The human touch for teleoperation

Kroep, Kees

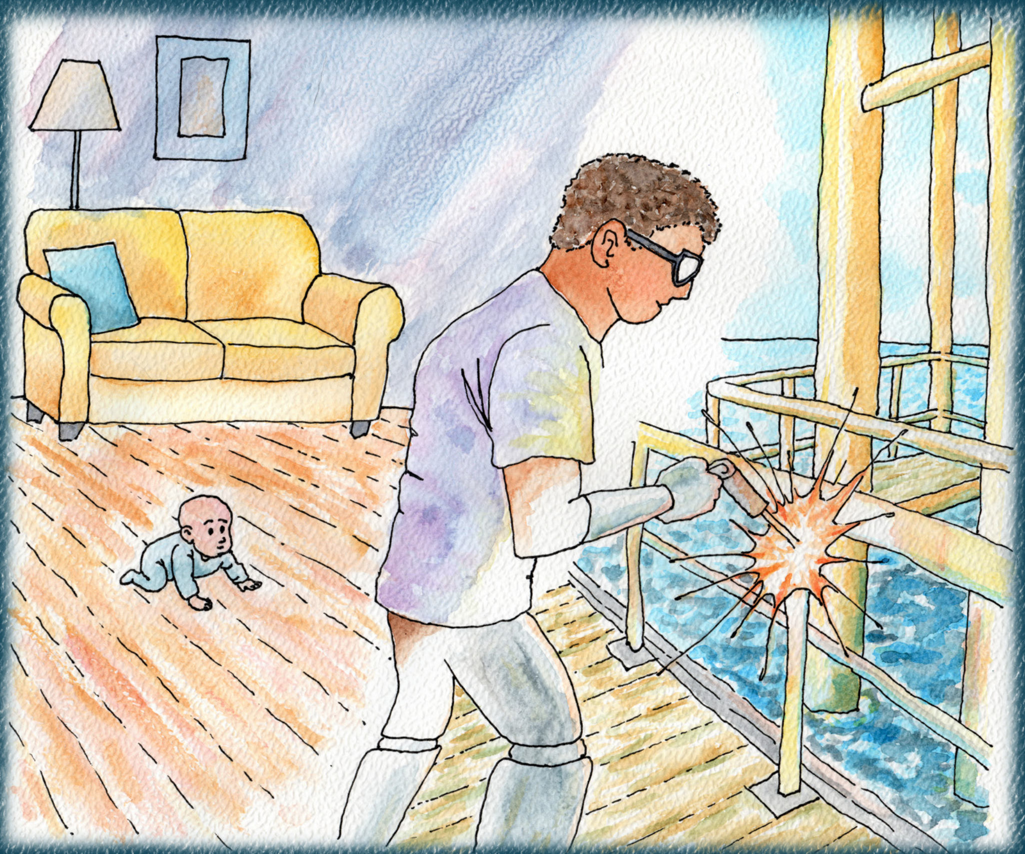
DOI
10.4233/uuid:5c977d98-2c74-4f5d-bb18-b77aa6b7d825

Publication date
2025

Document Version
Final published version

Citation (APA)
Kroep, K. (2025). *The human touch for teleoperation*. [Dissertation (TU Delft), Delft University of Technology]. https://doi.org/10.4233/uuid:5c977d98-2c74-4f5d-bb18-b77aa6b7d825

THE HUMAN TOUCH



FOR TELEOPERATION

H.J.C. Kroep

The human touch for teleoperation

THE HUMAN TOUCH FOR TELEOPERATION

Dissertation

for the purpose of obtaining the degree of doctor
at the Delft University of Technology,
by the authority of the Rector Magnificus Prof.dr.ir. T.H.J.J. van der Hagen,
chair of the Board of Doctorates,
to be defended publicly on Thursday 11 September 2025 at 15:00 o'clock.

by

Herman Johannes Cornelis KROEP

Master of Science in Computer Engineering,
Delft University of Technology, The Netherlands,
born in Amsterdam, The Netherlands.

This dissertation has been approved by the promotors

promotor: prof. dr. K.G. Langendoen
promotor: dr. R.R. Venkatesha Prasad

Composition of doctoral committee:

Rector Magnificus,	Chairman
Prof. dr. K.G. Langendoen	Delft University of Technology, *promotor*
Dr. R.R. Venkatesha Prasad	Delft University of Technology, *promotor*
Independent members:	
Prof. dr. ir. G. N. Gaydadjiev	Delft University of Technology
Prof. dr. S. Giordano	University of Pisa, Italy
Prof. dr. M. Zimmerling	TU Darmstadt, Germany
Prof. dr. A. M. L. Kappers	Eindhoven University of Technology, The Netherlands
Dr. inż. L. Ambroziak	Bialystok University of Technology, Poland
Reserve member:	
Prof. dr. J. Dankelman	Delft University of Technology

Keywords: telepresence, human in the loop, human computer interaction, tactile internet, digital twins, ultra reliable low latency communication, model mediated teleoperation

Front & Back: Watercolor and fountain pen drawing by H. J. C. Kroep.

ISBN 978-94-6384-841-1

An electronic version of this dissertation is available at http://repository.tudelft.nl/.

To be rooted is perhaps the most important and least recognized need of the human soul.

Simone Weil

SUMMARY

After the potential of this work is realized, people will be able to physically manipulate remote environments. For example, a skilled artist in Tokyo could paint delicate calligraphy on a canvas in Paris, feeling each stroke as if they were local. A surgeon in London could operate on a patient in a remote village, sensing the precise resistance of tissue through robotic instruments. A firefighter in Los Angeles could save people from a burning building without the need to put his own life at stake. Extending our human touch across great distances opens doors to new forms of work, collaboration, and human connection without needing physical presence.

Realizing this vision requires the successful implementation of Haptic Bilateral Teleoperation (HBT). An HBT system must fulfill two core requirements: precise replication of the operator's actions by a remote robot and accurate, responsive feedback to guide those actions. These requirements are inherently subjective, varying across individuals, tasks, and applications, adding significant complexity to both the system design and evaluation.

At first glance, realizing HBT may seem an insurmountable challenge. Conventional wisdom suggests that the stringent network requirements, such as ultra-low latency and near-perfect reliability, far exceed the capabilities of current network technology. The latency constraints are so strict that even fundamental physical limits, such as the speed of light, impose onerous restrictions on the maximum feasible distance between the operator and the remote environment.

Overcoming these challenges demands a holistic approach. On the one hand, we must push network technology to its limits, striving for lower latency, higher reliability, and optimized communication protocols explicitly tailored for HBT applications. On the other hand, we must also explore alternative approaches that lower the network requirements of HBT systems, especially the latency requirement. For both of these directions, it is essential to have a deep understanding of the entire HBT system, particularly the role of the human operator. Unlike most systems, where performance is measured through objective metrics, HBT introduces a distinctive challenge: HBT systems must be designed for both technical performance and the user's subjective experience.

In this dissertation, we first provide a deeper understanding of HBT systems and examine how network behavior influences user experience. In particular, we identify the underlying reasons behind the stringent network requirements. First, through multiple repeated user studies, we demonstrate that the reliability of the kinematic demands and force modalities is low, especially at the packet rate 1 kHz. Even with 50%, packet loss, we demonstrate that users are largely unaffected due to strong temporal correlation in these modalities.

More importantly, we pinpoint the fundamental cause of the strict low-latency requirement. It is not merely the presence of delay but rather the unintended forces that arise due to the combination of active force feedback and a closed-loop control system.

This interaction is unique because users do not perceive latency directly. Instead, they experience the resulting unnatural forces.

Because the main cause for the stringent network requirements is so specific, it provides a clear target for research. Next, we explore multiple approaches to address this particular interaction, which is the primary source of stringent latency constraints. First, we optimize the MAC protocols with a strict focus on minimizing latency for both the kinematic and force modalities. Next, we investigate methods to manipulate the transmitted data in a way that does not impede the human operator, aiming to mitigate the adverse effects of network latency on force feedback. Finally, we take a more radical approach by moving away from direct transmission of force feedback altogether, instead leveraging predictive models to estimate force feedback locally.

An important insight from this dissertation is the path forward for HBT systems. Future HBT systems should integrate predictive force feedback with live video transmission, leveraging the advantages of each modality. Predictive force feedback offers a viable alternative to the stringent latency constraints of transmitted force feedback. Minor inaccuracies in force feedback are often imperceptible to human operators. Meanwhile, live video transmission circumvents the complexities of visual prediction while operating within a latency range of approximately 100 ms. This is significantly more feasible than the 1 ms latency required for direct force feedback transmissions.

This dissertation has three important takeaways. First, it provides a deeper understanding of how network performance shapes user experience in HBT. Second, it demonstrates alternative approaches that enable HBT beyond direct network improvements. Third, it proposes a path forward that integrates live video with predictive force feedback. Despite these advancements, significant challenges remain. Scaling HBT to highly dynamic environments, where unpredictability complicates prediction of force feedback, remains a major hurdle. Additionally, managing discrepancies between the operator's predictive experience and the actual remote events is crucial to maintaining intuitive and stable interactions. While these challenges persist, none appear insurmountable. With continued progress, HBT can become a transformative technology, opening doors to new forms of work, collaboration, and human connection without needing physical presence.

CONTENTS

1

Introduction

With technological advancements, humans have continually expanded their capabilities. At the click of a button, we can now extend our eyes and ears to observe events of the past, anticipate the future, and experience real-time happenings across the globe.

In this work, we explore the next step in human empowerment: expanding our physical reach over great distances and transferring human skill to where it is needed most. Imagine what this future could look like.

1.1. The remote calligraphy artist

We start by providing an illustrative example. Consider a homeowner who wishes to personalize his living space with traditional Kanji characters, symbols that convey meaning and aesthetic beauty. To achieve this, he decides to commission a renowned calligrapher from Japan. The logistical and financial impracticalities of transporting the artist across the globe for this task prompt a search for alternative solutions. The homeowner would like to let the calligrapher perform his craft without leaving his country. A robotic device is placed in the home, and a link is established between the robot and the calligrapher, allowing him to execute his craft remotely.

For the calligrapher to effectively perform his art remotely, he need access to several crucial capabilities:

1. The capacity to monitor the remote environment while facilitating precise hand-eye coordination.
2. The ability to sense the pressure exerted on the brush to apply the strokes well.
3. The capability to control the robotic device to mirror his actions at the remote site.

For all three challenges, it can be difficult to fully grasp the complexity of delivering a satisfying user experience. To help illustrate these challenges, we will demonstrate them through a series of simple experiments that can be easily followed while reading. In this way, the reader can experience some of the underlying concepts firsthand, providing some intuition behind the themes explored in this thesis.

1.1.1. HAND-EYE COORDINATION

The first required capability is the ability to perform hand-eye coordination. The artist must see their strokes as they form, adjusting the angle and speed in real time. A delay in visual feedback can disrupt this delicate balance, leading to inaccuracies. To experience the negative effects of video delay yourself instructions are provided in experiment 1.

Experiment 1: Hand-eye coordination

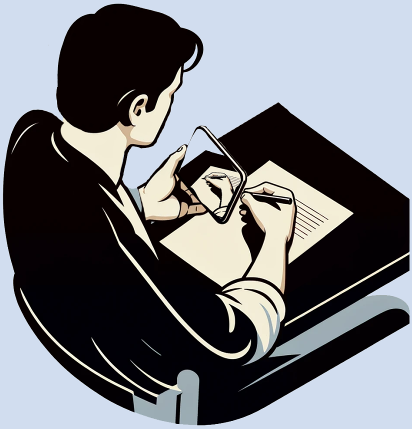


Figure 1.1: Illustration of try-it-yourself hand-eye coordination experiment.

This is a short experiment designed to experience the negative effects of video latency on hand-eye coordination. You need a pen, a piece of paper, and a smartphone with a camera app. The experiment is illustrated in Figure 1.1.

1. Open the camera app on your smartphone.
2. Position the phone so the camera is pointing at the paper where you will be writing, and the screen blocks your direct view of your pen tip.
3. Write on the paper while focusing your gaze on your phone's screen, using only that as your only visual guide.
4. Alternate between looking at your phone screen and directly at your pen tip and observe the differences in experience.

The latency introduced by the smartphone camera app can vary. For example, if less processing is done on the phone before displaying the picture, the latency will drop. One way to approximate this latency is by filming a stopwatch with the smartphone and then taking a picture of both the stopwatch and the camera screen together. The difference between the two times approximates the camera's latency. At the time of writing, we measured this latency to be around 250 milliseconds for our smartphone camera apps. Under these conditions, we observed a clearly noticeable delay that significantly worsens the writing experience.

