. Otherwise, I don't know what exactly the pieces of square sponge and felt pads are.

Response 1.11: This is an excellent suggestion. Additional information is provided in the caption to address this. The addition reads:
"There is suspension fitted below the whiteboard in the form of a sponge and felt pads. This also makes the whiteboard glide consistently over the surface."

Comment 1.12: Section 6, on page 8, left column, lines 22-24. The author said that two control strategies are investigated. The second control strategy involves equation (1) and equation (6). I feel confused about how equation (1) and equation (6) are used together. Maybe the method lacks a whole expression on how to use equation (1) and equation (6) to create a target position. Or in my understanding, X(t) = {p_p, f}(I guess, but this equation is not provided in the manuscript). Still, I cannot fully understand the notation {*, *}. Does that mean the control signal includes position part p_p and force part f, and each part satisfies equation (1)?

Response 1.12: The reviewer has made several comments that helped us improve the quality of writing related to the mathematical descriptions in the manuscript. All of the points mentioned have been improved on, and an additional layer of polish has been applied to these sections of the manuscript.

Comment 1.13: Section 6, on page 8, left column, lines 32-37. The author uses the point cloud from the real sense camera to track the position of the whiteboard. I suggest adding a reference to the method the author used to obtain and process the point cloud.

Response 1.13: The method used to track the whiteboard was custom-made for the project. The technique is very good at providing low latency and high accuracy tracking for the purpose of this specific application. It does not generalize however, so we do not recommend using this particular method everywhere. When going from the provided solution to a more general-purpose implementation, we recommend more advanced computer vision techniques to track objects. We have added more details to the tracking method design in the manuscript.

In particular, the 2nd paragraph of Section 6.1 now reads: The whiteboard's movement is limited to a 1 DOF motion along rails on a surface. Due to this, the possible locations of the affixed 3d part are minimal. Only a narrow section of the point cloud, where only the 3D printed part's points exist, is used to refine the tracking. 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. Still, a low latency measurement is highly beneficial when using the measurement to have the robot compensate for the whiteboard's movement.

Reviewer: 2

Comments:
Dear Authors,

Comment 2.1: This paper presents a study on an intercontinental bilateral teleoperation system between the Netherlands and India. The researchers used a haptic device on the operator side and a local physics engine to collect the operator’s intent and feedback. The system architecture is presented, and the corresponding experiments are performed. However, the technical and scientific novelty and contribution of the paper are not clear enough. Further clarification is needed regarding the technical and scientific novelty of the proposed teleoperation system, especially regarding the “Operator Intent”. As a result, I recommend that the authors revise the paper before submitting it for publication in TMC. I have provided my comments below to help guide the revision process.

Response 2.1: This comment is similar to Reviewer 1’s Comment 1.3 on the clarity of Sections 3 and 5. Both sections have been rewritten to describe technical and scientific novelty better.

Comment 2.2: (Page 1, Motivation) The research motivation for intercontinental bilateral teleoperation needs to be strengthened in the introduction section, especially in the current structure of the paper. When introducing the research content of this paper, there is a lack of summary of previous work and an explanation of the necessity of ultra-remote teleoperation, which appears too abrupt.

Response 2.2: This is a crucial suggestion about stressing intercontinental bilateral teleoperation, which is quite specific. In reality, the more important property is teleoperation under high network latency, which considerably improves the feasibility of teleoperation with existing systems. We proved our system can act under high latency by deploying the system over an intercontinental distance.

We *changed the paper title and introduction* to address the reviewer's concerns and frame this from a more accurate perspective. The title of the paper has changed to: *“Utilizing Operator Intent for Haptic Teleoperation under high latencies.”*

Comment 2.3: (Page 1, Title & Fig. 1) As stated by the authors, this work is about the “bilateral” teleoperation. However, the force feedback from the robot side seems missed. And if so, the fig.1 should be re-corrected to avoid misleading to the readers, and also the title of this paper.

Response 2.3: In this work, the application, indeed, does not feature the robot directly communicating force measurements. The performance is considerably better for the operator because of the lack of measurement noise on the force feedback calculations the robot comes equipped with, and it is not the focus of this work.

To address the reviewer's concern, we have changed “*bilateral teleoperation*” to “*haptic teleoperation*” throughout the manuscript. We have also improved on Fig. 1 to reflect the system we built, along with consistent explanations and wording throughout the manuscript.

Comment 2.4: (Page 1, Video) The video provided by the authors seems missed the dynamic interference from the remote human, which is the most impressive highlight for this work. Pls add the corresponding demo. And if possible, the video can be provided as the supplementary material.

Response 2.4: Unfortunately, we mistakenly uploaded the incorrect video footage. This has been corrected in the new submission.

Comment 2.5: (Page 5, Section 5) The authors state that the network intercontinental link has an average latency of 179 ms. More configuration details for the network link should be provided, such as the device-to-device connection and the network service provider.
(Page 7, Equation (6)) How to confirm the value for the params (1/Ks)*f in the real systems?

Response 2.5 In order to explain k_s better, the following text is added to the third paragraph of Section 5.5: “The value of k_s is a parameter that determines the corresponding amount of force that should be applied proportional to the penetration depth of the operator's position into the surface of an object. The system designer can set the parameter, where a higher k_s yields a stiffer interaction and an increased susceptibility to noise.”

Comment 2.6: (Page 8, Section 6.3) There is no results shown in the paper for the increased latency using NetEM.

Response 2.6: This is correct. In the revised version, we used a real network instead of a simulated one. The mention of NeTem has been removed from the manuscript.

Comment 2.7: (Page 8, Section 6.4) The authors state that a motor drives the whiteboard, but the hardware setup details are missing. More details should be provided in figures or videos.

Response 2.7: A more elaborate description of the hardware setup has been provided. In particular, the following was added to the second paragraph of Section 6.3:
"The whiteboard is fitted onto a lead screw, which allows for a predictable sideways motion that can imitate the behaviour of the remote human. In this case, the system is configured to make repetitive oscillating motions."

