Breaking the Latency Barrier

Practical Haptic Bilateral Teleoperation over 5G

Kroep, Herman; Coppens, Stijn; Wösten, Koen; Bhattacharjee, Anup; Venkatesha Prasad, R. R.

DOI
10.1145/3716550.3722020

Publication date
2025

Document Version
Final published version

Published in
Proceedings of the ACM/IEEE 16th International Conference on Cyber-Physical Systems, ICCPS 2025, held as part of the CPS-IoT Week 2025

Citation (APA)
Kroep, H., Coppens, S., Wösten, K., Bhattacharjee, A., & Venkatesha Prasad, R. R. (2025). Breaking the Latency Barrier: Practical Haptic Bilateral Teleoperation over 5G. In *Proceedings of the ACM/IEEE 16th International Conference on Cyber-Physical Systems, ICCPS 2025, held as part of the CPS-IoT Week 2025* Article 2 ACM. https://doi.org/10.1145/3716550.3722020

Breaking the Latency Barrier: Practical Haptic Bilateral Teleoperation over 5G

Herman Kroep
Delft University of Technology
Delft, Netherlands
h.J.C.kroep@tudelft.nl

Stijn Coppens
Delft University of Technology
Delft, Netherlands
sdd.coppens@gmail.com

Koen Wösten
Delft University of Technology
Delft, Netherlands
koen.wosten@gmail.com

Anup Bhattacharjee
Delft University of Technology
Delft, Netherlands
a.k.bhattacharjee@tudelft.nl

R.R. Venkatesha Prasad
Delft University of Technology
Delft, Netherlands
R.R.VenkateshaPrasad@tudelft.nl

Abstract

Haptic bilateral teleoperation holds promise for applications such as telemaintenance, remote manipulation, and disaster response, yet delivering precise, low-latency force and video feedback remains challenging. This study advances haptic bilateral teleoperation by combining live video with Model Mediated Teleoperation (MMT) to enable predictive force feedback. While this method has benefits, several non-trivial challenges, such as synchronizing the model with user's and remote robot's actions, arise. A novel algorithm is developed that allows the robotic device to replicate interactions predictively experienced by the operator. We validated this approach in a fully functional system that performs reliably despite significant network delays. The latency performance of the system is extensively characterized, achieving a motion-to-pixel latency of 58 ms. A user study revealed that operators did not perceive network latency of at least 75 ms, resulting in a 133 ms motion-to-pixel delay requirement. Additionally, a 5G latency analysis demonstrated that effective haptic teleoperation is achievable with both operator and remote ends connected via 5G. This provides a path away from strict latency requirements toward practical teleoperation solutions using currently available technology.

1 Introduction

Haptic bilateral teleoperation allows humans to manipulate remote environments through robotic devices, extending their expertise over the internet to a remote place. The operator receives visual feedback through a monitor and force feedback through a haptic device. This provides the necessary sensation to execute physical actions in the remote environment as shown in the Fig. 1.

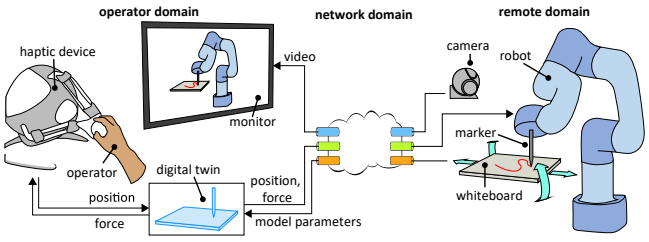

Figure 1: An operator uses a haptic device to guide a robot equipped with a marker to draw on a whiteboard. Simultaneously, a 3-RRS manipulator controls the orientation of the whiteboard as the operator makes the drawing. The operator receives live video footage on a monitor and predictive force feedback from a local physics simulation with a digital twin of the remote environment.

Such a system has the potential to enable people to work and contribute from anywhere to anywhere, significantly diminishing the dependence on their physical presence. This can offer cost savings, quicker emergency responses, and large environmental benefits. It also allows people to act in inaccessible or risky areas, from modifying satellites in orbit to rescuing people from burning buildings.

Three key components are required for the effective human operation of a remote robotic device. First, the robot must replicate the operator's movements, which requires transmission of the operator's actions. Second, visual feedback on the robot's real-time performance must be provided. Third, force feedback must be provided to convey the robot's interactions with the environment.

We consider two methods for providing feedback to the human operator. The first method is to directly measure, using a camera for visual feedback or sensors for force feedback and transmitting this data directly to the operator. The

ICCPS '25, May 6-9, 2025, Irvine, CA, USA

ACM ISBN 979-8-4007-1498-6/2025/05.
https://doi.org/10.1145/3716550.3722020

second method uses a local simulation to calculate predictive feedback. Continuously rendering images of the local simulation can be used for predictive visual feedback, while a physics simulation can calculate predictive force feedback.

The first method of measuring and directly transmitting feedback does not require constructing a model, making it scalable and well-suited for general-purpose teleoperation. However, this approach faces the significant obstacle of meeting extremely low latency requirements. Research shows that delay exceeding 5 ms in conveying force feedback impairs user experience [5]. Despite efforts to minimize network delay, achieving such a low threshold is still highly challenging. Nonetheless, this direct measurement method has been successfully implemented in controlled environments, such as hospitals, where a wired connection between the operator and robot can restrict the delay under the requirement threshold.

The second method of predicting feedback uses a local simulation to predict feedback, a technique known as Model Mediated Teleoperation [9, 10, 13]. Instead of communicating measured force feedback, the remote domain extracts model parameters from the remote environment. These model parameters are used in the operator domain to predict feedback based on the operator's actions before the robotic device imitates the actions. This approach provides a low-delay feedback source through simulation but requires a highly accurate local model that reflects the remote environment. Constructing such a model is challenging even in a known environment. It becomes particularly difficult in unfamiliar environments with hard-to-capture details, such as objects out of view or properties like density. Additionally, certain environments are far easier to simulate accurately than others; scenarios involving liquids or deformable objects, for instance, demand performance that is not available in the state of the art.