The experiment above aims to demonstrate the discomfort and difficulty caused by the round-trip delay between executing an action and receiving a visual response. Even without a networked connection, this latency already has a significant impact on task performance. This underscores the critical challenge of providing low-latency video feedback, particularly when accounting for the additional delays introduced by a network.

1.1.2. FORCE FEEDBACK

The calligrapher requires precise control over the pressure applied with the brush, as varying pressure levels significantly affect the final drawing. Unlike visual feedback, force feedback directly influences how the user interacts with their environment. It is not merely about perceiving resistance but actively responding to it. A useful analogy is pressing down on a table: the table provides resistance, preventing further movement regardless of the operator's actions or experience. The operator's actions—such as leaning or applying pressure—must be met with immediate and proportional feedback to avoid impossible outcomes, like passing through the table. A crucial realization is that when subject to time delay, it is not the operator's perception that breaks down, but the physical interaction itself. Instructions are provided to experience this concept in experiment 2.

Experiment 2: Tapping the table

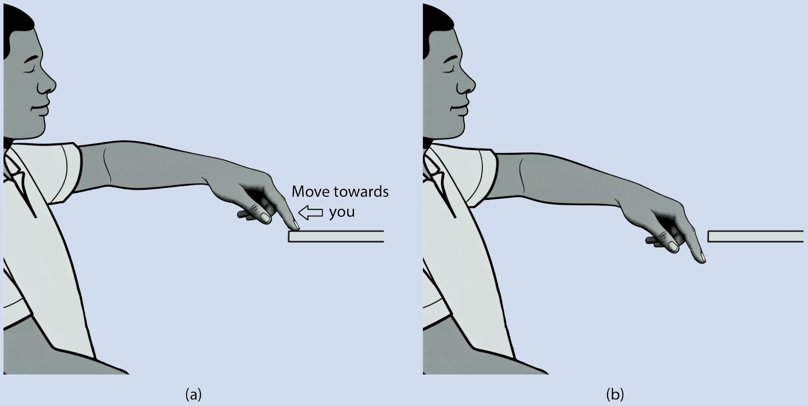


Figure 1.2: Illustration of experiment involving tapping on the table with your eyes closed. First you start tapping on the table (a) and then move your hand closer towards you until you no longer hit the table (b).

Here is a simple experiment you can perform to gain intuition on the interaction between operator action and force feedback. The experiment is illustrated in Figure 1.2.

1. Position yourself in front of a desk or table.
2. With your eyes open, repeatedly tap against the edge of the desk with moderate force.

3. Close your eyes and move your hand towards you until you no longer hit the table.
4. Pay attention to the movement of your finger once it passes the edge. Observe the depth to which your finger dips below the table surface.
5. Repeat steps 2 through 4, but this time, apply only a very light touch. Notice the difference in how far your finger dips.

The purpose of the experiment is to demonstrate that meaningful interaction with the environment is only possible when the environment accurately responds to the operator's actions. For instance, if a virtual table fails to stop the operator's hand, it does not imply an intent to break the table; rather, the expectation is that the table will exert just enough force to halt the motion. Facilitating such interactions requires force feedback with a latency of at most 1 ms to ensure a good interaction [1].

1.1.3. MAKING THE ROBOT MOVE

Thus far, we have examined the visual and force feedback provided to the calligrapher. However, a crucial element remains: the ability to manipulate the robotic device. This introduces an additional layer of delay, compounding the existing challenges posed by visual and force feedback. The calligrapher's movements in Japan must be captured, transmitted to the robotic device at the homeowner's location, and accurately replicated by the robot.

More energy must be expended for the robot to recreate the operator's motions with minimal delay to reduce latency between the received signal and the robot's movements. However, this increased energy makes it more difficult for the robot to apply light pressure and raises the risk of overshooting. A slower response could mitigate these issues, but time for such a delay is a luxury that often cannot be afforded. To illustrate this challenge further experiment 3 is provided.

Experiment 3: Finger tracking

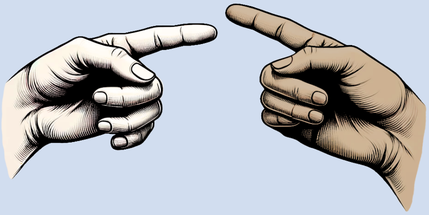


Figure 1.3: Illustration of the experiment with one pointing finger tracking another pointing finger.

For this experiment, two people are needed. One will act as the leader and the other as the follower. The leader uses his pointing finger to track a trajectory through the air, while the follower uses his pointing finger to track the pointing finger of the leader. The experiment is illustrated in Figure 1.3. Several factors can be varied, in no particular order:

1. The predictability of the leader's trajectory
2. The overall speed of the leader's trajectory
3. The smoothness of the leader's trajectory
4. The aggressiveness with which the follower is tracking the leader

The purpose of the experiment is to demonstrate that latency between the operator and the robot is unavoidable unless the operator's motions are highly predictable, enabling the robot to anticipate actions well in advance. The faster and more abrupt the operator's movements, the more pronounced the robot's lag becomes. Smooth trajectories are easier for the robot to follow accurately than erratic ones. When the robot attempts to track the operator's movements aggressively, it accelerates more, increasing the likelihood of overshooting.

The remote calligraphy artist serves as an example of the functionality we want to achieve: the ability to physically manipulate a remote environment, while getting both visual and active force feedback from the interaction. But what would be the consequences of deploying such functionality globally? What if this was incorporated into peoples daily lives?

Imagine a world where the necessity for physical transportation for both observation and interaction is significantly reduced. Our ability to contribute anywhere would no longer require our physical presence. Remote work would reach past desk jobs, allowing us to work from anywhere to anywhere for most tasks. The dependence on transport infrastructure and large cities would diminish, enabling people to live wherever they choose while maintaining access to work, amenities, and services.

Reducing the need for transportation alone could yield substantial benefits, including cost savings, decreased time spent in traffic, faster aid response times, and mitigated climate change impacts. Individuals could choose to live in communities and environments they prefer without affecting their job opportunities or being tethered to high housing costs in urban centers. People could spend more time with their families and less time commuting. This would especially empower working couples to spend more time with their kids for a broader selection of work fields.

Besides being able to work from anywhere, this technology holds the promise of enabling humans to perform tasks in hard-to-reach places, such as modifying satellites in orbit, rescuing people from burning buildings, or cleaning up nuclear disaster sites like Fukushima. It could also empower humans to accomplish feats beyond their natural capabilities. Instead of manipulating a similarly sized robot, one could control a large robotic device designed to handle objects far exceeding human carrying capacity. Alternatively, one could operate a very small robotic device to execute actions with a level of

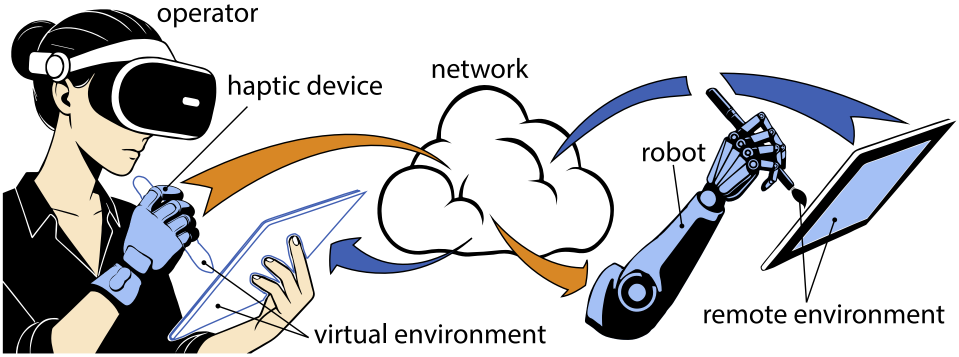


Figure 1.4: An illustration of a HBT application. The operator interacts with a haptic device. The haptic device records the actions of the operator and sends them over the network to a robot in a remote environment. The robot imitates the operator's recorded actions in the remote environment. Force and visual feedback are sent through the network to the operator. Finally, the operator experiences the visual and force feedback through a monitor and haptic device, respectively. In this case, the haptic device is the glove the operator is wearing. The operator virtually observes the remote environment through a VR headset.

precision unattainable by human hands alone. The possibilities are endless.

At the heart of this envisioned future lies the concept of Haptic Bilateral Teleoperation (HBT), which enables humans to manipulate a remote environment through a robotic device.

1.2. REQUIREMENTS OF HAPTIC BILATERAL TELEOPERATION

Let us specify a HBT system and examine its main challenges. An overview of such an application is shown in Figure 1.4.

In this work, we define HBT based on how a system should function. For a HBT system to be deemed effective:

1. The system must facilitate the operator to convey their intended actions in a manner that results in precise and satisfactory imitation by a remote robotic device.

2. The system must provide the operator with accurate video and force feedback to support their actions.

Both of these requirements are subjective, which means they can vary significantly among individuals, applications, and levels of importance. For example, a system that falls short for a high-stakes, precise task like a remote surgery might still be deemed acceptable for a less critical task, such as remote repair.

1.3. FROM A NETWORKING VANTAGE POINT

To improve HBT, one can target specific components of the system and optimize their performance based on key metrics. Moreover, advancements from broader research efforts can also drive significant performance improvements. In recent years, this phenomenon has contributed to notable progress, with the pursuit of low-latency network connections being a prime example.

If all performance indicators of a component are either the same or better, the new component will outperform its predecessor. However, it is beneficial to be able to make informed trade-offs, which requires delving deeper into the application as a whole. In the case of networking, a relevant trade-off is if it is worth it to accept higher packet loss in exchange for lower round-trip latency. Furthermore, it is difficult to determine clear requirements for each component.

The Tactile Internet (TI), a term introduced by Fettweis et al., provides a set of requirements tailored for applications like HBT [1]. At its introduction in 2014, TI called for a mere one millisecond of round-trip delay and a reliability of 99.99999%. An assessment was made that throughput was not the limiting factor in these types of applications. The proposed requirements, if met, provide a level of performance that would enable the intended applications to bear fruit.

The concept of the TI has played a significant role in shaping the direction of Ultra-Reliable Low-Latency Communication (URLLC) efforts [2, 3, 4]. However, it presents a critical issue: the exact specifications it advocates lack thorough justifications. A latency of 1 ms is practically unattainable, and there is value in critically assessing these requirements to explore alternative approaches with more attainable network requirements. This mismatch renders the TI a double-edged sword. On one hand, it serves as a beacon, guiding research and development toward low latency and high reliability. On the other hand, it potentially dissuades researchers from pursuing alternative solutions that, while deviating from these strict requirements, could offer significant advancements.

1.4. CHALLENGES IN REALIZING HAPTIC BILATERAL TELEOPERATION

Enabling a calligraphy artist to create a remote painting requires significant effort, requiring improvements in multiple areas, such as network performance and application development. In this work, we aim to chart a path toward the first practical application of HBT over long distances. We aim to achieve a point where a human operator can have a satisfactory experience while performing a task remotely. The research goal of this thesis is stated as follows:

How to realize haptic bilateral teleoperation across long distances?

The first focus on characterizing the performance of a network facilitating a HBT application. We consider the following subquestions:

Sub-Question 1: How can we characterize network performance when used to transmit kinematic data with a low-latency requirement?

Sub-Question 2: What is the correlation between network performance and the specifics of the system?

Next, we focus on improving the performance of HBT systems. We first focus on optimization.

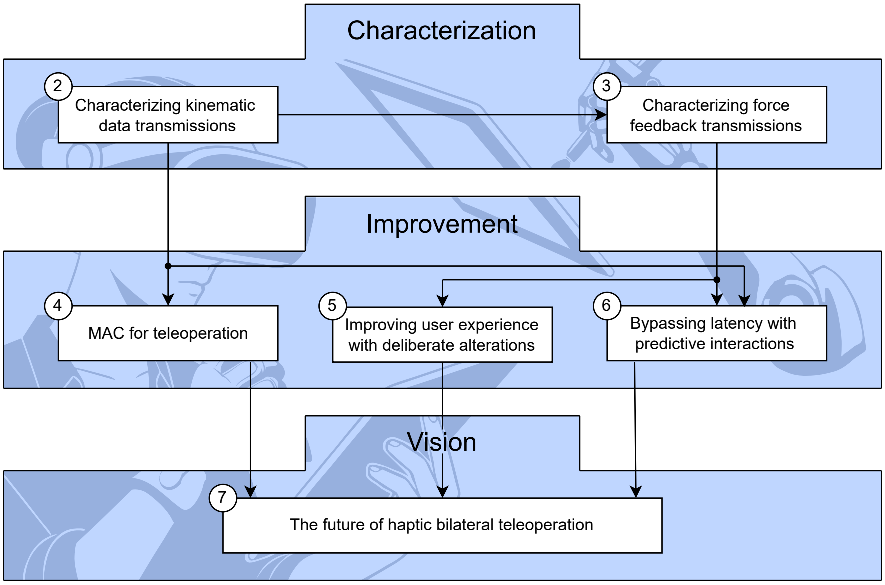

Characterization
2
Characterizing kinematic data transmissions
3
Characterizing force feedback transmissions
Improvement
4
MAC for teleoperation
5
Improving user experience with deliberate alterations
6
Bypassing latency with predictive interactions
Vision
7
The future of haptic bilateral teleoperation

Figure 1.5: A visual representation of the thesis outline. The numbers correspond to the chapters in the thesis.