Reviewer: 3
Comments:
Comment 3.1: From the experiment performed in the paper, the inclusion of a physics engine provides no real benefits as the local proxy cannot move any dynamic objects in the local digital twin. To demonstrate the benefit of the proposed solution, it is suggested that the authors extend this framework to a grasping task or other 3D object manipulation (which is challenging for MMT systems). Again, the idea of using operator intent is interesting, however, the experiment set up in the given paper does not sufficiently evaluate the idea.

Response 3.1: The reviewer is under the impression that the physics engine does not provide a real benefit due to the lack of active manipulation of 3D objects in the remote environment. While it is true that manipulating objects in the way described would be an exciting follow-up to the work shown here, we disagree with the reviewer that this is a requirement to benefit significantly from using a physics engine.

The primary responsibility of the physics engine is to predict an instant force feedback response to the operator's actions. Without such a physics engine, these calculations would have to be performed using a mathematical model, which would be impractical as complexity grows. In this case, we have demonstrated the ability of the physics engine to provide predictive force feedback with enough quality to enable operators to draw on a remote canvas.

Comment 3.2: For the four scenarios described in section 6.5, it seems that the authors changed multiple variables between some of the scenarios (especially from scenario 2, to scenario 3). This makes it harder to do direct comparisons.

Response 3.2: The reviewer points out that the exact differences between the scenarios are not apparent. We make it clear as follows:

Scenarios 1 and 2 are used to compare between live vision and rendered footage.

Scenarios 3 and 4 are used to compare the presence of the proposed method to combat mismatch.

The difference between Scenario 1 and Scenarios 3 and 4 is the manual addition of 1 second latency. Scenarios 2 and 3 differ because 1 second of artificially added latency in the control loop and rendered footage are used instead of live vision.

Comment 3.3: For the results of the user study, not sure how picture matching is done in subjective evaluation. Furthermore, it would be interesting to see if the imitation controller had any impact on the task execution time, which is an essential metric for teleoperation services.

Response 3.3: The picture matching is a subjective question asked to the participant. They rate the ability to picture-match themselves.

The reviewer's suggestion to look at task execution time is undoubtedly intriguing. We have looked into execution times when doing the project. Still, we determined that the participant's experience with the tasks was insufficient to use parameters that describe the task performance of the operator. The range of task performance values can vary highly between participants and even between repeated executions of the same task.

We are interested in finding ways to suppress this variance to examine this direction in the future, but it is out of the scope of this work.

Comment 3.4: In the introduction, the authors present Tactile Internet and MMT as two alternative approaches. We have a different view. MMT is one of the possible approaches to realise Tactile Internet when the distance between the operator and the remote environment is considerable. The paper *"Toward QoE-driven Dynamic Control Scheme Switching for Time-Delayed Teleoperation Systems: A Dedicated Case Study"* explains this well.

Response 3.4: The reviewer brings up a valuable point. It was not our intention to describe an exhaustive list. The wording in the manuscript has been adjusted to reflect this sentiment.

In particular, in the third paragraph of Section 1, we added: *“In this work, we consider two approaches for mitigating the effects of this latency.”*

Comment 3.5: The authors mentioned scalability in Section 3.4; however, I am concerned about the scalability of the proposed approach. It would be nice to discuss how to apply the proposed approach for different teleoperation tasks.

Response 3.5: We have changed the explanation for how to apply the approach to make it clearer how the method can be generalized. Most relevant changes are made to Section 5.

Comment 3.6: Last but not least, the presentation of this paper shall be significantly improved and a thorough proofreading is needed. The author might finish this manuscript in a hurry. For example,
1. Please properly use list, page 1 and page 4.
2. Many mathematics equations are missing index. The mathematics equations are only indexed from Section 5.1.
3. Many grammar mistakes.
4. First letter of a sentence is not capitalised in multiple places
5. Please either use "Figure x", or "Fig. x", making them consistent throughout the entire paper.

Response 3.6: All of the reviewer's comments have been investigated, and corrections have been made. Mathematics equations are given an index when referenced in the text.

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.P. Prabhakar, S. Coppens, R.R. Venkatesha Prasad, K. Van Berlo

Abstract—Bilateral teleoperation is a promising technology with applications such as telemaintenance and disaster management. However, it faces significant challenges when the application is subjected to simultaneously a high network latency and a dynamic environment with moving objects. This work aims to extend Model Mediated Teleoperation (MMT) to overcome its 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 $8000\,\mathrm{km}$ away (the Netherlands – India) and subjected to an average of $179\,\mathrm{ms}$ 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 second.

1 Introduction

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

In a haptic teleoperation setup, an operator engages with a device at the operator's end. The movements and manipulations performed on this device are then transmitted to the remote side. The remote robotic device imitates the operator's actions. The operator who then experiences feedback from the remote interactions. The feedback can be in the form of audio-visual and haptic feedback. The addition of haptic feedback, in particular, gives the operator the experience as if they were physically present in the remote environment and allows the operator to express actions in a more refined and natural manner.

In practice, realizing teleoperation over a network is challenging due to the delay imposed by the network. As the distance between operator and 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 $1\,\mathrm{ms}$ 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 ($\approx$ 150 km or $1\,\mathrm{ms}$ in terms of delay) supported due to fundamental physics-imposed limitations on network latencies. However, even without the maximum distance, conceptualizing any network with such low latency is profoundly challenging.

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 communicate actions to the remote robot, receiving instantaneous feedback based on the local model of the remote environment.

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.

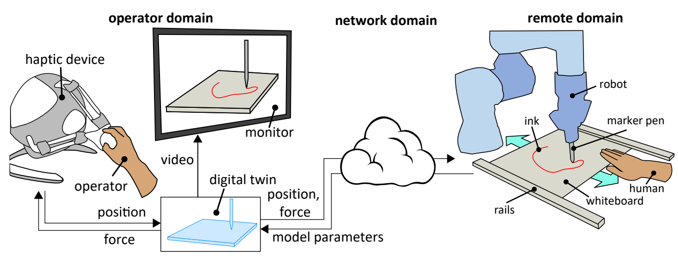


Fig. 1. We develop an approach allowing teleoperation over long distances. We demonstrate our effectiveness of approaches by creating the application shown in the figure. An operator uses a haptic device (in TUDelft the Netherlands) to guide a robot, equipped with a marker, to draw on a canvas situated on a different continent (IISc, Bangalore, India). Simultaneously, another individual alters the canvas's position as the operator makes the drawing. See https://drive.google.com/drive/folders/1kzseNW7iwhsAkVAYOEEOUp8AzoUtMxg9?usp=sharing for two videos.