To maintain the accuracy of the local model, it has to be updated with the latest observations from the remote environment while the operator is actively interacting with it. The added network delay further complicates this task. While recent studies showcase this method's potential to handle a significant network delay, current implementations are limited to static and rigid objects [5, 6, 11].

Given the significant challenges existing approaches pose, we arrive at the following questions: *How to realize haptic bilateral teleoperation in today's networks?* It is obvious that MMT is needed to deal with higher network latencies, but this comes with the challenges above. Thus, we pose the following research question: *How to realize a high-quality haptic bilateral teleoperation session for a human operator with MMT (local models) while dealing with changes in the dynamic remote environment?* We propose a hybrid solution that leverages both approaches: visual feedback is delivered through live video transmitted from a camera and displayed on a monitor. In contrast, predictive force feedback is generated through a local physics simulation. This approach addresses the core limitations of each method: while force feedback requires ultra-low delay, video feedback is less delay-sensitive. Additionally, predicting force feedback is more feasible than delivering an immersive, accurate, real-time rendered visual.

This proposed solution, though promising, still faces significant challenges, which we will outline and solve in this work. It must undergo thorough validation to ensure the approach is feasible and provides a satisfactory user experience. Additionally, delay requirements must be established for the new approach. The operator will experience predictive, near-instantaneous force feedback alongside video feedback with greater delay, and this discrepancy must be explored. Another key challenge is that force feedback is provided for actions that have not been executed yet, which could create a mismatch between the operator's experience and the robot's eventual actions.

To address these, we develop an algorithm that causes the robot to execute deliberate actions to ensure its movements retrospectively align with the operator's prior experience. We construct a real-world system based on this concept, demonstrating its feasibility for long-distance operation and evaluating its performance through extensive user studies. Video footage of the experimental setup is provided at this URL[1] .

All the evaluation and user studies have been performed in a live private 5G network. The contributions of this work are as follows:

1 **Framework:** We present a framework for haptic bilateral teleoperation that integrates live video measurements with model-mediated teleoperation for predictive force feedback.
2 **Algorithm:** We propose an algorithm that preserves interactions experienced by the operator under significant network latency.
3 **Implementation:** We implement and validate the framework and algorithm in a working system, enabling haptic bilateral teleoperation under considerable network latency.
4 **Characterization:** We characterize key system delays through objective measurements, providing a low-latency baseline performance.
5 **User trials:** Through a user study, we identify latency tolerance thresholds for force and visual feedback.
6 **5G testbed:** We analyze round-trip latency over a 5G standalone network. This demonstrates that our system supports effective haptic bilateral teleoperation with both operator and remote ends connected via 5G.

The rest of this work is structured as follows. Section 2 provides an overview of relevant literature. Section 3 presents

[1] https://surfdrive.surf.nl/files/index.php/s/qrVIfA4SLLoFyIg

the proposed framework for haptic bilateral teleoperation. Section 4.1 describes our proposed algorithm for handling interaction mismatches under network latency. Section 5 outlines the experimental setup used to evaluate the system. Section 6 analyzes the results, including objective measurements, algorithm performance, and findings from a user study. Finally, Section 7 concludes the work.

2 Related Works

The Tactile Internet was initially coined to achieve sub-1 ms latency for applications with tactile feedback, making real-time teleoperation feasible [1]. However, the initial latency requirement has since been shown to be less strict, and closer to sub-10 ms latency [6]. Advances in 5G, Software Defined Networking (SDN), and Network Function Virtualization (NFV) have contributed to reducing latency [11].

Model Mediated Teleoperation (MMT). This addresses network delays by using a predictive model of the remote environment, but this approach faces limitations. The *model jump effect* occurs when updates in the local model cause jarring changes in feedback, affecting operator control. Strategies like delayed updates and adaptive controllers reduce these impacts [8]. MMT also struggles in dynamic environments, especially when interacting with nonlinear objects, due to model mismatch, which can destabilize the system [4].

Human Intent in Teleoperation. Integration of human intent in teleoperation improves adaptability and can reduce latency requirements by allowing predictive adjustments. While intent recognition has been studied in robot-human collaboration [2], its application in MMT for replicating user actions remains limited. Related fields, such as Learning from Demonstration (LfD) and imitation learning, use pre-trained models to transfer human skills to robots [7], though real-time applications present unique challenges.

User Experience Metrics. User experience in teleoperation is challenging to quantify, as traditional control-theoretic metrics may not reflect subjective perception. In the human-machine-interaction field more human-oriented metrics like perceived transparency, Just-Noticeable Difference (JND), and task completion time are used to gauge usability and satisfaction [3, 12].

3 Haptic Bilateral Teleoperation framework

This section outlines a framework for designing a long-distance haptic bilateral teleoperation system that combines live video and predictive force feedback based on Model-Mediated Teleoperation (MMT) principles. As shown in Fig. 2, the framework has three key components: the Operator Domain, the Network Domain, and the Remote Domain.

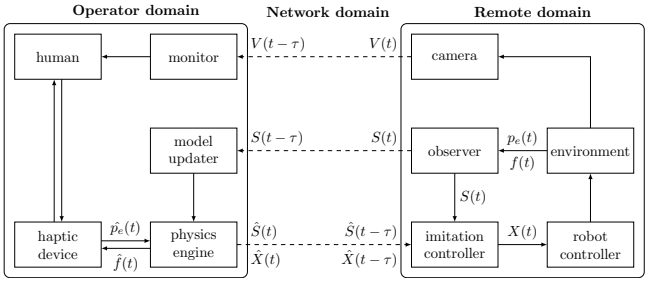

Figure 2: Illustration of our proposed framework.

In the Operator Domain, a human operator interacts with a digital twin of the remote environment using a physics engine and haptic device. The digital twin mirrors the remote environment, giving the operator predictive force feedback for immediate haptic responses. A model updater uses remote measurements to keep the local model current.