Sub-Question 3: How can we use insights from characterizing network performance to improve networks design for Haptic Bilateral Teleropation?

Finally, we consider changes to the application itself, aiming to enhance the user experience despite the presence of a low-performing network.
Sub-Question 4: How can we leverage knowledge about human perception to improve the user experience?
Sub-Question 5: How can we relax the delay requirement with alternative feedback mechanisms?

1.5. Contributions and Outline

In this thesis we start from characterizations and improvements to the network, to finding ways of reducing the burden on the network instead. The ultimate goal is to come up with a feasible approach that will enable long-distance HBT in the near future. The structure of the thesis is illustrated in Figure 1.5.

Chapter 2 - Characterizing kinematic data transmissions. In this chapter, we present a novel metric to assess the quality of kinematic data streams. Traditionally, when assessing the performance of a network transmitting such data streams, one considers either network performance indicators, such as latency and packet loss, or directly examines the differences between the input and output data. Methods that directly compare input and output data have the advantage of characterizing the effects of communication strategies that depend on the measured data. However, these methods have difficulty making accurate comparisons when latency is present. The proposed metric distinguishes between variations caused by differences in value and those resulting from differences in time. For instance, such variations may arise from packet loss or network latency, respectively. This results in a more accurate representation of the network's effect on the system, particularly because latency and noise affect the system very differently [5, 6].

Chapter 3 - Characterizing force feedback transmissions. In this chapter we consider the interactions between the operator's actions, the robot's response, and the resulting force feedback experienced by the operator. The key consideration here is the presence of a feedback loop. The timing and manner in which force feedback is relayed to the operator influence the operator's actions, creating a trajectory that, in turn, affects the force feedback. This chapter illustrates how it is not the network latency that is directly experienced by the operator, but instead the effects of latency on the force feedback, which can increase dramatically as a consequence. The chapter presents a novel method of determining the required network performance to facilitate a given teleoperation application.

Chapter 4 - MAC for teleoperation. In this chapter, we design a MAC protocol to optimize the user experience for teleoperation. The proposed approach is based on the findings in Chapter 2, which show that kinematic data is highly resilient to data loss, but has a stringent latency requirement. This contrasts sharply with video traffic, which, due to its encoding, has low resilience to data loss, but less stringent latency requirements. Furthermore, kinematic data is small in size and highly frequent, in contrast to the large less frequent video packets. We propose ViTals, a novel MAC protocol that facilitates the simultaneous transmission of all data streams required for teleoperation, optimizing the quality of each stream based on their individual requirements.

Chapter 5 - Improving User Experience with Deliberate Alterations. In this chapter, we focus on user perception. The key concept is that certain alterations are highly perceivable to humans, while others are nearly unnoticeable. The goal of this work is to identify network-induced alterations that are highly perceivable and mask their impact by introducing deliberate alterations that are nearly unnoticeable. We propose the Adaptive Offset Framework to leverage gaps in human perception and improve the user experience.

Chapter 6 - Bypassing Latency with Predictive Interactions. In this chapter, we adopt a different approach to circumvent the negative effects of latency in direct communication of force and video measurements. By constructing a digital model of the remote environment and running it in a local physics simulation, we can calculate predictions of instantaneous visual and force feedback. This approach, known as Model Mediated Teleoperation, provides significant improvements in latency requirements, but introduces a different set of challenges that scale with the dynamics and complexity of the application. In this chapter, we focus on enhancing the system's ability to handle dynamic environments.

Chapter 7 - The Future of Haptic Bilateral Teleoperation. In the final chapter, the works are summarized, and an outlook is formulated for the future. A combination of low-latency networks, predictive force feedback, and live video is identified as key enablers of scalable HBT systems. Key remaining challenges are acknowledged, but the chapter concludes with an optimistic outlook on the future of HBT and its impending realization.

2

Characterizing Kinematic Data Transmissions[1]

2.1. Introduction

In this chapter, we address Sub-Question 1 as stated in Section 1.4: *How can network performance be characterized for transmitting kinematic data under low-latency requirements?* Reliable network performance assessment is crucial for Haptic Bilateral Teleoperation (HBT) applications but is complicated by the diversity of transmitted modalities and their requirements. In this chapter, we contribute to this question by introducing a novel metric for evaluating the kinematic data stream.

Our analysis builds on the following system design. An operator uses a haptic device to transmit kinematic data (position and orientation) over a network to a remote domain, where a robot arm replicates these movements. Simultaneously, modalities such as audio-visual and force feedback from the remote domain are sent back to the operator, enabling task execution as if the operator is physically present in the remote environment. A schematic of this setup is shown in Figure 2.1.

Despite focusing on the kinematic modality, all modalities are interdependent; for instance, a temporary drop in kinematic updates can halt the robotic device movement, which makes the video feed stagnant. Adding to the complexity is the presence of a human operator within the loop, introducing elements of subjective experience and unpredictable responses from remote domains. These dynamics complicate the task of evaluating the network's performance in facilitating HBT. As we will see, there are several intricacies to consider when characterizing the performance of a network that facilitates the kinematic modality. We elaborate on two important metrics for characterization: Quality of Service (QoS) and Quality of Experience (QoE).

QoS metrics are based on standard network performance indicators such as delay,

[1] This chapter is based on the publication titled *"Setting the Yardstick: A Quantitative Metric for Effectively Measuring Tactile Internet"* and its extension *"ETVO: Effectively Measuring Tactile Internet With Experimental Validation"* [5, 6].

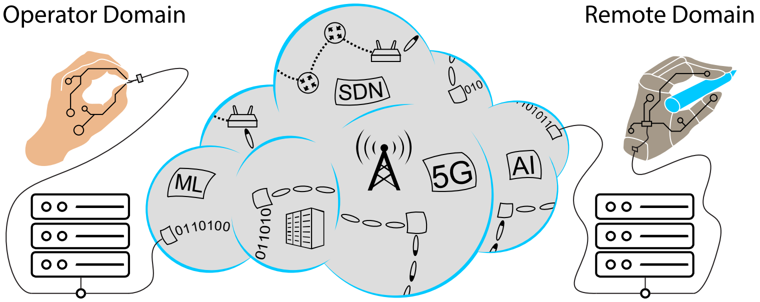


Figure 2.1: A schematic representation of Tactile Internet (TI) illustrating the interplay between the operator and remote domains.

throughput, jitter, and reliability. In the context of Tactile Internet (TI), particular emphasis is placed on end-to-end delay and reliability, with commonly stated targets of 1 ms latency and 99.999% reliability [1, 7, 8]. However, these metrics primarily reflect network dynamics and not the relationship between the network performance and the operator's experience. This limitation highlights the need for more nuanced, signal-aware approaches to characterize TI systems effectively.

QoE metrics adopt a human-centric approach to evaluate user experience. The ideal QoE metric should estimate user experience objectively, without relying on extensive user studies involving participants grading their interaction through a TI system. However, existing approaches cannot distinguish between degradation (offset) in time and value (amplitude) domains. For instance, metrics based solely on Root Mean Square Error (RMSE) struggle to account for time misalignments in signals, which are expected in TI systems due to network latency. These limitations highlight the need for more robust and fine-grained frameworks to characterize the performance of TI systems accurately. Addressing these challenges forms our primary motivation.

Our approach is to devise a method capable of extracting fine-grained time- and value-offset between the sensed and reconstructed signals in a TI session. This method can be applied to any end-to-end TI system (starting from sensors on one end to the actuators on the other end) in a manner that is agnostic to the underlying network. We take DTW as the starting point since it is the widely used tool for determining sample-wise similarity between the two-time sequences; we show its limitations in the context of TI and build on it.

Contributions

The contributions in this chapter are listed below.

① We present a detailed analysis of characterizing TI sessions using DTW and identify areas of improvement for the stated task (Section 2.3).

② We introduce a concrete mathematical framework, called *Effective Time- and Value-Offset (ETVO)*, which extracts fine-grained time and value-offset between sensed and reconstructed signals of a TI system. This framework comprehensively characterizes TI session performance in a system-agnostic manner and represents the first work of

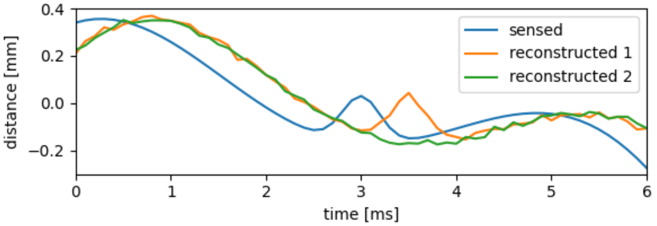


Figure 2.2: Illustration of the problem of RMSE when signals have time-offset. Two possible reconstruction signals – 'reconstructed 1' and 'reconstructed 2' – are shown along with the sensed signal. While the shape of 'reconstructed 1' is identical to the sensed signal, it is delayed. On the other hand, 'reconstructed 2' misses the peak completely. However, the RMSE of 'reconstructed 1' is higher than that of 'reconstructed 2' due to insensitivity to time-offset.

its kind (Section 2.4).

③ We propose two novel metrics: average effective time-offset (T_{ETVO}) and average effective value-offset (E_{ETVO}). These metrics enable the comparison of performance across different TI solutions.

④ We demonstrate the effectiveness of ETVO and its improvement over DTW through objective analysis conducted on a realistic TI setup (Section 2.6).

⑤ We validate ETVO by conducting subjective experiments on a realistic TI setup under a wide range of network conditions. The results show that the proposed metrics correlate strongly with user grades (Section 2.6.2).

⑥ We theoretically derive the expected average delay of TI sessions and confirm that it aligns with T_{ETVO} measurements (Section 2.6.2).

⑦ We demonstrate the insufficiency of both QoS and QoE methods in characterizing TI sessions and identify specific areas where they fail to provide accurate insights (Section 2.6.2).

2.2. Related Work

In this section, we provide an overview of the QoS and QoE metrics developed for evaluating TI systems. Additionally, we examine generic similarity metrics that are pertinent to our proposed metric. The related work discussed here are relevant for other chapters that will reference this section, and is therefore presented in greater detail.

Quality of Service (QoS)

Several modular designs of TI systems use traditional QoS metrics, such as delay, jitter, packet loss, and throughput, for characterizing TI performance. For instance, *Admux*, an adaptive multiplexer for TI proposed by Eid et al. [9], utilizes all of these metrics, whereas the multiplexing scheme by Cizmeci et al. focuses on throughput and delay [10]. Hinterseer et al. [11] proposed a haptic codec that reduces application throughput by transmitting only the perceptually significant samples. Similarly, the congestion control scheme by Gokhale et al. [12] targets delay and jitter to ensure they remain within permissible QoS limits. Several works [2, 3, 4] have leveraged advancements in 5G networks to address the stringent URLLC requirements of TI, providing a comprehensive discussion on their vision and progress in this direction.

While QoS metrics serve as indicators of network performance, they are fundamentally signal-agnostic do not consider the correlation between network dynamics and the operator's experience. This disconnect is particularly evident in TI applications, where characterizing the effects of realistic network conditions requires understanding their influence on signal reconstruction and the resulting user experience. For example, the *Perceptual Deadband (PD)* protocol [13, 14] demonstrates the potential of signal-aware approaches. By exploiting the limits of human perception, PD reduces data rates in a way that is tightly coupled to the TI application, thereby overcoming the limitations of purely QoS-based methods. Unlike teleconferencing applications, which primarily rely on QoS metrics, TI applications demand deeper insights into how network performance and signal properties interact.