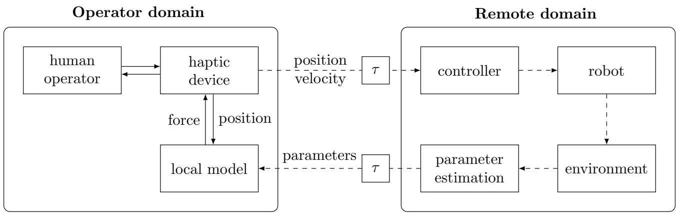


Fig. 2. General structure of a Model-Mediated Teleoperation system.

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 heavily depends on the local model being a faithful representation of the remote side. This is especially the case in dynamic environments since moving objects complicate updating the local model. 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 significantly improve the scalability of MMT solutions, we advocate utilizing physics engines that prioritize speed over accuracy and include methods to deal with the consequences of these 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 whiteboard over an $8000\,\mathrm{km}$ distance (TUdelft in the Ntherlands and IISc, Bangalore, India). See the video[1].

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

The rest of this work is structured as follows. Section 2 offers an overview of related literature. Section 3 outlines design principles aimed at enhancing MMT. Section 4 introduces a mathematical framework for MMT applications. Section 5 applies the framework to a specific application. Section 6 details the experimental setup of our user study. Section 7 presents an analysis of the user study results. Finally, Section 8 summarizes and concludes our findings.

2 Related works

2.1 Tactile Internet

Multiple empirical studies demonstrate that teleoperation applications where force feedback is transmitted over the network requires sub $10\,\mathrm{ms}$ 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 achieve a reduction in latency, which advances 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 small, but extremely latency constraint, making SDN and NFV ideal solutions to provision for this type of traffic.

2.2 Model Mediated Teleoperation

MMT solves the problem of performing teleoperation with significant network latency. MMT, however, poses 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, leading to an undesirable experience, and an undesirable control signal being 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 unintentionally cause 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.

2.3 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 field of MMT, it has been

1. https://drive.google.com/drive/folders/1kzseNW7iwhsAkVAYOEEOUp8AzoUtMxg9?usp=drive_link

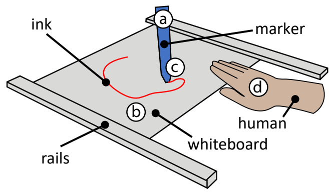


Fig. 3. Illustration of an application where a marker draws on a whiteboard while a human moves the whiteboard between two rails.

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.

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], [21], [22]. 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, there are imitation learning techniques that aim to make AI agents behave as a human would when presented with the same scenario as the AI is currently in [23].

3 Design Principles for human centered MMT

In this section, we present design principles to make a system for user centered teleoperation. It is important to first understand why an understanding of operator intent is needed in the first place. The reason is that the operator is operating under conditions that differ from the reality when the robot is mimicking the operators actions. This effect is exacerbated at increases in latency. When working with a local model, there can also be differences due to inaccurate measurements of the remote environment or limitations in the simulation accuracy.

It is important to minimize these differences, but it is unrealistic to require a design that does not have any mismatch between the operator and remote side. Therefore, we propose an approach where the mismatch is embraced, and solutions are placed to address the consequences of these mismatches. We propose to consider the operator intent and perception to provide the context to optimize the choices the robot makes. In Section 5, we validate these design principles with a concrete application and user study.

3.1 Mismatches

If the operator were perceiving the remote environment with high accuracy and without latency, there would be no need for the analysis presented here. Thus, the reason why the operator intent and perception need to be analyzed is because the perceived environment and real environment have mismatches. We first need to consider the nature of those mismatches we can expect.

Measurement noise: Feedback to the operator is based on measurements made in the remote domain. When using model-mediated feedback, the simulation requires a virtual replica based on observations from the remote environment. This replica will have inaccuracies due to both measurement noise and the inevitable simplifications used.

Simulation noise: The deployed simulation cannot replicate a real environment accurately enough to predict the future. Additionally, we advocate for using simpler, faster engines commonly found in video game design to optimize computational latency. As a result, objects will not respond exactly as simulated. This includes limitations in the complexity of objects; for example, if fluids are involved, accuracy can be expected to be very low.

External influences: The remote environment can feature influences not captured by the model. These could be events happening outside the scope of the constructed model or unpredictable active participants on the other side, such as humans.

Latency: Finally, there is latency. The key effect of latency is amplifying the negative effects of all other influences. An inaccurate simulation that predicts only a few milliseconds before being corrected will have a considerably smaller mismatch than a correction after 100 ms.

3.2 Capturing Operator intent

We introduce the notion of operator intent, which, within teleoperation, signifies how an operator would act if they were directly in the remote environment instead of the experience that was provided to the operator by the teleoperation system. It is not necessary to fully grasp operator intent to have an opportunity to improve the user experience. However, considerable improvements in user experience can be achieved by designing for operator intent.

In a simple setup, the only transmitted information would be the trajectory of the haptic device manipulated by the operator, without a clear way of extracting additional info. Because the operator is interacting with a local simulation, there is an opportunity to capture significantly more information. We separate out information about the operator's behavior, and the objects behavior separately.

Operator's behavior: By engaging with a local model, the operator can express their trajectory and the force they apply simultaneously. Thus, the operator's actions encompass information on both position and force. Not only that, but the operator's position and force can also be considered in the context of the local environment. The operator's trajectory can also be considered in relation to any object in the simulation environment. Similarly, the amount of applied force can be considered relative to the object's in the environment.

Object's behavior: The consequences of the operator's actions on the objects in the environment can be separately observed. Similar to the operator's behavior, these can be expressed as a global trajectory or as the trajectory relative to other objects in the environment. Furthermore, the applied forces to these objects can be captured too. Additional things can be considered, like whether an object falls, breaks, or is deformed.