In the Remote Domain, a robotic device mirrors the operator's actions, interacting with physical objects in the environment. Sensors continuously monitor the state of the environment, updating the digital twin to maintain alignment between the virtual and physical spaces. In this work, motors manipulating the environment also serve as sensors, further detailed in Section 5. In this work, we denote the counterpart of any parameter θ in the remote domain as $\hat{\theta}$ in the operator's domain.

In the remote domain, S represents the observed state of the environment, including attributes of all observed objects like position, orientation, shape, mass, friction coefficients, center of mass, and inertia. These properties are either provided accurately upfront or estimated and refined during runtime through subsequent observations.

In the Operator Domain, an operator interacts with a digital twin of the remote environment using a physics engine. This digital twin, denoted as $\hat{S}$, conveys the state of the remote environment and is used by the physics engine in the Operator Domain.

The operator uses a haptic device to interact with a local physics engine. The haptic device measures only the position of its *end-effector*, which is the point on the robotic device that interacts with the environment. A haptic rendering algorithm in the physics engine converts the end-effector's position to a corresponding position in the virtual environment and calculates the applied force. The control signal $\hat{X}$ denotes the operator's state in the virtual environment. The predicted applied force, referred to as $\hat{\boldsymbol{f}}$, is returned to the operator. The physics engine can be seen as a function that changes 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 $t-e$ represents the previous discrete step of the physics engine.

The full description of the virtual state $\hat{S}$ and the operator $\hat{X}$ are transmitted to the *remote domain*, arriving with an added network delay of τ. The imitation controller then takes into account the delayed state in the operator domain and the state of the local environment to adjust the control signal, resulting in

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

The imitation controller is designed to act when discrepancies exist between the states S and $\hat{S}$. In those cases, the imitation controller adjusts the control signal to prioritize the operator's intent. Here, operator intent refers to the actions or goals the human operator intends to perform in the remote domain. The method for discerning operator intent and perception is determined beforehand and incorporated into the design of the imitation controller. In this work, we propose the demonstration mismatch algorithm, further explained in Section 4.1.

The control signal drives the robot controller, which manages the specific robotic device. The robot controller operates on the output of the imitation controller. The robotic device measures end-effector position $\boldsymbol{p}_e$ and applied force $\boldsymbol{f}$. Here, the end-effector is the part of the robot that interacts with the environment. The observer consists of sensors in the remote domain that track position in real-time, orientation, and motion of objects. With these measurements and data from the robotic device, the observer estimates the state S of the remote environment. Accurate and fast object tracking is crucial, especially when the data influences the control strategy of the imitation controller.

The observed state of the remote environment is transmitted back to the operator domain, arriving with a network delay of τ. This feedback includes video and force measurements resulting from active remote interactions, which should be relayed to the operator immediately. The measured state of the remote environment, S, is used to update the digital twin in the operator domain, $\hat{S}$, as follows:

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

The update strategy should be designed to disturb the operator minimally. This can involve delaying the model updates when the operator actively interacts with an object, a technique that has shown potential in previous studies [8, 14].

The visual feedback can be treated as a completely independent system. This is beneficial, as any improvements in delay in live streams can be directly used to improve the overall system without the need for integration.

The *network domain* includes all system components responsible for synchronizing data between the operator and remote domains. Teleoperation applications involve various modalities, each with distinct requirements. Kinematic data, which includes positions and orientations of objects, demands low delay but has a small data size, requiring only a few bytes per object. This data is typically resilient to loss because subsequent packets negate the need for retransmission of previous ones. As demonstrated by Kroep et al., a teleoperation setup can maintain a satisfactory user experience even with 50% packet loss [5]. The resilience is enhanced by the robot arm controller functioning as a low-pass filter due to both the robot's acceleration constraints and its lower update frequency (60 Hz versus a 1 kHz packet rate).

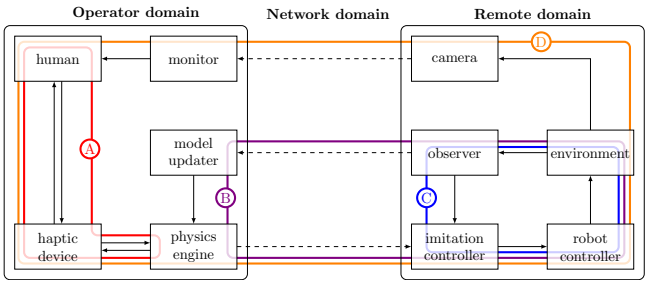

Figure 3: Illustration of critical feedback loops in our proposed framework.

On the other hand, data related to objects' shapes, physical properties, and audio-visual content involve larger data payloads but are more tolerant of delays while requiring higher reliability. Therefore, the network must handle various data types, balancing high-volume transmission with specific delay requirements for each. In complex environments, the rate of data generation may surpass available bandwidth, necessitating the prioritization of crucial data based on the operator's actions and proximity to objects. Effective network design must incorporate advanced protocols, efficient bandwidth usage, data compression, and priority management to address these challenges.

3.1 Critical Feedback Loops

Multiple feedback loops must operate in concert to ensure smooth system functionality, each with specific delay requirements. We outline these critical feedback loops below, as illustrated in Fig. 3, and provide a mathematical basis for calculating allowable delay within each loop.

Predictive Force Feedback (A). This loop handles the predictive force feedback experienced by the operator. When the operator interacts with the haptic device, the movement is captured and processed by the physics engine to generate predictive force feedback, which is then transmitted back to the haptic device without requiring data from the remote environment.

Model Update Feedback (B). This loop updates the local model to reflect changes in the remote environment. Higher delay, here, leads to greater divergence between the local model and the remote environment, increasing the risk of a mismatch between the operator's experience and the robot's actual execution. Maintaining low delay is critical to minimize these discrepancies and keep the operator's experience accurate.