Quality of Experience (QoE)

Subjective QoE metrics evaluate teleoperation quality by involving human participants, typically 15–20, who grade their experience. Examples include Basdogan et al. [15] and Yuan et al. [16]. While effective, this method is resource-intensive, driving the need for objective metrics that estimate teleoperation quality without extensive user studies.

Several objective QoE metrics have been proposed. Hinterseer et al. [13, 14] introduced the Perceptual Deadband (PD) scheme, which leverages the logarithmic relationship between human perception and haptic stimuli, validated using the Peak Signal-to-Noise Ratio (PSNR) of reconstructed haptic signals. Sakr et al. [17] extended this with the Haptic Perceptually Weighted Peak Signal-to-Noise Ratio (HPW-PSNR), while Chaudhuri et al. [18] proposed the Perceptual Mean Square Error (PMSE), mapping MSE to human perception.

Hassen et al. [19] introduced the Haptic Structure SIMilarity (HSSIM) index to improve objective estimates of human perception by measuring the similarity between original and reconstructed haptic signals. However, these metrics often rely on Root Mean Square Error (RMSE), which struggles with time-domain offsets. Delays, packet losses, and jitter are common in TI systems, directing to mismatches between sensed and reconstructed signals.

Generic similarity metrics

Determining the similarity between two signals is a classical signal processing problem and has been extensively researched due to its numerous applications, such as speech and gesture recognition [20]. In this section, we discuss some of the techniques devised for this purpose and examine their applicability in extracting time and value-offset, which are crucial for TI applications. Cross-correlation computes the time-offset between the two signals that maximizes their dot product [21]. It is well known that shared networks usually manifest highly non-deterministic and time-varying characteristics. Hence, a constant delay is an incorrect choice for representing the entire TI characteristics. Another popular method known as Dynamic Time Warping (DTW) exists for signals encountering a time-varying delay [22]. DTW conducts an exhaustive search to achieve sample-wise matching between the two signals in a manner that minimizes the cumulative Euclidean distance. It provides an extremely useful construct in determining *how similar two signals* *are*. DTW functions as a practical starting point because it can already compare signals that differ in time. DTW is designed to find the similarity in sequences, for example, that two spoken words are the same, even when spoken at different speed and/or pitch. On the contrary, in teleoperation, the sensed and the reproduced signals are expected to be broadly similar. Hence, our problem is to find out *how two similar signals are different*. While DTW completely solves its intended purpose, it is not designed for the stated objective of characterizing TI systems. Hence, we take DTW as the starting point in this work and perform substantial modifications to serve our purpose.

Several follow-up works on DTW exist, with each of them attempting to outperform DTW in one or more aspects. The most widely recognized ones include Edit Distance on Real sequences (EDR) [23], Edit distance with Real Penalty (ERP) [24], and Longest Common Sub-Sequence (LCSS) [25]. However, they manifest the inherent characteristics of DTW and hence are unsuitable for TI, as will be detailed in the following sections. In the next section, we provide the necessary background of DTW as it forms the basis of the ETVO design.

2.3. DTW: Background and analysis

DTW measures the similarity between two sequences encountering time-varying delay [22] and is extremely useful for sequence classification problems like correlation power analysis, DNA classification, and notably, speech recognition. DTW provides a distance score based on the l^2-norm and is therefore similar to RMSE. An important observation is that the unit of DTW's outcome is not time. Therefore, the score does not represent a delay. This is not a concern in applications where DTW is used. Its typical use is the identification of two time-series being similar. For example, DTW can be used to identify a spoken word to match with a word in an existing library, even if the word is spoken at a different pitch or speed. In these scenarios, RMSE would report a large error, while DTW's ability to warp the time-series would produce a significantly lower score, indicating their similarity. DTW is a valuable starting point for us because it has structures in place that allow for a sample-wise comparison between time-series, but it does not produce an indication of delay.

2.3.1. Mathematical Representation

DTW constructs a *warp path* that indicates a sample-wise mapping between two time-series that minimizes their cumulative l^2-norm. Given $\tilde{\boldsymbol{f}}, \tilde{\boldsymbol{g}} \subset \mathbb{R}^N$ as two N-length discrete time-series, let $\tilde{\boldsymbol{W}}$ denote the set of all possible warp paths between $\tilde{\boldsymbol{f}}$ and $\tilde{\boldsymbol{g}}$. Let the $(k+1)$-th point of a warp path be denoted as $\tilde{\boldsymbol{w}}(k) = (\tilde{\boldsymbol{w}}_0(k), \tilde{\boldsymbol{w}}_1(k)) \in \tilde{\boldsymbol{W}}$, where $\tilde{\boldsymbol{w}}_0, \tilde{\boldsymbol{w}}_1 \subset \mathbb{N}^K$ and $K \in [N, 2N-1]$. For example, the warp path in Figure 2.3 is given as [(0,0), (1,0), (2,0), (3,0), (4,1), (5,2), ...]. Essentially, $\tilde{\boldsymbol{w}}_0$ and $\tilde{\boldsymbol{w}}_1$ return the indices of $\tilde{\boldsymbol{f}}$ and $\tilde{\boldsymbol{g}}$, respectively.

The entries in $\tilde{\boldsymbol{w}} \in \tilde{\boldsymbol{W}}$ must meet the following conditions:

1. Monotonicity and continuity:

$$\begin{aligned}\tilde{\boldsymbol{w}}_0(k) &\le \tilde{\boldsymbol{w}}_0(k+1) \le \tilde{\boldsymbol{w}}_0(k)+1,\\ \tilde{\boldsymbol{w}}_1(k) &\le \tilde{\boldsymbol{w}}_1(k+1) \le \tilde{\boldsymbol{w}}_1(k)+1.\end{aligned}$$

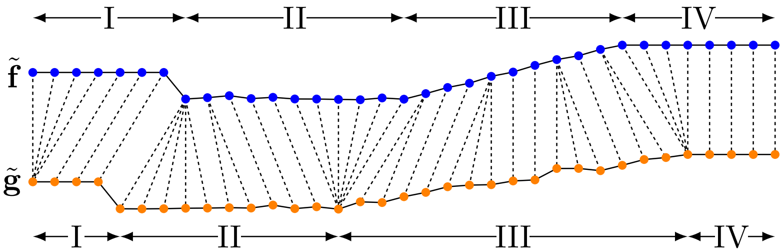


Figure 2.3: Example of sample-wise alignment between signals $\tilde{\boldsymbol{f}}$ and $\tilde{\boldsymbol{g}}$ as per DTW. The dashed lines indicate the mapping between the samples.

2. Boundary:

$$\tilde{\boldsymbol{w}}(0) = (0,0), \tilde{\boldsymbol{w}}(K-1) = (N-1, N-1). \tag{2.1}$$

The effect of these conditions is that subsequent samples are always put after their predecessors. DTW chooses the warp path that gives the minimum error (l^2-norm) between $\tilde{\boldsymbol{f}}$ and $\tilde{\boldsymbol{g}}$ [26]. Hence, we get the error computed by DTW as

$$\mathrm{DTW}(\tilde{\boldsymbol{f}}, \tilde{\boldsymbol{g}}) = \min_{\tilde{\boldsymbol{w}} \in \tilde{\boldsymbol{W}}} \sum_{k=0}^{K-1} \tilde{\delta}(\tilde{\boldsymbol{w}}(k)), \tag{2.2}$$

where $\tilde{\delta}$ is the distance, between two samples. In this case $\tilde{\delta}(\tilde{\boldsymbol{w}}(k)) = (\tilde{\boldsymbol{f}}(\tilde{\boldsymbol{w}}_0(k)) - \tilde{\boldsymbol{g}}(\tilde{\boldsymbol{w}}_1(k))^2$. The computation of $\mathrm{DTW}(\tilde{\boldsymbol{f}}, \tilde{\boldsymbol{g}})$ is carried out as follows:

1. Populate a cost matrix $\tilde{\boldsymbol{C}} \subset \mathbb{R}^{N \times N}$. Every point in this matrix gives a value indicating the cheapest path to that point from the start. Every element is given by,

$$\tilde{\boldsymbol{C}}[i,j] = \tilde{\boldsymbol{\delta}}(i,j) + \min(\tilde{\boldsymbol{C}}[i,j-1], \tilde{\boldsymbol{C}}[i-1,j-1], \tilde{\boldsymbol{C}}[i-1,j])$$

2. Backtrack from $\tilde{\boldsymbol{C}}(N-1, N-1)$ to $\tilde{\boldsymbol{C}}(0,0)$ to construct the warp path $\tilde{\boldsymbol{w}}$.

The time complexity of DTW is $O(N^2)$, although several algorithms for speeding up the computations exist [20, 27].

2.3.2. Challenges in applying DTW to TI

In the context of TI, $\tilde{\boldsymbol{f}}$ and $\tilde{\boldsymbol{g}}$ represent the sensed and reconstructed signals, respectively.

Boundary conditions cause unrealistic artifacts

The boundary conditions in Equation (2.1) ensure that the extreme ends of the sequences are invariably aligned with each other. As a consequence, the delay is forced to be zero at the extreme ends. Segments ’I’ and ’IV’ in Figure 2.3 illustrate this. For TI applications, any non-zero delay systems will have a significant mismatch at the endpoints. This can be particularly significant when analyzing small sequences.

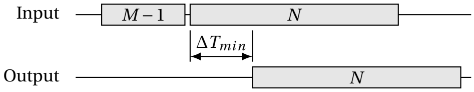


Figure 2.4: Illustration of extending the input sequence by $M-1$ samples in ETVO to avoid the start and end artifacts of DTW.

Unconstrained delay adjustments

The *warp path* produced by DTW, can be considered a representation of sample-wise delay but is generally not significant outside the algorithm. In practice, the warp path can be unrealistically erratic, with high-frequency oscillations not originating from the TI system's behavior. For applications like speech recognition, high-frequency components in the warp path are of no consequence.

We intend to use the warp path as the estimated delay of a TI system, and for this purpose, both average delay and variations in delay are essential. Segments 'II' and 'III' in Figure 2.3 provide examples of multiple shifts in delay that are disproportional to the compared signals. When observing the warp path, a TI system can appear to have a high variation in delay, irrespective of the actual variation.

DTW prefers to change the delay when the velocity is as small as possible because that lowers the l^2-norm. This can cause the observed change in delay to be out of sync with the actual change in delay. Multiple examples can be found in Figure 2.3. Segments 'I' and 'IV' start with an adjustment of delay. Despite that, the changes happen toward the end of the corresponding segments. At the start of Segment 'II', there is a considerable delay change in a few samples before a small peak that causes the change.

In order to resolve the above issues and design suitable performance metrics for TI, we perform substantial refinements to DTW, as described in the next section.

2.4. Design of TI Measurement Framework

In this section, we present the mathematical foundation of the proposed framework for the characterization of TI sessions – *Effective Time- and Value-Offset (ETVO)*. Using this framework, we introduce two metrics: *Effective Time-Offset* (ETO) and *Effective Value-Offset* (EVO) to indicate the time- and value-offset, respectively, between the sensed and reconstructed signals. We use *effective* to indicate that the values show how the system appears to behave when considering it as a black box. For example, if a prediction method is used to make it seem like the signal is advanced by 2 ms, ETVO should conclude that the delay is 2 ms less. Note that the unit of the value-offset matches the unit of the analyzed signals, which can be position, velocity, force, and temperature, among others.

2.4.1. Proposed ETVO framework

We now discuss our refinements for resolving the previously discussed issues of DTW for TI applications through the design of the ETVO framework.