Hard transitions In many situations, even a slight change in position or force can make a significant difference. For example, the difference of pressing a button or not, or the difference between an object lingering on the edge of a table or falling over. If these hard transitions are present and not directly tracked, a high level of accuracy is needed to make sure the operator experience and reality in the remote domain do not converge significantly.

Identifying hard transitions and separately ensuring that they are consistent can be a good way to avoid the need for high levels of accuracy for the entire application, while still maintaining a good user experience.

3.3 Operator Perception

Besides operator intent, another key concept to consider is operator perception. In the context of teleoperation, the operator perception is described as to what extent the operator can perceive a deviation of the imitation to the demonstration and how this affects the user experience.

The controller in the remote domain attempts to manipulate the robot to imitate the operator's intent. In order to do so, the controller requires some room for deviations. These deviations are used to improve the realization of operator intent. The more freedom the controller has to alter the trajectory of the robot, the easier it becomes to align the environments. However, it is important to ensure that the deviations are only marginally detectable by the human operator.

One must consider the operator's perception of these deviations. Recognizing differences between the operator's original actions and the imitated trajectory is not as simple as quantifying the deviation using metrics like the root mean square error (RMSE) between the original and imitated trajectories. This has been demonstrated in multiple prior works [9], [10], [24]. Humans perceive deviations in a more nuanced manner. For instance, while differences like low-frequency deviations in absolute position and variations in scale might be challenging to detect, the difference between hovering over and hitting an object or small high-frequency vibrations become immediately apparent. Furthermore, how deviations are perceived and impact the user experience can be subjective and vary between operators.

3.4 Scalability

Traditional model-mediated teleoperation (MMT) methods often use handcrafted solutions, mainly due to their aim to achieve precise alignment between the virtual and remote environments. However, as we have identified the potential to accommodate a certain degree of mismatch, this opens doors for alternative approaches to representing the remote environment.

We advocate for the use of real-time physics engines. These engines, already prevalent in the robotics and computer graphics fields, can model a diverse array of scenarios in real time. For operator interactions in an MMT system, the level of accuracy provided by a physics engine is sufficient. This is primarily because the operator's perception cannot detect minor inaccuracies, provided the simulated interaction is plausible.

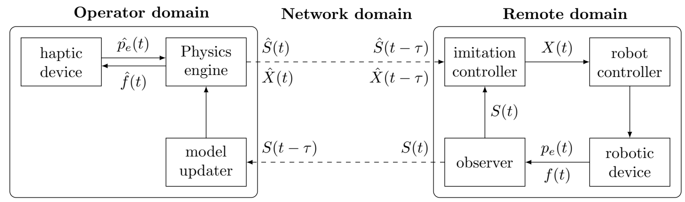


Fig. 4. Illustration of our proposed framework that aims to extend MMT to work with complex and dynamic environments by considering operator intent.

Furthermore, leveraging physics engines presents a set of pragmatic advantages. They are often supported by extensive communities, many are open-source, and they can be adapted to specific requirements.

In Section 5, we describe how we deploy a physics engine to construct the application used in our user study.

4 A FRAMEWORK FOR MMT WITH OPERATOR INTENT

In this section, we outline a framework for designing a system for teleoperation over long distances. The system is built on the MMT system design and emphasizes the design principles given in the previous section. An overview of the framework is illustrated in Fig. 4. The system features three main parts (domains): The Operator Domain, Remote Domain, and Network Domain. Throughout this work, for any parameter θ in the remote domain, we use $\hat{\theta}$ to denote its counterpart in the operator's domain.

We denote S as the observed state of the environment in the remote domain. The state of the environment includes attributes like object location, orientation and shape, mass, friction coefficients, center of mass, and inertia. These properties are either accurately provided as priors or are provided with an initial guess which can be improved with subsequent observations in the remote domain at runtime.

In the operator domain, a physics engine enables operators to engage with a digital twin of the distant environment. The digital twin of the environment in the remote domain used by the physics engine in the operator domain is denoted as $\hat{S}$.

The operator manipulates a haptic device to interact with the local physics engine. The haptic device measures only the position of its end-effector, which is the endpoint on the robotic arm, and is denoted as $\hat{\boldsymbol{p}}_e$. A haptic rendering algorithm within the physics engine converts the position of the end-effector to a position in the virtual environment and an applied force. The description of the operator's state in the virtual environment is the control signal $\hat{X}$. The predicted applied force is indicated as $\hat{\boldsymbol{f}}$, which is fed back to the operator without network delay. With this, we can consider the physics engine as a function that modifies the state of the environment and the operator represented by

$$\left(\hat{S}(t), \hat{X}(t)\right) = \text{physicsEngine}\left(\hat{S}(t-e), \hat{X}(t-e), \hat{\boldsymbol{p}}_e(t)\right),$$

where $t-e$ indicates the previous discrete step of the physics engine.

Both a full description of the virtual state $\hat{S}$ and the operator $\hat{X}$ are transmitted to the remote domain. The data arrives with an added network latency of τ. The imitation controller considers the delayed state in the operator domain and the state of the local environment to modify the control signal. Then we get

$$X(t) = \text{imitationController}\left(\hat{X}(t-\tau), \hat{S}(t-\tau), S(t)\right).$$

Note that the imitation controller should be designed so that if there is no mismatch between the two states, the imitation controller should not modify the control signal provided by the operator. This means that when $S(t) = \hat{S}(t-\tau)$, one gets $X(t) = \hat{X}(t-\tau)$. However, When the two states have mismatches, the imitation controller is responsible for modifying the control signal to prioritize the realization of operator intent. The strategy for discerning operator intent and operator perception can be determined beforehand and used to design the imitation controller.

The control signal is used to drive the robot controller, which covers any intricacies related to the used robotic device. The robot controller is completely agnostic to the considerations of operator intent and only considers the output of the imitation controller. The robotic device measures the position of the end-effector $\boldsymbol{p}_e$ and optionally the force applied to it as $\boldsymbol{f}$. The observer is a collection of sensors in the remote domain that track the realtime position, orientation, and motion of every object in the environment. Combined with the measurements from the robotic device, the observer constructs an estimation of the state of the remote environment S. It is important that object tracking is done with high accuracy and low latency, especially when the tracking information is directly being used in the control strategy in the imitation controller.