Robot-Object Interaction (C). This loop governs the robot's ability to interact accurately with objects in the remote environment. delay within this loop impacts the robot's responsiveness and precision in replicating actions involving physical contact or manipulating objects.

Motion-to-photon latency (D). This loop covers the time from when the operator performs an action to when the visual outcome appears on the monitor. Low delay in this loop is essential for effective hand-eye coordination. Most of the components in the system directly affect the delay of this loop, including the network. The ability of the non-network components to act at low delay directly influences the network's delay requirements.

The latency in these feedback loops is analyzed in Section 6.1, where we provide empirical measurements to evaluate their baseline performance.

4 Demonstration mismatch algorithm

When the robot imitates the operator's actions, the operator has already experienced predictive feedback as a consequence of his actions. This structure, where the operator receives feedback based on a task yet to be executed, is crucial for meeting delay requirements. Therefore, an algorithm is needed that acts as a self-fulfilling prophecy for the feedback provided to the operator. This algorithm should adjust the robot's trajectory to ensure the imitated task execution closely matches the operator's experienced feedback.

4.1 Consequence of Mismatch

For the analysis in this work, we consider a general scenario where the end-effector of the robot applies a specific amount of pressure to a precise contact point on an object. Assuming accurate tracking methods, there will be minimal differences between the position of the object in the operator environment and the remote environment. This is because, for the system to be satisfactory, a network delay of less than 100 ms is necessary, as shown in Section 6. Thus, we can assume that mismatches will be small, in the range of 1 cm displacement.

Consider a solid object represented by a 2-dimensional manifold in a 3-dimensional space. We assume that the displacement is small enough to consider the local area around the contact point as a tangent space with tangent vector T. The displacement caused by the delay in tracking the object in motion can be divided into a component parallel to the tangent vector T and a component orthogonal to the tangent vector T.

Assuming that the direction of motion of the object is uncorrelated with the tangent vector, the average amount of displacement parallel to the tangent vector can be expressed as:

$$E_{\text{par}} = \frac{2}{\pi}|v_{\text{obj}}|, \tag{1}$$

and the average lateral displacement as:

$$E_{\text{lat}} = \frac{2\sqrt{2}}{\pi}|v_{\text{obj}}|. \tag{2}$$

The lateral displacement will result in the pressure being applied to an incorrect spot. In the context of drawing, ink is deposited in a displaced location. Parallel displacement, however, has a more significant effect. It can determine whether the end-effector touches the object when it should not or fails to touch it when it should. Additionally, parallel displacement can cause variations in the applied pressure.

When the end-effector is already in contact with the object, any parallel displacement into the object will result in increased pressure. This increase in pressure can be modeled using a spring constant k that characterizes the compliance controller of the robot.

Assume that the end-effector is displaced into the object by a distance E_{par}. The force applied by the robot F_{applied} is given by

$$F = kE_{\text{par}} = \frac{2}{\pi}k|v_{\text{obj}}|. \tag{3}$$

Therefore, if the system is tuned for stiff dynamics, represented by a high k, a small displacement will yield a large increase in applied force.

4.2 Position Relative to Object

Since the operator is working within a simulation, both the position of the operator and the position and orientation of the objects are precisely known. This information can be used to extract the operator's actions relative to the object. As the positions and orientations of objects in the remote environment are tracked in real time, this data can be utilized to construct a trajectory for the robot relative to the object. It is important to consider the position and orientation of the object to accurately determine the relative position.

The variables used in the math in this section are all time-dependent. For the sake of clarity, the time dependence is implicit in the expressions for the rest of this section.

To mathematically describe the position and orientation of the operator in the remote domain, while preserving the relative position to the object, we need to translate to the local frame and then back to the global frame for both position and rotation. Let $\hat{\boldsymbol{p}}_{\text{obj}}$ and $\hat{\boldsymbol{R}}_{\text{obj}}$ be the position and the

rotation matrix of the object in the operator domain, respectively. Let $\hat{\boldsymbol{p}}_e$ be the position vector of the end-effector in the operator domain. Let $\boldsymbol{p}_{\text{obj}}$ and $\boldsymbol{R}_{\text{obj}}$ be the position and the rotation matrix of the object in the remote domain.

To obtain the equivalent absolute position in the remote domain that preserves the relative position to the object, we transform the end-effector we get

$$\boldsymbol{p}_e = \boldsymbol{R}_{\text{obj}} \hat{\boldsymbol{R}}_{\text{obj}}^{-1} (\hat{\boldsymbol{p}}_e - \hat{\boldsymbol{p}}_{\text{obj}}) + \boldsymbol{p}_{\text{obj}}$$

where $\hat{\boldsymbol{R}}_{\text{obj}}^{-1}$ is the inverse of the rotation of the object in the operator domain.

When the end-effector is in close proximity to the object, this approach has benefits, but at longer distances problems arise. First of all, it would be odd for the end-effector to respond with precision to an object that is moving slightly while far away. Secondly, any differences in rotation at longer distances will result in significant velocity in the remote end-effector. Finally, this approach does not scale to multiple objects.

4.3 Demonstration Mismatch Algorithm

Preserving the relative position has many benefits in close proximity to an object. However, this effect should diminish as the distance increases. Additionally, a method is needed to generalize the algorithm to any number of objects. A mapping can be created between the two environments, establishing a transition region between the absolute position and the relative position with respect to the object. Depending on the degree of difference between the absolute and relative positions, a short transition region can lead to significant unintended movement of the robot as the operator moves through this region. This added movement can be noticeable to the operator and can impart a significant amount of kinetic energy to the robot that was not present in the operator's actions.

We first consider a single object in the environment. The object can be of any shape, but a method is needed to find the closest point on the object to any given point in space. We define the point on an object that is closest to the end-effector as

$$\hat{\boldsymbol{o}}_i = \arg\min_{\hat{\boldsymbol{o}}_i \in \hat{\boldsymbol{O}}_i} ||\hat{\boldsymbol{p}}_e - \hat{\boldsymbol{o}}_i||,$$

and $\boldsymbol{o}_i$ as the equivalent point of $\hat{\boldsymbol{o}}_i$ on the object in the remote domain.