Relaxation of boundary conditions

We first address the boundary conditions described in Section 2.3.2 by adjusting the mathematical structure. Let $\boldsymbol{f}$ and $\boldsymbol{g}$ denote slices of the sensed and the reconstructed

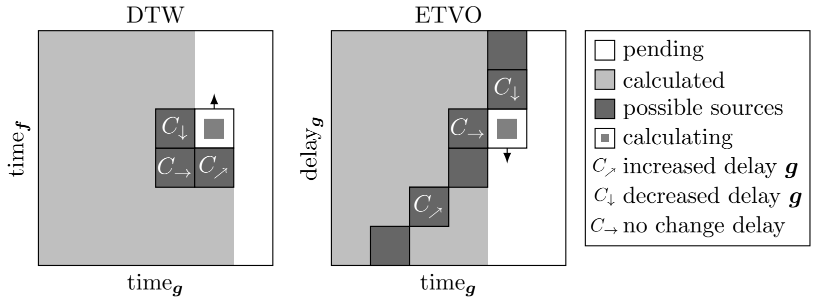


Figure 2.5: Illustration of the population of $\boldsymbol{C}$ in both DTW and ETVO. Different types of changes in delay, indicated as $C_\rightarrow$, $C_\downarrow$, $C_\nearrow$ are present in both the DTW and ETVO table to show their correspondence. A key difference between DTW and ETVO is that the latter also calculates multiple steps, increasing the possible sources as indicated with dark gray squares.

signals, respectively. For ease of explanation, we use the same notations as in DTW but remove the accent (˜) to denote the ETVO counterparts. We define the range of possible time offsets as a fixed number. For TI systems, this is desirable because the range of expected time offsets is primarily caused by the network and not the session length. The minimum time offset is $\Delta T_{\text{min}} \in \mathbb{R}$ and the maximum time offset is $\Delta T_{\text{max}} \equiv \Delta T_{\text{min}} + MT$, where $M \subset \mathbb{N}^+$ and T is the sampling period. Given N as the length of $\boldsymbol{g}$, $\boldsymbol{f}$ should be of length $N + M - 1$ to ensure a range of M time offsets. If the first sample of $\boldsymbol{g}$ is located at $t = 0$, then the first sample of $\boldsymbol{f}[k]$ should be located at $t = -\Delta T_{\text{min}} - (M-1)T$. This is illustrated in Figure 2.4.

With the new structure, we redefine the warp path to be used as a representation of sample-wise delay. Let $\boldsymbol{W} \subset \mathbb{N}^N$ denote the power set of possible warp paths to align $\boldsymbol{g}$ onto $\boldsymbol{f}$. The optimal warp path is denoted as $\boldsymbol{w} \in \boldsymbol{W}$, where $\boldsymbol{w}[k]$ indicates that $g[k]$ corresponds to $f[k - w[k]]$. We denote ETO as the sample-wise time offset corresponding to the alignment between $\boldsymbol{f}$ and $\boldsymbol{g}$ and is expressed as

$$\text{ETO}[k] = \Delta T_{\text{min}} + w[k]. \tag{2.3}$$

We define the associated cost matrix as $\boldsymbol{C} \subset \mathbb{R}^{N \times M}$, where the x-axis indicates the sample index of $\boldsymbol{g}[k]$, and the y-axis is corresponding to time-offset. Figure 2.5 illustrates this concept, wherein the value at each entry of $\boldsymbol{C}$ indicates the cumulative cost of getting to that point. Specifically, the cost indicates l^2-norm of the most efficient warp path from the start of $\boldsymbol{g}$ to the current point. The propagation through $\boldsymbol{C}$ is

$$\boldsymbol{C}[i,j] = \delta(i,j) + \min(\boldsymbol{C}[i-1,j], \boldsymbol{C}[i-1,j-1], \boldsymbol{C}[i,j+1]),$$

where $\delta(i,j) \equiv (\boldsymbol{g}[i] - \boldsymbol{f}[i-j+M-1])^2$. The three directions for calculating $\boldsymbol{C}$ correspond directly to the three directions in DTW as defined in Equation (2.3). These new directions are indicated with $C_\nearrow$, $C_\downarrow$, and $C_\rightarrow$ indicating an increase, decrease, and no change in delay, respectively. An illustration of the resulting system and how the directions correlate between ETVO and DTW is shown in Figure 2.5. For this translated system, the monotonicity and continuity condition is given as $0 \leq \boldsymbol{w}(k+1) \leq \boldsymbol{w}(k) + 1$. For DTW, swapping $\boldsymbol{f}$ and $\boldsymbol{g}$ leads to the same result. However, for the ETVO structure, the order of the signals is important. $\boldsymbol{g}$ projected onto $\boldsymbol{f}$ and $\boldsymbol{f}$ projected onto $\boldsymbol{g}$ would yield completely different results.

An important effect of these changes is that it removes the boundary conditions enforcing the first and last sample of $\boldsymbol{f}$ and $\boldsymbol{g}$ to pair up. As a result, our framework now has the option to report non-zero delays for every sample in $\boldsymbol{g}$. The first column of $\boldsymbol{C}$ is initialized as $\boldsymbol{C}(0,*) = [0]^M$. Every starting delay is assigned a zero cost. To remove the ending artefact, we let the last sample of ETO be chosen as the cheapest option, so that

$$\boldsymbol{C}(N-1, \boldsymbol{w}[N-1]) \leq \boldsymbol{C}(N-1, j), \qquad\qquad \forall j \in [0, M-1].$$

As a consequence, not every sample of $\boldsymbol{f}$ has to be assigned a sample in $\boldsymbol{g}$. Therefore the DTW boundary condition given in Equation (2.1) is discarded for samples in $\boldsymbol{f}$.

Constraining delay adjustments

In order to mitigate the issue of unconstrained delay adjustments in DTW (described in Section 2.3.2), we come up with substantial refinements to its design. For DTW, the warp path is designed as an intermediary, but for ETVO, we use the warp path as an indicator of the time-varying delay. First, let us define what a delay adjustment is in the context of ETVO. It is the change in estimated delay per unit time. $C_{\downarrow}$ and $C_{\nearrow}$ represent an increase and decrease in delay, respectively. A change in delay does not have to be of magnitude one but can be any positive integer. The dark gray squares in Figure 2.5 indicate this.

In order to address the unconstrained delay adjustments, penalties are introduced to suppress adjustments that result in relatively minor improvements. We describe how multiple penalties are needed that target several aspects to achieve the intended result. We present the mathematical foundation behind the cost matrix $\boldsymbol{C}$ and describe the rationale behind the penalties.

$$\begin{aligned}
C_{\rightarrow}[i,j] &= \boldsymbol{C}[i-1,j], \\
C_{\downarrow}[i,j] &= \min_{k \subset \mathbb{N}^+} \Big(\boldsymbol{C}[i,j+k] + \sum_{l=1}^{k-1} \delta(i,j+l) + kP_{\text{prop}} + P_{\text{fixed}}\Big), \\
C_{\nearrow}[i,j] &= \min_{k \subset \mathbb{N}^+} \Big(\boldsymbol{C}[i-k,j-k] + \sum_{l=1}^{k-1} \delta(i-l,j-l) + kP_{\text{prop}} + P_{\text{fixed}}\Big).
\end{aligned} \tag{2.4}$$

For every delay adjustment, we introduce two variables – P_{fixed} and P_{prop}. These correspond to a fixed penalty for every delay adjustment and a penalty proportional to the size of the delay adjustment, respectively. P_{fixed} suppresses the number of delay adjustments, and P_{prop} affects the magnitude of each adjustment. Together, these penalties suppress the delay adjustments estimated by the algorithm. The variable P_{prop} balances between time and value-offsets. High penalties reduce the time-offsets and increase the value-offsets. ETVO performance approaches DTW when the penalties tend to zero. P_{fixed} and P_{prop} both reduce changes in time-offset at the expense of more value-offset, but with slightly different effects. P_{prop} has a larger effect on the size of adjustments, while P_{fixed} has a larger effect on the frequency of adjustments. The best candidate for each direction is calculated as shown in Equation (2.4) and is illustrated in Figure 2.5.

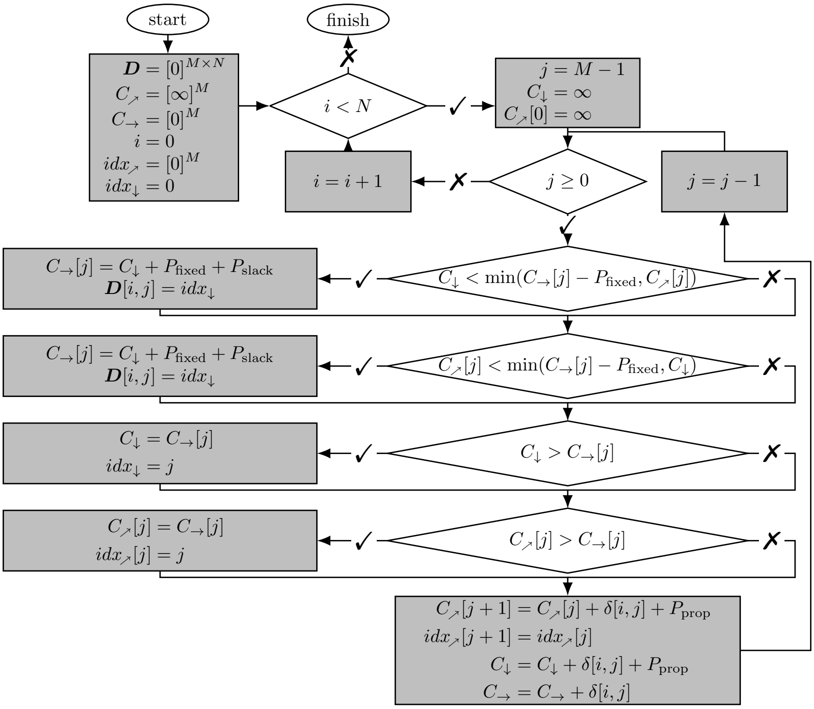

start
finish
$\boldsymbol{D} = [0]^{M\times N}$
$C_{\nearrow} = [\infty]^M$
$C_{\rightarrow} = [0]^M$
$i = 0$
$idx_{\nearrow} = [0]^M$
$idx_{\downarrow} = 0$
$i < N$
✓
✗
$j = M - 1$
$C_{\downarrow} = \infty$
$C_{\nearrow}[0] = \infty$
$i = i + 1$
$j \geq 0$
$j = j - 1$
$C_{\rightarrow}[j] = C_{\downarrow} + P_{\text{fixed}} + P_{\text{slack}}$
$\boldsymbol{D}[i,j] = idx_{\downarrow}$
$C_{\downarrow} < \min(C_{\rightarrow}[j] - P_{\text{fixed}}, C_{\nearrow}[j])$
$C_{\rightarrow}[j] = C_{\downarrow} + P_{\text{fixed}} + P_{\text{slack}}$
$\boldsymbol{D}[i,j] = idx_{\downarrow}$
$C_{\nearrow}[j] < \min(C_{\rightarrow}[j] - P_{\text{fixed}}, C_{\downarrow})$
$C_{\downarrow} = C_{\rightarrow}[j]$
$idx_{\downarrow} = j$
$C_{\downarrow} > C_{\rightarrow}[j]$
$C_{\nearrow}[j] = C_{\rightarrow}[j]$
$idx_{\nearrow}[j] = j$
$C_{\nearrow}[j] > C_{\rightarrow}[j]$
$C_{\nearrow}[j+1] = C_{\nearrow}[j] + \delta[i,j] + P_{\text{prop}}$
$idx_{\nearrow}[j+1] = idx_{\nearrow}[j]$
$C_{\downarrow} = C_{\downarrow} + \delta[i,j] + P_{\text{prop}}$
$C_{\rightarrow} = C_{\rightarrow} + \delta[i,j]$

Figure 2.6: Flowchart for finding the optimal way of traversing the delay, given the constraints specified for ETO.

In the case of DTW, the delay adjustments do not have to align with the actual events that trigger the delay changes. It is beneficial for the algorithm to make changes when there is the least amount of velocity. The reason is that when the delay is adjusted, some samples are counted multiple times, and their contribution is less when the velocity is closer to zero. However, this tendency has little to do with when a change in delay actually occurs. For TI, such behavior makes analysis hard and makes the session quality estimation inaccurate. ETO should not be influenced by an event that occurs in the future. Note that P_{fixed} and P_{prop} do not address this issue of timing the delay adjustments. Therefore, we propose to introduce slack in delay adjustments where their timing is postponed until the slack penalty P_{slack} is breached. P_{slack} acts on top of P_{fixed} and P_{prop} for every delay adjustment, but is only added after an adjustment is made. The addition of P_{slack} increases the likelihood that the delay adjustments match the events that cause them. With this, the overall cost matrix $\boldsymbol{C}$ is given as follows.

$$\begin{aligned}\boldsymbol{C}[i,j] = \delta(i,j) + \min(&C_{\rightarrow}[i,j], C_{\downarrow}[i,j], C_{\nearrow}[i,j]) \\ &+ P_{\text{slack}} \quad \text{if } C_{\rightarrow}[i,j] > \min(C_{\downarrow}[i,j], C_{\nearrow}[i,j])\end{aligned}$$

Defining EVO

Unlike DTW, where the residual distance for every sample in the *warp path* is aggregated into a single number similar to RMSE, we represent the value-offset as a time series that

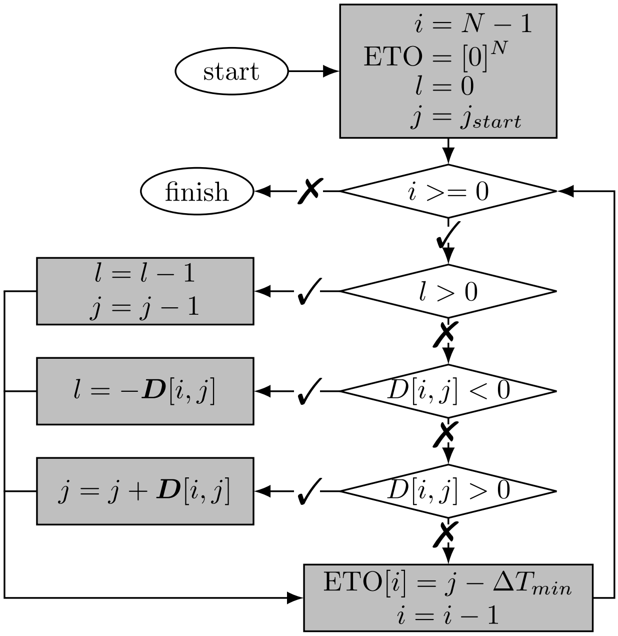


Figure 2.7: Flowchart of the backtracking algorithm used to extract the ETO from direction matrix $\boldsymbol{D}$.