The observed state of the remote environment is sent back to the operator domain, where it arrives with τ network delay. Audio, video, and force measurements that result from active remote interactions can be included in the feedback and should be immediately relayed back to the operator. The measured state of the remote environment S is used to update the digital twin in the operator domain $\hat{S}$. We get

$$\hat{S}(t) = \text{modelUpdater}\left(S(t-\tau), \hat{S}(t), \hat{X}(t)\right).$$

The update strategy should be designed so that it minimally disturbs the operator. The updating strategy can involve postponing model updates when the operator is actively interacting with an object, which has been shown to have potential [15], [16].

The network domain encompasses all system elements that synchronize data between the operator and remote domains. Teleoperation applications feature multiple modalities with highly varying requirements. Kinematic data, which captures object positions and orientations, has stringent latency requirements but is compact, taking up only a few bytes per object in the environment. Typically, this type of data is highly resistant to data loss because subsequent packets remove the need to retransmit prior ones. Kroep et al. demonstrated a teleoperation setup with satisfactory user experience using a network with 50% packet loss [9]. This is further amplified by the fact that the controller of the robot arm serves as a low pass filter both due to the acceleration limitations of the robot and the lower update rate (60 Hz against a 1 kHz packet rate). Conversely, data detailing objects' shapes, physical attributes, and audio-visual content have a larger payload but are significantly more tolerant of latency while requiring high reliability. Therefore, the network must proficiently manage diverse data types, ensuring high-volume transmission while adhering to the varying latency demands of each attribute. Especially in complicated environments, data generation may outpace available bandwidth, necessitating prioritization of vital data based on the operator's actions and vicinity to objects. For this reason, adept protocols, efficient bandwidth utilization, data compression, and priority management should be considered in the network design.

5 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. 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 3.

5.1 Mismatches

The first step is to identify the mismatches that can occur in this application. This will be done as described in Section 3.1

measurement noise: In this case, 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 latency of 179 ms. In particular the external influences will be communicated with this latency, causing challenges to make the application work. In a simple implementation the whiteboard position relative to the marker will be consistantly different between operator and remote domain, causing any drawing to be considerably deformed.

5.2 Capturing operator intent

The next step is to identify operator intent as described in Section 3.2.

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

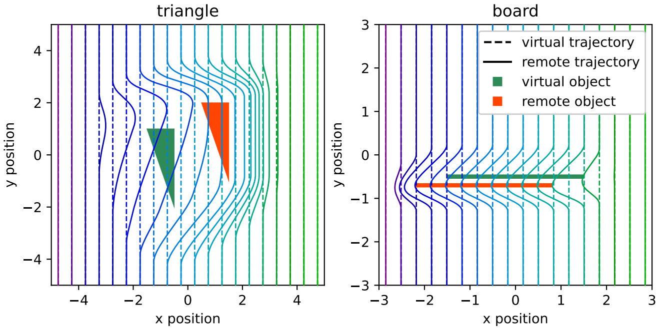


Fig. 5. Illustration of the behavior of the imitation controller design. For a given set of virtual trajectories of $\hat{\boldsymbol{p}}_p$ in the operator domain, the modified trajectory of $\boldsymbol{p}_p$ in the remote domain is shown.

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.

Hard transitions: In this case, the most important hard 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 $1\,\mathrm{mm}$ can cause a significant difference.

5.3 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.

5.4 Translation between absolute and relative trajectory

Based on the insights obtained previously, we identify the key mismatch to consider 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 seamless, ensuring that any movement in the operator domain corresponds to a monotonically increasing movement in the remote domain. In other words, any motion in the operator domain should not result in a contrary movement in the remote domain.

In this section, we describe how to design an imitation controller that adjusts the measured behavior of the operator to respect the operator intent in our specific application. $\hat{X}$ describes the position of the end effector that directly corresponds with the haptic device in the operator domain. We take X_{relative} as the position in the remote domain that has the same position related to the whiteboard as in the operator domain. Next, a transition factor α is chosen to smoothly transition between $\hat{X}$ and X_{relative}. We get

$$X(t) = \alpha\hat{X}(t-\tau) + (1-\alpha)X_{\text{relative}}(t). \quad (1)$$

Note that $\hat{X}(t-\tau)$ includes the communication delay τ.

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,

$$X_{\text{mod}}(t) = \{\hat{\boldsymbol{p}}_p(t-\tau) + \boldsymbol{p}_o(t) - \hat{\boldsymbol{p}}_o(t-\tau), \hat{\boldsymbol{f}}(t-\tau)\}, \quad (2)$$

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 of 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

$$\hat{\boldsymbol{p}}_{\text{min}} = \arg\min_{\boldsymbol{p}\in P}(\boldsymbol{p} + \hat{\boldsymbol{p}}_o - \hat{\boldsymbol{p}}_p). \quad (3)$$

To smoothly transition between $\hat{X}$ and X_{mod}, we design a smooth transition function that is used over the transition region to calculate the transition factor α. As a transition function, we introduce a cubic spline as

$$g(x) = \begin{cases} 0 & \text{if } x \leq 0, \\ 1 & \text{if } x \geq 1, \\ 3x^2 - 2x^3 & \text{otherwise,} \end{cases} \quad (4)$$

where x is any input to the function g, in this case correlated to the distance between the operator position and the object.

The transition region should be chosen as such that the monotonicity condition is preserved. This means that any movement in the operator domain will not lead to any movement in the opposite direction in the remote domain. This condition holds when $\frac{d}{dt}\hat{\boldsymbol{p}}_p(t-\tau) \cdot \frac{d}{dt}\boldsymbol{p}_p(t) > 0$.