To map between the operator and remote domains, we first calculate the distance between the closest point on the object in the operator domain and its counterpart in the remote domain. This distance takes into account the difference in position and rotation of the object between the two domains.

A smooth transition function is applied to ensure a gradual change from relative to absolute positioning. We first define a ratio based on the distance of the end-effector to the object and the amount of displacement of the object:

$$\xi_i = \frac{||\hat{\boldsymbol{p}}_e - \hat{\boldsymbol{o}}_i||}{||\hat{\boldsymbol{o}}_i - \boldsymbol{o}_i||}.$$

We then choose the transition between absolute and relative as:

$$f(\xi_i) = \begin{cases} 3\left(\frac{2\xi_i}{3}\right)^2 - 2\left(\frac{2\xi_i}{3}\right)^3 & \text{if } \xi_i \leq 1, \\ 1 & \text{otherwise.} \end{cases}$$

Note that $\max\left(\frac{\partial f(\xi_i)}{\partial \xi_i}\right) = 1$, which ensures that any movement of the operator in one direction in the operator domain cannot result in movement in the opposite direction in the remote domain.

For a single object, we can then obtain an expression for the transformed location of the end-effector in the remote domain as

$$\boldsymbol{p}_e = \hat{\boldsymbol{p}}_e + (1 - f(\xi_i))(\boldsymbol{o}_i - \hat{\boldsymbol{o}}_i).$$

To generalize this approach to any number of objects, we consider the influence of multiple objects simultaneously. Each object will exert a different degree of influence on the end-effector's position, proportional to the inverse of its proximity to the end-effector. Weights are then calculated based on these distances, with each weight being the inverse of $||\hat{\boldsymbol{p}}_e - \hat{\boldsymbol{o}}_i||$. These weights are normalized such that their sum is unity, which leads to

$$w_i = \frac{||\hat{\boldsymbol{p}}_e - \hat{\boldsymbol{o}}_i||^{-1}}{\sum_j ||\hat{\boldsymbol{p}}_e - \hat{\boldsymbol{o}}_j||^{-1}}.$$

The final expression for the transformed location of the end-effector in the remote domain is given by

$$\boldsymbol{p}_e = \hat{\boldsymbol{p}}_e + \sum_i w_i (1 - f(\xi_i))(\boldsymbol{o}_i - \hat{\boldsymbol{o}}_i).$$

The algorithm is illustrated in Fig. 4, where the algorithm adjusts the remote robot's position to ensure that it contacts the remote object at the same point where the operator interacted with the virtual counterpart.

5 Experimental setup

5.1 Remote drawing application with low baseline delay

In this work, we developed a remote drawing application in which a human operator uses a haptic device to control a whiteboard marker attached to a robotic arm. The robotic arm replicates the operator's movements to draw on a moving whiteboard in the remote environment.

To achieve a minimal baseline delay, several design choices were made. First, we opted to emulate network delay by

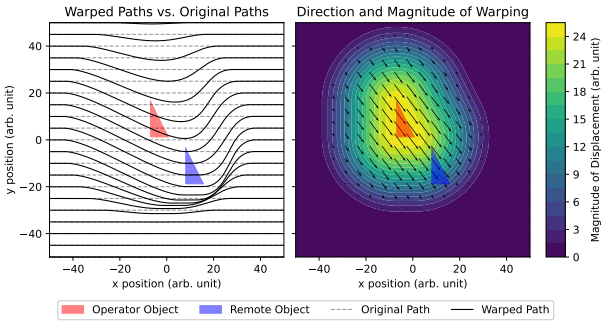

Figure 4: Altering the position of the remote robot position due to a mismatch in position between operator and remote objects. The path of the robot position is shifted so that it hits the remote object at the same spot where the operator hits its virtual counterpart.

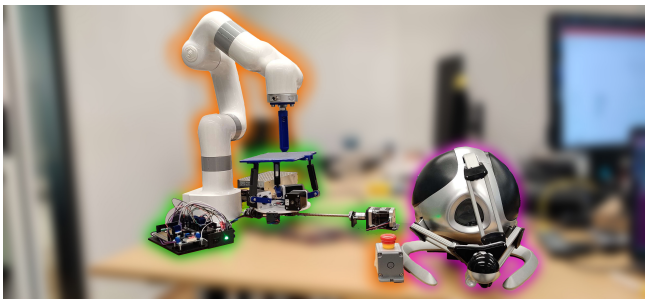

Figure 5: The experimental setup. The robot (Ufactory lite 6) is highlighted in orange with an extension that holds a whiteboard marker, power supply, and emergency button. Highlighted in green is the steward platform attached to a lead screw contraption and the motor drivers. Highlighted in pink is the haptic device (Novint Falcon).

specifically adding a delay to the data going from the operator to the remote domain with NeTem. The video footage is not transmitted over a network; this eliminates the need for encoding and decoding, lowering the delay further.

A second robotic device controls the whiteboard's motion, enhancing safety by restricting movement to defined limits and ensuring consistent behavior for reliable experimental results. The motor angles provide direct position and orientation data, eliminating the need for external sensors. With the whiteboard's path predetermined, a simple predictive model compensates for sensing delays, enabling accurate, zero-delay state perception.

5.2 System overview

The system is structured around the three main domains described in Section 3. The Remote Domain has a Ufactory lite 6 robotic arm fitted with a whiteboard marker. The whiteboard's movements are controlled with a lead screw contraption that provides lateral movement and a 3-RRS parallel manipulator that provides tilt. The whiteboard's position and orientation are determined directly through the motor angles, which act as built-in sensors for tracking the exact pose. These motor readings are sent to the system in realtime, allowing the whiteboard's pose to be accurately monitored without requiring external sensors. Integrating systems in the remote environment is managed through ROS2, which uses topics to connect the robotic arm, whiteboard manipulator, and communication with the Operator Domain. The robotic devices are shown in Fig. 5.