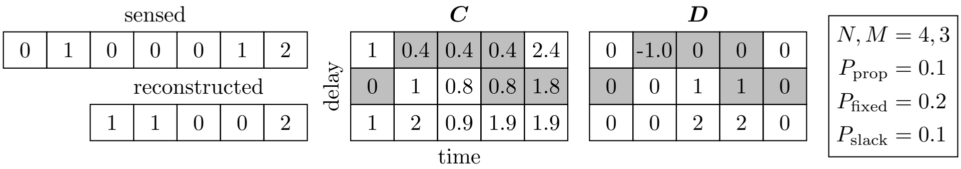
we call *Effective value-offset* (EVO). Every sample of EVO indicates the error computed by l^2-norm from all samples of $\boldsymbol{g}$ compared to the corresponding sample in $\boldsymbol{f}$, excluding the penalties. When ETO increases or stays the same, only one sample of $\boldsymbol{g}$ is compared to $\boldsymbol{f}$. However, when ETO decreases, the EVO value for that sample is the l^2-norm between the output sample and several input samples. This enables obtaining fine-grained information on how samples contribute to the value-offset. The mathematical description of EVO is given by

$$\text{EVO}[i] \quad = \quad \begin{cases} \sum_{l=\text{ETO}[k+1]}^{\text{ETO}[k]} \delta(i,l) & \text{if ETO}[i] > \text{ETO}[i+1], \\ \delta(i, \text{ETO}[i]) & \text{otherwise.} \end{cases}$$

Due to this, there are spikes in EVO every time the ETO reduces by a large amount.

COMPUTATIONAL COMPLEXITY

Besides presenting the ETVO framework, we also provide an efficient way of calculating ETO and EVO. The addition of P_{fixed} results in a larger set of values to consider when finding the optimal path. Instead of the three adjacent locations, one has to consider a total of M entries. Besides considering multiple entries, when backtracking to retrieve the delay, one must consider the number of steps taken. To store that information, we propose a direction matrix $\boldsymbol{D} \subset \mathbb{Z}^{M \times N}$. The number stored in $\boldsymbol{D}(k, i)$ indicates that the next point is at $i + \boldsymbol{D}(k, i)$. The resulting algorithm for populating $\boldsymbol{D}$ is illustrated with a flow chart in Figure 2.6.


sensed
reconstructed
delay
time
C
D

sensed

0	1	0	0	0	1	2

reconstructed

1	1	0	0	2

$\boldsymbol{C}$

1	0.4	0.4	0.4	2.4
0	1	0.8	0.8	1.8
1	2	0.9	1.9	1.9

$\boldsymbol{D}$

0	-1.0	0	0	0
0	0	1	1	0
0	0	2	2	0

$N, M = 4, 3$
$P_{\text{prop}} = 0.1$
$P_{\text{fixed}} = 0.2$
$P_{\text{slack}} = 0.1$

Figure 2.8: Numerical example of ETVO including the direction matrix. The gray cells indicate the optimal path chosen by ETVO.

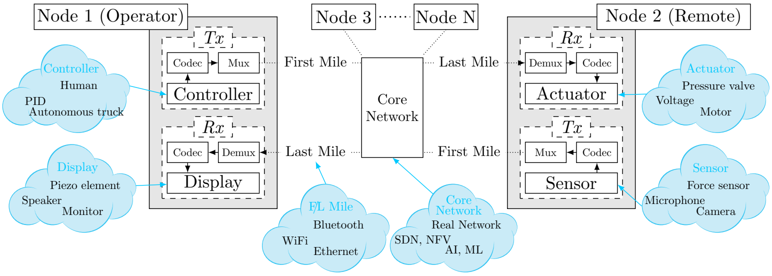


Figure 2.9: A high-level architecture of the proposed TI framework for TI applications depicting fundamental and supplementary components.

The backtracking algorithm is shown in Figure 2.7. The size and complexity of populating $\boldsymbol{D}$ and the backtracking algorithm scale linearly with signal length. The complexity is therefore $\mathcal{O}(N)$. A numerical example of how $\boldsymbol{C}$ and $\boldsymbol{D}$ are populated is provided in Figure 2.8.

2.4.2. Quantitative Metrics for TI

ETVO framework produces two time series – ETO and EVO. While it is crucial to extract fine-grained information about effective offsets for monitoring the performance in real-time and adapting the communication accordingly, it is also important to use them for performance benchmarking and comparing different TI solutions. Long-term averages serve this purpose better than time series. To this end, we propose two quantitative metrics that can be derived from ETO and EVO.

1. T_{ETVO} – the average end-to-end delay of ETO.
2. E_{ETVO} – the average l^2-norm of EVO.

In this work, we use the above metrics for experimental evaluation of the effectiveness of ETVO in measuring TI performance. We intend to use ETO and EVO for TI performance monitoring and real-time adaptation in a future extension.

2.5. TIXT - A Tactile Internet eXtensible Testbed

One of the principal challenges in developing bilateral teleoperation systems is verifying the performance of proposed solutions. We introduce a framework to establish a practical

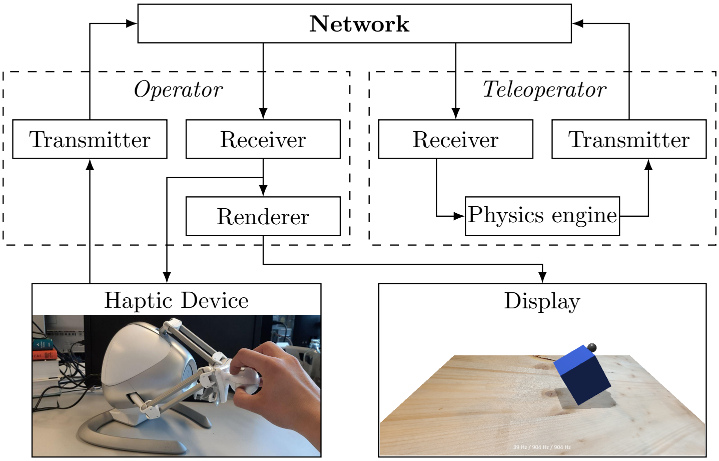


Figure 2.10: A schematic overview of our experimental setup. The operator and teleoperator modules run on different computers that are not collocated. The physics engine resides in the controlled domain, resembling a real TI system using Novint Falcon haptic device.

and accessible TI testbed platform. This platform incorporates a variety of standard TI functionalities designed for immediate, off-the-shelf applications.

The TI framework we propose is distinguished by its notable features:

- Modular and extensible implementation of fundamental components of TI
- Easy modification, deployment, and replication
- Robust to characteristics of peripheral devices such as haptic-audio-video interfaces
- Easy configuration of system parameters.

For efficient communication, one has to add tools for sending signals efficienty. These include, for example, codecs for effective signal encoding and media multiplexers to combine independently sensed information (kinematic, force, audio, and video).

A TI testbed was proposed in [28] and has been utilized to support haptic codec standardization activities [29]. The testbed simulates a TI session by having the human participant interact with a virtual environment (VE) via both haptic and visual feedback. The haptic device provides measurements at 1 kHz, and the VE calculates force feedback at 1 kHz. A visual rendering of VE is produced at 60 Hz. The haptic device used in this setup is a Novint Falcon. Force calculation and visual rendering in the VE are implemented inside of the Chai3D engine. Unfortunately, this testbed lacks the network component. Hence, we perform significant refinements to the testbed in [28] to realize a networked TI testbed.

We propose an architecture that includes a network, as depicted in Figure 2.9. We present the resulting testbed as *TIXT* – a Tactile Internet eXtensible Testbed – as a key step towards developing a comprehensive, generic-purpose TI testbed.

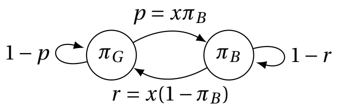


Figure 2.11: Gilbert Elliot model and inclusion of scalar x that allows one to change the distribution between bursty and uniform behavior without affecting the average packet loss. π_G and π_B are the average probability of successful and failed packet transmissions, respectively. p and r are the chance of switching states.

Design of TIXT

We extend the before mentioned testbed by decoupling the testbed into an operator domain module and a remote domain module, each residing on a different workstation connected by a network. Figure 2.10 shows an overview of the entire system. The operator domain module senses the position of the haptic device. The remote domain module houses the simulation of physics aspects. The physics simulation is a substitute for a TI application where the remote domain module would house a real physical environment. The remote domain module receives haptic device data through the feedforward channel and feeds it into the physics environment.

As explained in Section 2.1, a realistic TI application is characterized by kinematic information communicated from the operator to the remote domain and haptic-video information back to the operator domain. Typically, haptic/kinematic and video streams have heterogeneous characteristics and requirements. Video traffic has a much higher bit rate than haptic/kinematic data. On the other hand, video traffic is more tolerant to latency, but highly sensitive to losses (<2%) [30, 31].

For performance evaluation of solutions focusing only on haptic/kinematic data, it is important to minimize the negative impact of video traffic on user perception. This applies to ETVO as it deals with characterizing the offsets between sensed and reconstructed kinematic/haptic signals accurately.

We came up with a simple solution to address the above challenge for virtual environment interactions. Instead of transmitting the video feed from a camera in the remote domain, we send only the kinematic information (position and orientation) of all dynamic objects in the VE along with the computed haptic feedback to the operator domain. The kinematic information is used to update the visual display of VE in the operator domain. Note that this alternative lends itself well to the evaluation of ETVO and is not necessarily meant for usage in real-world TI applications.

We use data generated by our networked testbed to provide examples that demonstrate the efficacy of ETVO on a fine-grained scale. We also add white Gaussian noise to the sensed signals to evaluate ETVO's robustness to channel noise.

Netem, a standard network emulation tool, is used to emulate various network conditions, ensuring strong control over the network performance. This control is desirable, as the main purpose of the experiment is to analyze the performance of ETVO and not the testbed. The workstations at the operator and the remote domains are connected to the university (shared) network and use Ethernet links to connect to a network switch. NetEM is switched on at the operator domain for applying the configured network setting to traffic flowing through it. For the objective evaluation of ETVO, we pick several network settings that help us to illustrate the working of ETVO. We will specify the chosen delay,

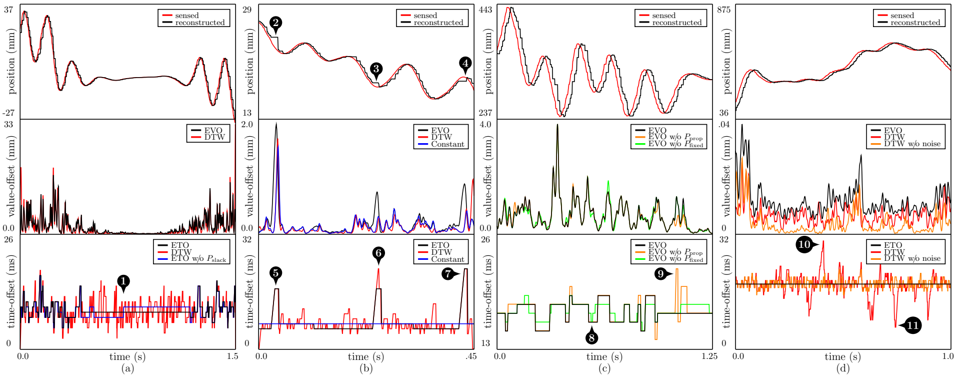


Figure 2.12: Comparison of the performances of DTW and ETVO frameworks using a wide variety of experimental setups showing the effects of (a) P_{slack}, (b) uniform packet losses and perceptual deadband (PD) scheme, (c) P_{prop} and P_{fixed}, and (d) addition of noise to the sensed signal.

jitter, and packet loss settings as we describe our findings in the next section. The bursty packet loss scenario is created using Netem's Gilbert-Elliot model. A bursty loss scalar x is introduced, indicating the correlation between average packet loss π_B and the probability of loss after a successful transmission r. Figure 2.11 shows how x affects the Gilbert Elliot model.