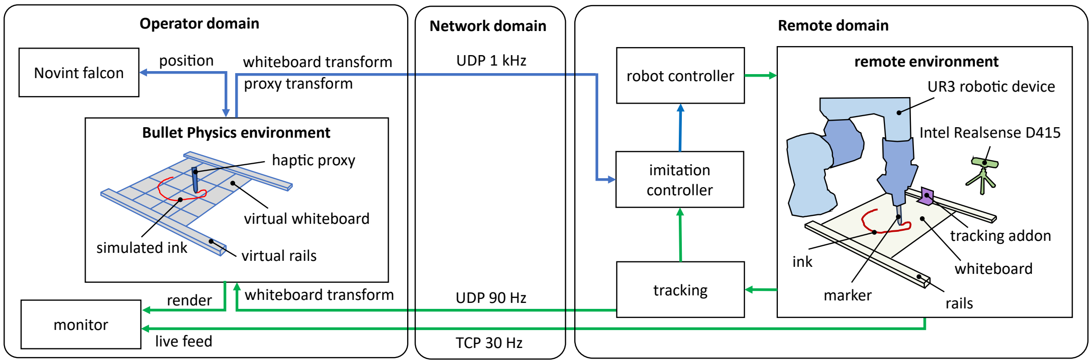


Fig. 6. Schematic overview of experimental setup. In the figure, we lay down the different components of our setup, showcasing how they relate to the proposed teleoperation framework and how data flows through the system. Blue arrows indicate high-frequency communication of $1\,\mathrm{kHz}$, while green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

The critical direction that determines whether the monotonicity condition is met is when $\frac{d}{dt}\hat{\boldsymbol{p}}_p$ and $\hat{\boldsymbol{p}}_{\text{min}}$ are in the same direction. Because the maximum slope of the transition function in Eq. (4) is 1.5, the transition needs to be at least $1.5|\boldsymbol{p}_o - \hat{\boldsymbol{p}}_o|$ when $|\frac{d}{dt}\hat{\boldsymbol{p}}_p \times \hat{\boldsymbol{p}}_{\text{min}}| = 0$ and $\frac{d}{dt}\hat{\boldsymbol{p}}_p \cdot \hat{\boldsymbol{p}}_{\text{min}} > 0$ to guarantee monotonicity. Here, we denote $|\cdot|$ as the l^2 norm of a vector.

In this work we use a transition length of $|\boldsymbol{p}_o - \hat{\boldsymbol{p}}_o|$ in all but the critical direction and $2|\boldsymbol{p}_o - \hat{\boldsymbol{p}}_o|$ in the critical direction. Thus, we can get the transition factor as

$$\alpha = \begin{cases} g\left(\frac{|\hat{\boldsymbol{p}}_{\text{min}} - \frac{1}{2}\text{proj}_{(\boldsymbol{p}_o - \hat{\boldsymbol{p}}_o)}\hat{\boldsymbol{p}}_{\text{min}}|}{|\boldsymbol{p}_o - \hat{\boldsymbol{p}}_o|}\right) & \text{if } \hat{\boldsymbol{p}}_{\text{min}} \cdot (\boldsymbol{p}_o - \hat{\boldsymbol{p}}_o) > 0, \\ g\left(\frac{|\hat{\boldsymbol{p}}_{\text{min}}|}{|\boldsymbol{p}_o - \hat{\boldsymbol{p}}_o|}\right) & \text{otherwise,} \end{cases} \tag{5}$$

where $\text{proj}_{\boldsymbol{b}}\boldsymbol{a}$ is the projection of $\boldsymbol{a}$ onto $\boldsymbol{b}$. Finally, we can use Eq. (2), (4), and (5) in (1) to obtain the control signal for 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 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.

5.5 Controller design

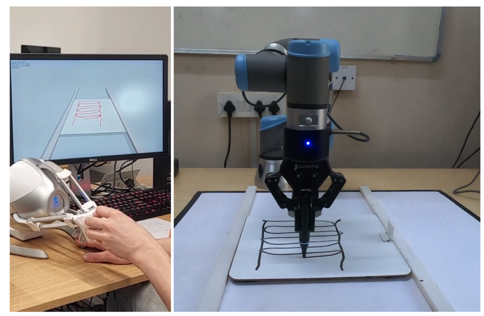
It is common for a robotic device in these types of applications to be controlled with a compliance controller that makes use of an accurate force-torque sensor [25]. In this application, there are no remotely initiated interactions that need to be relayed to the operator, as the alteration of the whiteboard 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 [25]. The applied force $\boldsymbol{f}$ is converted into a positional offset. We can calculate the target position fed to the position controller as

$$\boldsymbol{p}_{\text{target}} = \boldsymbol{p}_p + \frac{1}{k_s}\boldsymbol{f}. \tag{6}$$

In this case the position of the end-effector p_p is the modified version obtained using (1) and (5). Note that for safety, $\boldsymbol{p}_{\text{target}}$ is restricted within an operating range to avoid undesirable control signals when someone accidentally hits the Novint Falcon. Also note the spring constant k_s. The value of 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 Novint Falcon end-effector's proxy in the operator domain contacts the whiteboard without exerting force, its corresponding position in the remote domain remains $1\,\mathrm{mm}$ 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.



(a) (b)

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.

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 Eq. (6). In the second control strategy, the mismatch in the whiteboard position between the operator and remote domains is considered. Here, Eq. (1) and Eq. (6) 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 $X(t)$.

6 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 in a Western European institution and the remote domain in an Asian institution[2]. An overview of the application is provided in Fig. 6.

In the operator domain, the Bullet-Physics engine provides the local simulation [26]. We have adapted the physics engine to support a haptic rendering algorithm and interface with the 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 $1\,\mathrm{kHz}$. The physics engine is decoupled from the rendering engine so that the physics engine can run at $1\,\mathrm{kHz}$ while the OpenGL-based renderer runs at $60\,\mathrm{Hz}$. This ensures a smooth and responsive haptic response from the physics engine with sub $1\,\mathrm{ms}$ computational delay. A photo of the operator domain is shown in Fig. 7(a).