The camera setup in the remote domain is treated as a separate system. A Logitech HD 720p webcam is connected to a Jetson Nano B01, which transmits the video via a *gstreamer* pipeline. The video feed is displayed directly on a monitor. Removing the network entirely for the visual feedback helps to minimize the baseline delay.

The Operator Domain comprises a physics engine (Bullet) and a Novint Falcon haptic device. The Novint Falcon tracks its position at 1 kHz. The physics engine simulates forces and object interactions, updating at 1 kHz. The operator experiences force feedback through the Novint Falcon and visual feedback through a monitor.

Communication between domains occurs within the Network Domain via UDP. The network is simulated through NetEm, where the delay is only added to the packets from the operator to the remote domain, which ensures that visual delay is affected in a representative way despite there not being a network between the camera and monitor. The delay is adjusted with NetEm to simulate varying network delays.

5.3 5G Network Latency

In addition to the primary experimental setup, we conducted a separate test to evaluate end-to-end latency over a 5G wireless network. The setup is a private 5G standalone deployment that uses the n77 frequency band with exclusive network use during the tests. We connect the remote and operator domains to the 5G-NR routers. The remote domain is connected to a Siretta QUARTZ 5G Industrial Router, and the operator domain is connected to the Advantech ICR-4453W1S. Both sides have network-provisioned SIM cards that ensure a 5G connection. Latency-related data is collected using the internal telemetry features of the routers.

During the tests, three aspects were explored. First, there is a difference between a single 5G link at the remote domain and a wired operator domain, and both domains have a 5G link. Second, the difference between a small 64 B packet and a larger 1.4 kB packet. Finally, we distinguish between a connection where a PDU session has yet to be established

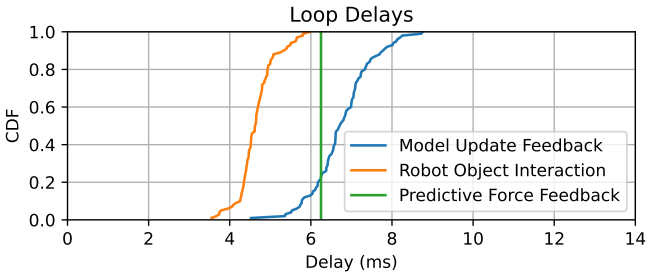

Figure 6: CDFs of latency for the predictive force feedback, model update feedback, and robot-object interaction. The predictive force feedback is characterized with a 480 Hz camera resulting in 2 ms precision.

for data transmission and one where a PDU session is active (pre-established).

5.4 User Study

We conducted a user study to evaluate system usability in a haptic bilateral teleoperation context. This study aimed to determine subjective latency tolerance and tipping points for both the network and force feedback delays.

Each participant began by selecting a complex drawing task they could reproduce consistently. After familiarization with the setup, participants were presented with a series of test cases that included varying network latencies (0, 25, 50, 75, 100, 125, 150, and 200 ms) and force feedback delays (0, 5, 10, 15 ms, and off). The drawing task was performed on a moving whiteboard for each test case. Following each test case, participants rated their experience on a 7-point Likert scale, focusing on perceived control, usability, and overall experience compared to the baseline condition. A total of 17 participants, aged 20 to 35, with technical backgrounds, participated in the study.

6 Results

6.1 Baseline System Latency

To characterize baseline system performance, we performed both internal and external measurements. These measurements quantify the baseline latency of the five feedback loops identified in Section 3.1. For all measurements, no additional emulated network latency was added.

Local Predictive Force Feedback.. It is the time between an operator's input and the resulting force feedback response. To measure this, the Novint Falcon is positioned so that its proxy makes direct contact with a highly rigid virtual surface. The Novint Falcon is then moved slightly toward the surface, and the time it takes for the device to accelerate in the opposite direction is used as an approximation of the predictive force feedback delay. This delay is observed using a 480 Hz high-speed camera. The results of these measurements is shown in Fig. 6, with an average delay of 6.25 ms.

Model Update Feedback. It is the time it takes for an action to manipulate an object to be sent from the operator domain to the remote domain, for the robot to execute that action, for the environment model to be updated, and for this information to be sent back to the physics engine in the operator domain. Key elements contributing to this delay include ROS2 topic delays, operator-remote communication, whiteboard motor communication, and forward kinematics processing. A cumulative distribution function of these measurements is shown in Fig. 6, with 7.36 ms average delay.

Robot-Object Interaction. It is the time it takes for the robot controller to recognize a change in whiteboard position. Major delay contributors include ROS2 topic delays, whiteboard motor communication, and forward kinematics processing. A cumulative distribution function of these measurements is shown in Fig. 6, with 5.12 ms average delay.

Glass-to-Glass Latency. It is the total delay from the moment a physical event is captured by a camera to its appearance on a display. This latency is measured with a high-speed camera filming both a motor-driven clock hand and its delayed display on a monitor. We calculate the delay by analyzing the angle difference between the real and displayed positions based on the motor's constant rotation speed. The average delay is 52.52 ms.

Motion-to-Photon Delay. It is the time from when a user initiates a movement until the corresponding action appears on the monitor. Nearly all components in the system contribute to this delay. To measure it, a high-speed camera records the moment the Novint Falcon is struck and the corresponding movement displayed on the monitor. Frame counting is used to estimate the delay, with an average delay of 57.49 ms, most of which is attributed to the glass-to-glass latency.

6.2 5G Network Latency Results

The 5G latency experiment results are presented in Fig. 7. Three factors were tested: single 5G link against dual 5G links, 64 B against 1.4 kB packet sizes, and active against not yet established PDU sessions.

Effect of an Ongoing PDU Session. Establishing a new PDU session significantly increases Roundtrip Time (RTT). Maintaining an ongoing PDU session is crucial for minimizing latency, which is ideal for applications like haptic bilateral teleoperation that require a continuous data stream in both directions.