We apply linear extrapolation at the receiver to satisfy the 1 kHz haptic refresh rate. This takes care of the irregular arrival of packets, especially when packet loss or PD is present. The linear extrapolation uses velocity based on sensed position samples in the operator domain, which is included in the packets. This adds redundancy to the system which improves performance in most cases.

2.6. PERFORMANCE ANALYSIS

To evaluate our proposed metrics, we develop a realistic TI testbed where a human user can interact with a remotely rendered virtual environment (VE) over a network. As a starting point for our testbed design, we consider a recently proposed testbed for simulating TI interaction [28].

2.6.1. OBJECTIVE ANALYSIS

The modifications to the basic DTW algorithm proposed in Section 2.4 can be categorized into two groups. The first group deals with transforming the algorithm into an asymmetrical structure without start and end artifacts. The second group concerns the addition of penalties, which are required for improving the fine-grained analysis significantly. To illustrate these different aspects of ETVO, we picked four fragments from the haptic data trace.

We start by gauging the sensitivity of each of the schemes to the signal variations. We set the network delay to 15 ms and jitter to 10 ms. We disable packet loss for this experiment. In Figure 2.12(a), it can be observed that at the extremes of the plot, ETVO shows fluctuations in time-offset estimation, but at areas with minimal changes, the frequency is reduced. This behavior reflects that delay will significantly impact the areas with extremes as opposed to the minimal areas. In contrast, DTW continuously fluctuates irrespective of the context. We also demonstrate the effect of P_{slack} by comparing ETO with and without P_{slack} (labelled as 'ETO w/o slack'). For the version without P_{slack}, it can be seen that the time offset changes in the minimal area (as indicated with ❶). ETO with P_{slack} postpones that decision to a more noticeable moment when the mismatch in delay leads to an observable difference. ETVO and DTW perform similarly in the value domain, despite the significantly higher number of delay adjustments performed by DTW. This example shows how ETVO makes evaluations that are context-aware. Further, note that DTW has a spike in value-offset on both edges because of the start and end artefacts. This behavior can be seen in the other examples as well.

In Figure 2.12(b) there are periods of considerable value-offset due to a combination of bursty packet loss and PD. Network delay and jitter from NetEm are disabled for this particular experiment. We add bursty packet loss with parameters p =5% and r =50% in the Gilbert-Elliott model. Additionally, we employ PD with a velocity deadband of 5%. There are three specific instances (markers ❷ - ❹) where the combined effect of PD and bursty losses lead to a significant error in the reconstructed signal. In this case, DTW relentlessly adjusts the time-offset as the PD and losses are slightly degrading the signal. ETVO Chooses only to act when the effect is significant enough (markers ❺ - ❼). The value-offset is smoothed with a Gaussian distribution for visual clarity.

We now show the distinct effects of P_{prop} and P_{fixed}, and demonstrate the importance of both. We use the same network settings as in Figure 2.12(b). We show the results in Figure 2.12(c), which has arrows with numbers that we will use as markers in this analysis. We consider three different settings for algorithm parameters:

(i) $[P_{\textbf{prop}}, P_{\textbf{fixed}}] = [0.025, 0.05]$ (black curve),
(ii) $[P_{\textbf{prop}}, P_{\textbf{fixed}}] = [0.05, 0]$ (amber curve), and
(iii) $[P_{\textbf{prop}}, P_{\textbf{fixed}}] = [0, 0.1]$ (green curve).

The values are chosen such that the overall strength of each setting is balanced but divided over P_{prop} and P_{fixed} differently to isolate the effect of omitting either of the penalties. Marker ❽ indicates an event where scenario (ii) adjusts in a large number of small steps because there is no extra cost associated with using multiple steps. Marker ❾ indicates an event where scenario (iii) causes a large step change but is limited in the number of steps because there is no extra cost associated with the size of a change. Scenario (i) has a similar performance in the value domain, but a significantly less cluttered ETO.

Figure 2.12(d) shows the effect that high-frequency noise has on DTW and ETVO. For this purpose, we add AWGN to the signal. We disable delay and packet loss for this experiment. Both DTW and EVO are plotted with the noise added, while DTW w/o noise is a version of DTW without the added AWGN. High-frequency noise is a good example of a common way of signal distortion that DTW cannot deal with properly. Note that ETO outperforms the best case DTW, i.e. DTW w/o noise, demonstrating its noise resilience. Further, one can also notice the vulnerability of DTW to even a marginal amount of noise, causing time-offset to fluctuate vigorously.

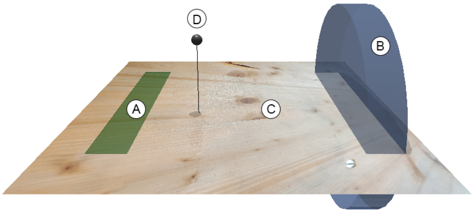


Figure 2.13: A snapshot of *target tracking* game developed for the subjective performance evaluation of ETVO. 'A' is the moving target that needs to be tracked by the slider indicated with 'B'. 'C' is a plane that serves as a rigid floor. 'D' is the cursor that represents the position of the Novint Falcon in the virtual environment. A downward line and a shadow are cast on the plane to help the participant understand the location of 'D' better.

2.6.2. SUBJECTIVE ANALYSIS

Apart from the objective analysis, the networked testbed should provide a platform to facilitate subjective analysis. The setup is designed so that human operators can experience TI sessions and grade them based on subjective experience. We use this setup to demonstrate the efficacy of ETVO qualitatively. There are a few requirements for an experiment that benefit the statistical relevance of the test results. ① The participants should perform the same task multiple times under different settings. ② To maximize the perception, the participants should concentrate. However, participants will have different levels of skill. Hence, the experiment must help the participants concentrate without placing high demands on their skill levels. ③ The task duration should be short and must enforce the operator to interact with the virtual environment continuously to generate haptic feedback. Long tasks can lead to fatigue, especially among older people.

To meet the above requirements, we designed a *target tracking* game that requires the participant to push a slider, labeled *B* in Figure 2.13, left and right. During the test, the target (labeled *A*) moves left and right. A participant has to push the slider to track the target as closely as possible. This task is consistent over multiple iterations, can challenge participants of any skill level, and because the slider has to move continuously, it invites continuous physics interactions. Hence all three of our requirements are met.

NETWORK EMULATION

During the experiments, users experience several instances of the same scenario while subjected to different emulated network settings as described in Section 2.5. To perform an extensive performance evaluation of ETVO, we consider a wide variety of network conditions. We take a set of values ranging from 0 to 16 ms for network delay. Uniform loss (UL) and burst loss (BL) are varied between 20% and 80%. Additionally, we use a set PD between 5% and 15%. We consider these settings in isolation and combinations. For the subjective analysis experiments, $x = 0.25$ was used. The linear extrapolation remains the same as that explained in Section 2.5.

EXPERIMENTAL PROCEDURE

Before the experiment, the participants are informed that the goal is to investigate the effect of perceptual degradation. Each participant gets as much time as they want to familiarize themselves with the application with perfect network conditions, i.e. zero

10	no perceivable impairment
8-9	slight impairment but no disturbance
6-7	perceivable impairment, slight disturbance
4-5	significant impairment, disturbing
1-3	extremely disturbing

Table 2.1: Correlation between user grade and user opinion.

delay and zero loss. After that, a sequence of tasks, each lasting 20 s, is given, with a randomly chosen network setting per task. Participants grade the experience of each task on a scale of 10. An indication of how the user grades correlate with user opinions is shown in Table 2.1.

PARTICIPANTS

The subjective study involved thirteen participants in the age group between 20 and 64 years, with an average of 30 years. Six participants were novice users of the haptic device. Nevertheless, every participant got ample time to familiarize themselves with the experimental setup. No participant suffered from known neurological disorders. Most of the data presented in this paper were collected during the COVID-19 pandemic. At all times, the safety regulations issued by the state were maintained, and extra care was taken to disinfect the equipment often. Because of these concerns, the number of participants is limited. This invites future research with more extensive data sets.

2.6.3. PERFORMANCE ANALYSIS

The data from all participants is aggregated and presented in Figure 2.14. The different types of network settings are separated by gray columns and the different measurements are separated by gray rows. The ETVO penalties are set to $[P_{\textbf{prop}}, P_{\textbf{fixed}}, P_{\textbf{slack}}] = [0.005, 0.01, 0.005]$. We separately take up the performance comparison of ETVO with QoS and QoE methods. In all of our experiments, we employ linear extrapolation at the receiver, as described in Section 2.6.2.

ETVO VERSUS QOS METHODS

In this section, we take up each network setting (described in Section 2.6.2) separately and shed light on the important observations. Each column in Figure 2.14 corresponds to a different network setting. To substantiate the performance of ETVO, we also present discussions relating to different network settings.

1. Network delay. Figure 2.14(a) corresponds to the setting where we introduce a range of network delays. As can be seen, T_{ETVO} can track the network delay with negligible deviation. In addition, it also indicates an offset of approximately 2.5 ms. This can be attributed to the discretization of haptic samples both at the transmitter and receiver, OS-specific scheduling processes, and processing delay. Since ETVO considers the entire TI system as a black box, it is capable of extracting these local delays whose characterization would otherwise necessitate thorough system profiling. As is expected, the delay has a negative correlation with user grades, and T_{ETVO} reflects this accurately. Further, E_{ETVO} correctly indicates negligible degradation in the value domain.

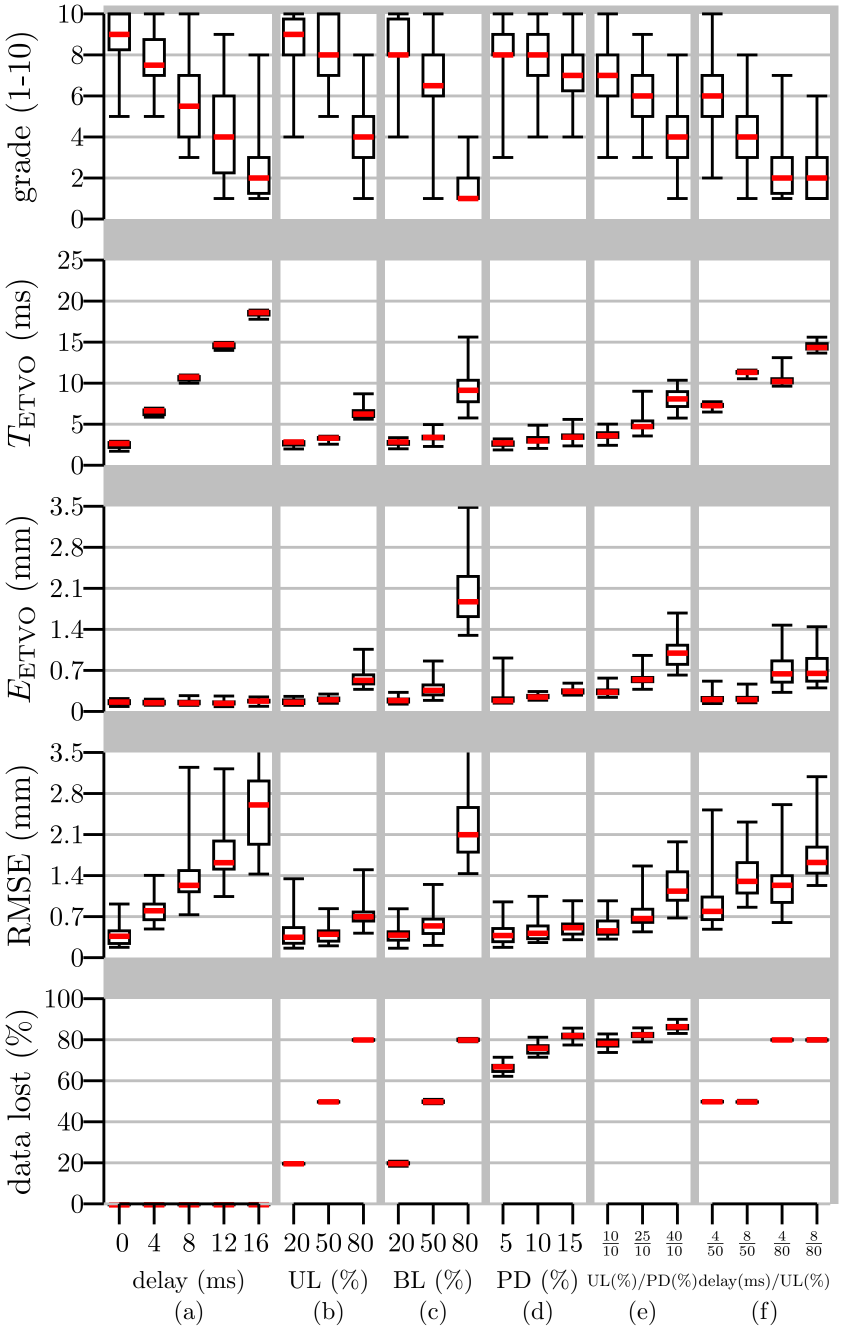


Figure 2.14: Demonstration of ETVO's strong correlation with the user grades along with comparison against QoS and QoE metrics. The experiments are performed under diverse settings of (a) constant network delay, (b) uniform random packet loss, (c) bursty packet loss, (d) perceptual deadband (PD) scheme, (e) uniform packet loss with PD parameter of 10%, (f) constant delay with uniform packet loss. Other acronyms used: UL - uniform loss, BL - bursty loss.