In the remote domain, we deploy a UR3 robot. Mounted on the end-effector of the UR3 is a gripper that holds a marker. Two rails secured to the table lock a whiteboard in a 1 DoF motion towards the base of the UR3. Pieces of square sponge and felt pads are attached on the bottom of the corners of the whiteboard to serve as a suspension of $4\,\mathrm{mm}$ until fully compressed. A ROS2 environment communicates directly with the UR3 [27]. Fixed to the movable whiteboard is a small 3D-printed part. An Intel Realsense D415 camera captures RGB-D images from the side and is used to track the 3D-printed part attached to the whiteboard. Tracking happens at $90\,\mathrm{Hz}$. 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.

6.1 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 easily 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.

2. More specifics on the institutions will be revealed on publication.

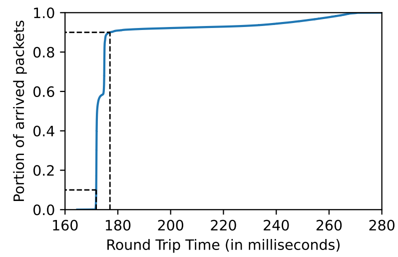


Fig. 8. Cumulative distribution of end-to-end network latency measured over 15 hours. $80\,\%$ of the packets have latencies between $172\,\mathrm{ms}$ and $177\,\mathrm{ms}$. The average latency is $179\,\mathrm{ms}$.

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, $90\,\mathrm{Hz}$ 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 model updater's only task is 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 $0.8\,\mathrm{m/s}$. 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.

6.2 Network

The operator and remote domains are separated by an approximate distance of $8000\,\mathrm{km}$. Kinematic and force data from the operator, as well as kinematic data from the whiteboard, are relayed over a UDP channel at $1\,\mathrm{kHz}$ and $90\,\mathrm{Hz}$, respectively. Both feedforward and feedback packets carry a 100-byte payload. 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

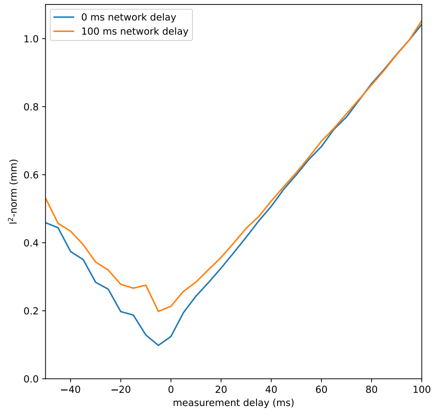


Fig. 9. This image shows the effect of network delay and measurement delay of the whiteboard on the error of the reproduced image. Negative measurement delay means prediction is used. If there is measurement delay, the robot cannot compensate for the movement in the whiteboard, resulting in an incorrect 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.

computer in the operator domain at a maximum rate of 30 Hz. The Round-Trip Time (RTT) for the UDP link was assessed over 10 hours, and the results are depicted in Fig. 8. The RTT reveals that 80 % of packets arrive between 172 ms and 177 ms. Last, to push the network further in our experiments, we used NetEM to increase the latency by 1 s.

6.3 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 hardcoded 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 effectively separates the effects of time mismatches and position mismatches [3]. It quantifies image quality using an ℓ^2-norm.

The results are shown in Fig. 9. 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 in a 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. While this 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.

6.4 User Study

In the user study, participants use the experimental setup to draw pictures in a remote environment under varying conditions. First, 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.

③ *Render with moving whiteboard, without imitation 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 imitation controller, increased latency*: The participant only observes the local render in the operator domain. The whiteboard is in constant motion. The imitation controller considers the movement of the whiteboard. For scenario ③ and ④ an artificial latency of 1 S 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

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

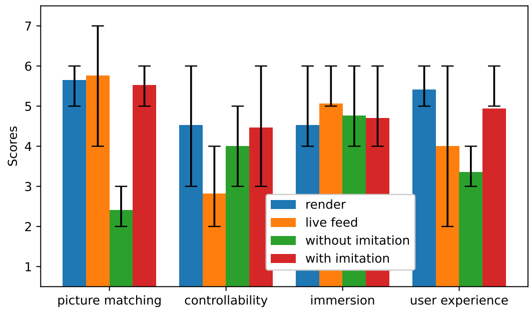


Fig. 10. The results of the user study. Four scenarios are rated on four categories using a 7-point Likert scale.

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.

7 Performance analysis

Fig. 10 showcases the overall results of the user study. The first two scenarios indicated as render and live feed, are identical except for the type of video feedback provided. In the first case, the local render is used, while in the second case, the live feed of the remote environment is used. Note that in both cases, the virtual environment provides instantaneous force feedback. In the scenario with the live feed, the participant experiences the force feedback of pressing on the whiteboard before he can visually see the marker in the remote domain touching the whiteboard.

There are two areas in which the live feed outperforms the local render. The first one is in picture matching. This outcome is anticipated since the direct footage of the remote environment is utilized. The only thing limiting the picture matching is the quality of the video feed. Conversely, the local render is only an approximation of the drawing in the remote domain. Furthermore, the mechanics of the robot controller filter out some of the lower frequencies in the operator trajectory.

Compared to observing a locally rendered approximation, a direct video feed from the remote environment was perceived to be more immersive. While realistic matching between the local renderings was not a focus of this work and can be significantly improved in the future, we expect the immersion of the real footage to not be exceeded by a local render in the near future.

Inference 1. The live feed offers a stronger immersion and perception of the true state of the remote environment.

Feedback from participants revealed a significant increase in perceived control when operating with the local render for visual feedback. Participants also remarked on the need to operate at a significantly slower pace when using the live feed to execute tasks. This behavior of using slow, methodical motions can help counteract the latency in the video feed. The difference in controllability translated to a strong preference for the local render experience, which was rated as markedly more desirable than its live feed counterpart.

Inference 2. The local render offers significantly better controllability resulting in a superior user experience.

For the next set of experiments, the whiteboard was continuously in motion, fully exposing the challenges of teleoperating in a dynamic environment with MMT. We also added 1 second of additional end-to-end network latency. Two control methods were assessed: (i) disregarding the whiteboard's relative position and (ii) taking the relative position into account. No discernible difference in controllability or immersion during task execution was observed between the two methods. However, after the task, there was a clear difference when observing the final drawing in the remote domain. Participants noted that when using the absolute control strategy, the final drawing significantly deviated from the drawing made in the operator domain. Participants indicated that it was more desirable to have a smooth experience during the task and have a mismatched result than to face the challenges of drawing with delayed visual feedback and getting an accurate outcome. Consequently, the absolute control strategy with the moving whiteboard was rated the worst overall experience.