Effect of Packet Size. In all scenarios, larger packet sizes increase latency. For a single link with an established PDU session, the mean RTT for 64 B packets is 10.6 ms, while for 1.4 kB packets it is 16.1 ms, with a difference of 5.5 ms. Such a

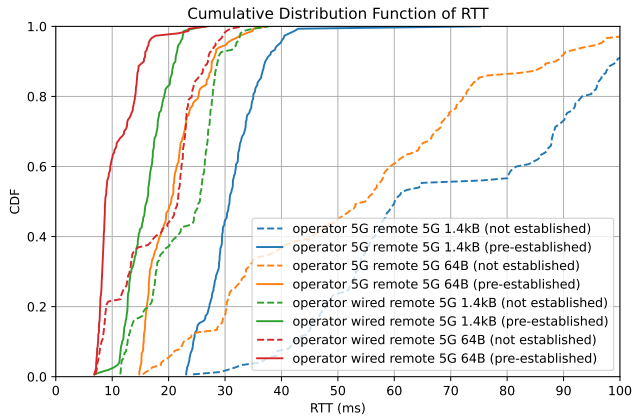

Figure 7: 5G network latency, comparing single vs. dual 5G links, effects of packet size, and impact of active (pre-established) vs. not yet established PDU sessions on Roundtrip Time (RTT). The results show that maintaining an active PDU session and using smaller packets minimizes latency, which is critical for applications like haptic bilateral teleoperation.

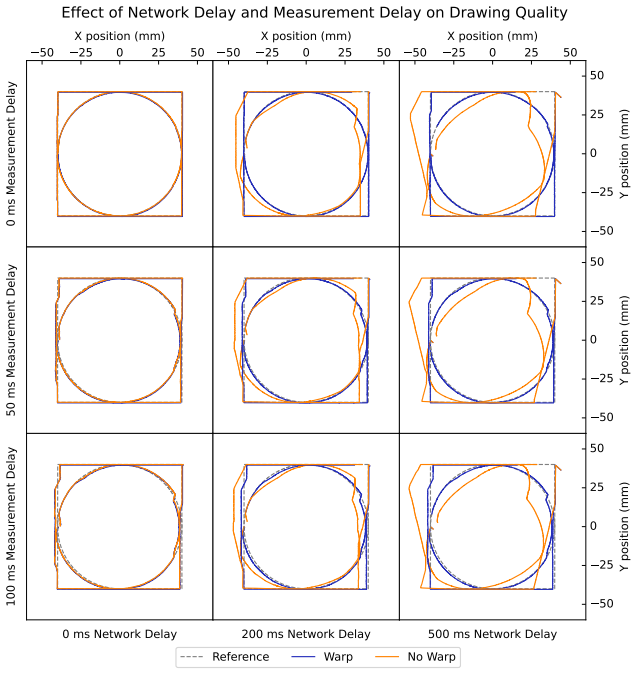

Figure 8: Drawings with and without the algorithm under network delays and measurement delays. Each grid item shows a reference (- -), a replication without the algorithm (—), and with the algorithm (—).

difference is significant for haptic bilateral teleoperation and suggests that smaller packets should be prioritized where possible.

Effect of Dual 5G Links. The addition of a second 5G link approximately doubles the RTT, highlighting that latency is primarily due to wireless links. Each 5G link adds a delay cost between 10 ms and 16 ms, depending on packet size.

6.3 Algorithm performance subject to network and measurement delay

To evaluate the proposed algorithm, we conducted an experiment in which the operator domain followed a preprogrammed reference path, drawing a square with a perfectly inscribed circle. The whiteboard moved back and forth with the lead screw mechanism during the drawing.

This experiment assessed the drawing accuracy with and without the algorithm under different network and measurement latency conditions. Here, measurement latency specifically refers to the delay between the whiteboard's movement and the robot controller receiving that information. The resulting drawings are shown in Fig. 8.

Measurement Latency Impact. Measurement latency affects both methods similarly, introducing offsets in the drawn path as delays increase. This outcome is expected, as the algorithm does not compensate for local measurement delays.

Network Latency Impact. Network latency greatly impacts drawing accuracy in the absence of the algorithm. With the algorithm, the drawing closely matches the reference, regardless of network latency.

In summary, the algorithm significantly enhances robustness to network latency but does not mitigate local measurement latency, which should, therefore, be minimized.

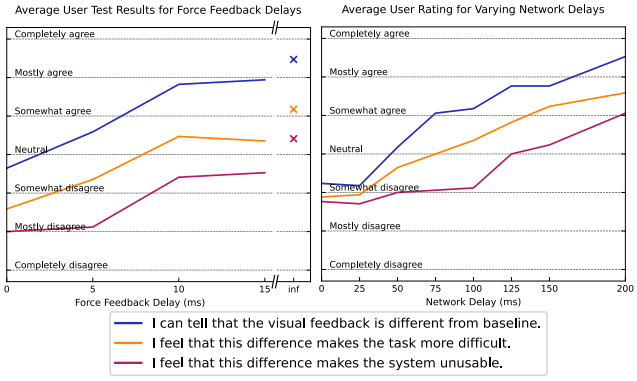

Figure 9: User study results showing the impact of force and visual feedback latency on the user experience.

Latency Requirements for Haptic Bilateral Teleoperation			
	Perception	Hindrance	Unusable
Force Feedback	9 ms	16 ms	>16 ms
Motion to pixel	133 ms	158 ms	>208 ms

Table 1: Requirements for haptic and visual delay in a remote drawing application. Note that the reported force feedback includes 6 ms of baseline latency and artificially added latency. Likewise, the motion-to-pixel latency includes 58 ms of baseline latency.

6.4 User study

To assess the system's requirements, we conducted a user study examining drawing accuracy and user experience under various levels of force feedback and visual latency. Participants tested multiple system configurations against a baseline they had become familiar with. They were asked to indicate whether they perceived any differences, if these differences impaired their ability to draw, and at what point the system became unusable.