Inference 3: As expected, a rudimentary implementation of MMT fails to handle dynamic environments, leading to large mismatch between what the operator was trying to achieve and what occurred in the remote side.

In stark contrast, when the imitation controller was deployed, participants observed hardly any difference between their drawing in the operator domain and the resulting drawing in the remote domain. Consequently, this scenario with a moving board was rated with a high overall experience, above the experience with live feed where the board was stationary. This demonstrates the efficacy of our imitation controller.

Inference 4: The performance of MMT approaches in dynamic conditions can be significantly improved by enhancing it with the capability to capture and preserve operator intent.

8 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 simultaneously a high network latency and a dynamic environment with moving objects. This work set out to extend Model Mediated Teleoperation (MMT) to overcome its 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 that focus 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 through objective measurements the system's ability to replicate actions produced by the operator. We built this application on a system where the operator and remote domain are $8000\,\mathrm{km}$ apart with an average end-to-end network latency of $165\,\mathrm{ms}$. 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 $1\,\mathrm{s}$.

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," *Journal of Intelligent & Robotic Systems*, vol. 104, pp. 1–23, 2022.

[2] M. Simsek, A. Aijaz, M. Dohler, J. Sachs, and G. P. Fettweis, "5G-Enabled Tactile Internet," *IEEE Journal on Selected Areas in Communications (JSAC)*, vol. 34, no. 3, pp. 460–473, 3 2016.

[3] F. Dressler, F. Klingler, M. Segata, and R. Lo Cigno, "Cooperative Driving and the Tactile Internet," *Proceedings of the IEEE*, vol. 107, no. 2, pp. 436–446, 2 2019.

[4] G. P. Fettweis, "The tactile internet: Applications and challenges," *IEEE Vehicular Technology Magazine*, vol. 9, no. 1, pp. 64–70, 2014.

[5] A. A. Ateya, A. Muthanna, A. Vybornova, I. Gudkova, Y. Gaidamaka, A. Abuarqoub, A. D. Algarni, and A. Koucheryavy, "Model mediation to overcome light limitations—toward a secure tactile internet system," *Journal of Sensor and Actuator Networks*, vol. 8, no. 1, p. 6, 2019.

[6] P. Mitra and G. Niemeyer, "Model-mediated telemanipulation," *The International Journal of Robotics Research*, 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," *International Journal of Advanced Robotic Systems*, vol. 15, no. 2, p. 1729881418761136, 2018.

[9] H. Kroep, V. Gokhale, J. Verburg, and R. V. Prasad, "Etvo: Effectively measuring tactile internet with experimental validation," *IEEE Transactions on Mobile Computing*, 2023.

[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 *Proceedings of the ACM/IEEE 14th International Conference on Cyber-Physical Systems (with CPS-IoT Week 2023)*, 2023, pp. 199–208.

[11] N. Promwongsa, A. Ebrahimzadeh, D. Naboulsi, S. Kianpisheh, F. Belqasmi, R. Glitho, N. Crespi, and O. Alfandi, "A comprehensive survey of the tactile internet: State-of-the-art and research directions," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 1, pp. 472–523, 2020.

[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 *2017 13th International Conference on Advanced Technologies, Systems and Services in Telecommunications (TELSIKS)*. IEEE, 2017, pp. 113–118.

[14] B. Willaert, H. Van Brussel, and G. Niemeyer, "Stability of model-mediated teleoperation: Discussion and experiments," in *Haptics: Perception, Devices, Mobility, and Communication: International Conference, EuroHaptics 2012, Tampere, Finland, June 13-15, 2012. Proceedings, Part I.* Springer, 2012, pp. 625–636.

[15] P. Mitra, D. Gentry, and G. Niemeyer, "User perception and preference in model mediated telemanipulation," in *Second Joint EuroHaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (WHC'07)*. IEEE, 2007, pp. 268–273.

[16] X. Xu, C. Schuwerk, and E. Steinbach, "Passivity-based model updating for model-mediated teleoperation," in *2015 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*. IEEE, 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 *2015 IEEE International Symposium on Haptic, Audio and Visual Environments and Games (HAVE)*. IEEE, 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 Transactions on Robotics*, vol. 31, no. 4, pp. 850–863, 2015.

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

[20] Y. Li and S. S. Ge, "Human–robot collaboration based on motion intention estimation," *IEEE/ASME Transactions on Mechatronics*, vol. 19, no. 3, pp. 1007–1014, 2013.

[21] A. T. Le, M. Guo, N. van Duijkeren, L. Rozo, R. Krug, A. G. Kupcsik, and M. Bürger, "Learning forceful manipulation skills from multi-modal human demonstrations," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 7770–7777.

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

[23] A. Hussein, M. M. Gaber, E. Elyan, and C. Jayne, "Imitation learning: A survey of learning methods," *ACM Computing Surveys (CSUR)*, vol. 50, no. 2, pp. 1–35, 2017.

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

[25] S. Scherzinger, A. Roennau, and R. Dillmann, "Forward dynamics compliance control (fdcc): A new approach to cartesian compliance for robotic manipulators," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 4568–4575.

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

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

BIOGRAPHIES

Kees Kroep received an MSc degree in electrical engineering at Delft University of Technology in 2018. Currently he is working toward his Ph.D. degree in Embedded and Networked Systems group at EEMCS, TU Delft. His Ph.D.work focuses on edge computing for tactile internet. His research interests include tactile internet and haptics technology.

R. Venkatesha Prasad is an associate professor at 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 63 MSc students. He has over 300 publications in the peer-reviewed international journals and 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 Transaction on Cognitive Communication and Networking. He was the vice-chair of IEEE Tactile Internet standardization workgroup and now a mentor. For more information, please refer to http://homepage.tudelft.nl/w5p50