Force Feedback Latency Sensitivity. As shown in Fig. 9, users noticed delays in force feedback almost immediately, with latencies above 5 ms negatively impacting control and perceived responsiveness. Without force feedback, participants felt a significant disconnect from the robot. However, even at the maximum tested latency of 10 ms, most still found the system usable. It is important to note that latency for force feedback is locally predicted and unaffected by network latency.

Visual Feedback Latency Sensitivity Visual latency is influenced by network latency. Fig. 9 includes added network latency, which should be considered on top of the system's baseline motion-to-pixel latency of 58 ms. Users began noticing delays at around 50 ms of added latency (total 108 ms), but did not find the system harder to use until 100 ms (total 158 ms). At 200 ms (total 258 ms), most users found the application unusable. These findings are summarized in Table 1. They indicate that haptic teleoperation systems may tolerate significantly higher latencies than traditionally recommended. Further studies are needed to refine requirements for different applications.

When comparing the network latency levels in the user study to the specifications measured for a 5G connection, we conclude that this system can deliver a satisfactory user experience even when both the operator and remote domains are connected via a 5G wireless network.

7 Conclusions

Haptic bilateral teleoperation offers immense potential for telemaintenance, remote manipulation, and disaster response applications. Still, it is hindered by the challenge of delivering accurate, low-latency force and video feedback to the operator. This work enhances teleoperation by integrating live video with Model Mediated Teleoperation (MMT) to provide predictive force feedback.

Central to our approach is an algorithm that enables a robotic device to replicate interactions predictively experienced by the operator, creating a "self-fulfilling prophecy" effect. We implemented and validated this framework in a functional system, achieving satisfactory haptic bilateral teleoperation under substantial network delays. Video footage of the experimental setup is provided at this URL[2] .

We identified four critical feedback loops and extensively characterized their latency for our system. We conducted a user study that demonstrates that a human operator does not perceive the presence of a motion-to-pixel latency of 133 ms, with a network latency of 75 ms.

We analyzed the round-trip latency when using a 5G standalone network and demonstrated that our system supports effective haptic bilateral teleoperation with both operator and remote ends connected via 5G. This insight sets a path away from unattainable latency requirements, moving toward practical and feasible haptic teleoperation using currently available technologies.

Acknowledgments

This research is supported by the FIDAL project (Field Trials beyond 5G with GA No 101096146), funded under the European Union's Horizon Europe research and innovation program (6G SNS). The fourth author also acknowledges support from the Future Network Services (FNS) project, funded by the National Growth Fund as part of the Dutch 6G flagship initiative.

References

[1] Gerhard P Fettweis. 2014. The tactile internet: Applications and challenges. *IEEE Vehicular Technology Magazine* 9, 1 (2014), 64–70.

[2] Fanny Ficuciello, Luigi Villani, and Bruno Siciliano. 2015. Variable impedance control of redundant manipulators for intuitive human–robot physical interaction. *IEEE Transactions on Robotics* 31, 4 (2015), 850–863.

[3] S. Hirche and M. Buss. 2012. Human-Oriented Control for Haptic Teleoperation. *Proc. IEEE* 100, 3 (March 2012), 623–647. https://doi.org/10.1109/JPROC.2011.2175150

[4] C. Kim and D. Yong. 2022. Analysis of Effect of Model Mismatch on Stability of Model-Mediated Teleoperation, Vol. 2022-March. https://doi.org/10.1109/HAPTICS52432.2022.9765604

[5] HJC Kroep, Vineet Gokhale, J Verburg, and R Venkatesha Prasad. 2023. Etvo: Effectively measuring tactile internet with experimental validation. *IEEE Transactions on Mobile Computing* (2023).

[6] Kees Kroep, Vineet Gokhale, Ashutosh Simha, R Venkatesha Prasad, and Vijay S Rao. 2023. 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)*. 199–208.

[2] https://surfdrive.surf.nl/files/index.php/s/qrVIfA4SLLoFyIg

[7] An T Le, Meng Guo, Niels van Duijkeren, Leonel Rozo, Robert Krug, Andras G Kupcsik, and Mathias Bürger. 2021. Learning forceful manipulation skills from multi-modal human demonstrations. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 7770–7777.

[8] Probal Mitra, Diana Gentry, and Gunter Niemeyer. 2007. 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, 268–273.

[9] Probal Mitra and Günter Niemeyer. 2008. Model-mediated telemanipulation. *The International Journal of Robotics Research* 27, 2 (2008), 253–262.

[10] Syeda Nadiah Fatima Nahri, Shengzhi Du, and Barend Jacobus Van Wyk. 2022. A review on haptic bilateral teleoperation systems. *Journal of Intelligent & Robotic Systems* 104 (2022), 1–23.

[11] Nattakorn Promwongsa, Amin Ebrahimzadeh, Diala Naboulsi, Somayeh Kianpisheh, Fatna Belqasmi, Roch Glitho, Noel Crespi, and Omar Alfandi. 2020. A comprehensive survey of the tactile internet: State-of-the-art and research directions. *IEEE Communications Surveys & Tutorials* 23, 1 (2020), 472–523.

[12] Yaguang Tao, Alan Both, Rodrigo I. Silveira, Kevin Buchin, Stef Sijben, Ross S. Purves, Patrick Laube, Dongliang Peng, Kevin Toohey, and Matt Duckham. 2021. A comparative analysis of trajectory similarity measures. *GIScience & Remote Sensing* 58, 5 (July 2021), 643–669. https://doi.org/10.1080/15481603.2021.1908927 Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/15481603.2021.1908927.

[13] Xiao Xu, Burak Cizmeci, Clemens Schuwerk, and Eckehard Steinbach. 2016. Model-mediated teleoperation: Toward stable and transparent teleoperation systems. *IEEE Access* 4 (2016), 425–449.

[14] Xiao Xu, Clemens Schuwerk, and Eckehard Steinbach. 2015. Passivity-based model updating for model-mediated teleoperation. In *2015 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*. IEEE, 1–6.