2. Uniform loss (UL). In Figure 2.14(b), we introduce UL in the network. Before we move to discuss the performance of ETVO, we discuss an important concept that is crucial for interpreting our results.

The discretization of haptic signals inherently results in a time gap between haptic updates, which we call *update duration*. This causes a lag between the master and controlled domains. which increases further when packets losses occur. In conventional networking applications, where latency constraints are far more relaxed, the update duration can be largely neglected. However, for TI systems this becomes significant. The average update duration, denoted by Δt_{update}, depends on the packet transmission rate and loss and can be expressed as

$$\Delta t_{\text{update}} = \frac{1}{2 f_s} + \frac{p}{f_s r (p + r)}, \tag{2.5}$$

where the first term is contributed by the sampling rate and the second by packet losses. f_s is the rate at which the haptic device is sampled.

We apply this to Figure 2.14(b). Here, we have a packet transmission rate of 1 kHz, and an average UL of 20%, 50%, and 80%, resulting in Δt_{update} of 0.75 ms, 1.5 ms, and 4.5 ms, respectively. Note that in this setup, the network delay is zero. It can be seen that T_{ETVO} computations corroborate well with the theoretical values accurately, in addition to the 2.5 ms offset that we discussed previously. Further, the trend of T_{ETVO} also matches that of user grade. On the other hand, QoS methods only measure only the packet loss present in the system without quantifying their effect on the user grades.

E_{ETVO} produces a similar trend as T_{ETVO}. A valid question is – if the trend of T_{ETVO} already matches the trend in the user grade, why do we need E_{ETVO}, or vice-versa? The answer to this can be found by comparing different network settings. If we compare the 4 ms delay case in Figure 2.14(a) with 80 % UL in Figure 2.14(b), we see that the T_{ETVO} is approximately equal. However, the corresponding user grades show a dramatic difference. Now, if we consider the information from E_{ETVO} we can see that the latter case reports a significantly higher E_{ETVO}. This explains the lower user grade. This example highlights the significance of the combination of T_{ETVO} and E_{ETVO} being crucial for accurate estimation of TI performance.

3. Bursty loss (BL). In Figure 2.14(c), we present the results for the BL scenario. The average update duration introduced previously and expressed as Equation (2.5) can be applied to the BL scenario also. However, the only difference compared to the UL scenario is the presence of a state-dependent aspect in BL. This means that whether the current packet is dropped depends on the state of the previous packets. Consequently, there is an increased chance of consecutive packet losses in the BL scenario than in the UL scenario. This dramatically increases the theoretical average update duration.

Using Equation (2.5) with $f_r = 1$ kHz, we obtain Δt_{update} of 1.5 ms, 4.5 ms, and 16.5 ms for BL of 20%, 50%, and 80%, respectively. It can be clearly seen that T_{ETVO} correctly reports a higher value than the corresponding values of UL. However, as can be noticed in Figure 2.14(c), the theoretical worst-case delay is significantly higher than what is projected by T_{ETVO}. The reason for this is twofold. Firstly, we use linear extrapolation in our experiments, while, for simplicity, we assumed a zero-order hold extrapolation in theoretical analysis. Linear extrapolation has a significantly higher impact for long episodes of packet loss. In some instances, the estimated velocity can even be higher than the sensed velocity, causing the linear extrapolation to lead the sensed signal. In this case, T_{ETVO} is measured to be lower than the actual delay. On the other hand, linear extrapolation may also produce overshoot, values that might not exist in the sensed signal. This will be captured by E_{ETVO} and not T_{ETVO}. Secondly, the ETVO penalties ensure that the time-offset is changed only when the value-offset reduces significantly. Because of this and the delay profile of bursty loss, the average delay as estimated by T_{ETVO} drops significantly.

Observation on TI reliability. Note that the settings 20 % UL Figure 2.14(b), 20 % BL (Figure 2.14(c)) and the 0 ms delay (Figure 2.14(a)) have no significant difference in user grade. In the case of UL, even up to 50 % loss may become unnoticeable. This indicates that the user experience is not degraded even at significantly lower reliability. This important finding corroborates with a few works that have investigated the haptic reliability requirement [32, 33, 34]. The reason that for this type of data reliability is of little significance, is because a kinematic data stream is being tracked at a very high packet rate. The position of an object doesn't change significantly within a small interval, and thus the error as a result of a lost packet is small.

4. Perceptual Deadband (PD). Next, we study the influence of the PD scheme without any packet loss in the network. As can be seen in Figure 2.14(d), the PD scheme dramatically reduces the number of transmitted packets. However, it is important to note that the PD scheme chooses to omit only the insignificant (redundant) data in the signal. Therefore, although the amount of packets received is significantly smaller, the user experience is good. It can be clearly seen that ETVO measurements match well with the user grades. Further, it can be seen that although the packets received in the case of PD of 15 % and UL of 80 % are similar, the user grade corresponding to the latter is substantially lower. While the packet reception rate is unable to identify this, ETVO is successful in capturing this aspect of the PD scheme.

5. Perceptual Deadband with uniform loss. We now include UL and PD schemes in conjunction. This scenario will see significant haptic updates being dropped by the network. As can be expected, packet loss has a more detrimental effect on the user experience than a scenario without a PD scheme. This can be clearly observed in Figure 2.14(e). Even a 20 % UL with PD of 10 % results in a noticeable change in user grade, whereas up to 50 % UL without PD scheme (Figure 2.14(b)) was barely perceivable. Indeed, ETVO can successfully capture this effect. Further, as per the packets received, the scenarios 25 % UL with PD of 10 % and 80 % UL without PD scheme (Figure 2.14(b)) behave in an identical manner. However, this contrasts with the user grade which is significantly lower in the former scenario. Once again, ETVO measures this accurately reporting higher T_{ETVO} and E_{ETVO} in the former scenario.

6. Network delay with uniform loss. In this setting, we use combinations of network delays (4 ms, 8 ms) and UL (50 %, 80 %). Figure 2.14(f) presents our findings of these scenarios. It can be seen that for a specific network delay, both T_{ETVO} and E_{ETVO} increase with UL. This is because with increasing UL, not only the update duration but also the value error increases. Further, for a specific UL, only T_{ETVO} increases with network delay whereas E_{ETVO} remains identical. This also makes sense as higher delay leads to degradation in the only time domain and not in the value domain. Interestingly,

T_{ETVO} does not accurately reflect the user grades specifically in case of (8 ms, 50 %) and (4 ms, 80 %). However, E_{ETVO} in the latter case is significantly higher signifying yet again the importance of using both T_{ETVO} and E_{ETVO} in conjunction for measuring the TI performance. On the other hand, the packet reception rate misses out on all the fine details that govern the overall performance. This highlights the contribution of ETVO in measuring the TI performance accurately.

ETVO VERSUS QOE METHODS

As a representative of this broad category of metrics, we use RMSE, since, as described in Section 2.2, the vast majority of QoE solutions for TI are RMSE-based. Hence, using RMSE helps us understand the fundamental limitations of these solutions. To reiterate, RMSE is oblivious to the time offset when comparing the sensed and reconstructed signals.

We consider the same network settings considered in the previous section. First, we consider the network delay only case in Figure 2.14(a). The RMSE measurements correspond to the position signal. Due to the inherent problem of RMSE, the effect of delay is treated as value error, and therefore the misaligned samples are directly compared to each other. As a consequence, the calculated error term becomes heavily dependent on the velocity of the signal (speed of movement). For example, for a velocity of zero, a mismatch will not yield an error, but for a high velocity, a mismatch will yield a large error term. Certainly, more delay makes the system worse, but the dependency on velocity introduces a large variance in the performance estimation. This can be observed in Figure 2.14(a) in the RMSE row. On the other hand, ETVO treats the time-offset and value-offset separately, so that the correct samples are compared to each other, leading to significantly better performance.

In Figure 2.14(f), there are combinations of delay and packet loss. For RMSE two observations can be made. Firstly, there is once again a high variance, that does not increase for higher packet loss. Secondly, the average RMSE has a similar trend to T_{ETVO}, but not the addition of E_{ETVO}. Thus, RMSE represents the average delay, with high variance, and this does not match the user grades. This illustrates the fundamental problem when not considering time mismatch. Due to this, samples are compared to the wrong counterpart, and therefore the shapes are incorrectly compared. These two examples illustrate the shortcomings of RMSE and by its extension all QoE methods that do not handle time mismatch. We also show how ETVO does handle mismatches and accurately reflects the user grades.

The problem of high variance in RMSE can also be observed in presence of packet losses, i.e. Figure 2.14(b)-2.14(e). In these cases, although the network delay is zero, the inherent system delay is still present. As a consequence, RMSE is still subjected to high variance. As opposed to this, even the small amount of delay is correctly reported by T_{ETVO}, and by its extension E_{ETVO} is more accurate.

2.7. CONCLUSIONS

In this chapter, we addressed the limitations of existing TI performance metrics when characterizing network performance that facilitates the kinematic modality for haptic bilateral teleoperation. We found the Dynamic Time Warping (DTW) algorithm used in speech recognition as a suitable starting point. We highlighted a few issues in applying

DTW directly for our goal. We developed an analytical framework – *Effective Time- and Value-Offset (ETVO)* – which addresses these issues and can be used to quantify network performance when facilitating bilateral teleoperation. Through objective analysis, using realistic experiments, we demonstrated the improvements of ETVO over DTW in terms of extracting fine-grained time and value offsets. Through subjective analysis, we showed the limitations of QoS and QoE metrics that are used for TI systems. Further, under a wide variety of network settings, we showed that ETVO measurements corroborate well with the user grades and also outperform QoS and QoE metrics. We derived an analytical expression for the average delay of TI sessions and showed that it matches well with ETVO measurements. Additionally, independent of ETVO analysis, we observed that even up to 50 % packet loss results in no significant reduction in user grades when transmitting kinematic data at a rate of 1 kHz.

Understanding the network's ability to support the kinematic modality is a key factor in evaluating its performance. However, assessing a complete haptic bilateral teleoperation system requires a broader perspective that encompasses the entire system, including the application outside the network. This broader analysis will be conducted in the next chapter.

3

Stringent network requirements due to stiff springs[1]

3.1. Introduction

In this chapter, we address Sub-Question 2 as stated in Section 1.4: *What is the correlation between network performance and the specifics of the system?* In Chapter 2, we examined the network's ability to facilitate kinematic data transmissions and its impact on user experience. However, a critical question remains: what underlying factors make a low latency so essential for a good user experience?

Existing metrics fall short because they fail to account for the *stimuli-response* relationship associated with humans interaction with remotely located physical objects. This gap complicates our understanding of the impact of latency and reliability on system performance. Gaining clarity on these requirements is crucial, as the stringent network requirements pose some of the most significant challenges to realizing haptic bilateral teleoperation.

The lack of accurate network requirements risks significant over-provisioning of resources to support haptic bilateral teleoperation. At the same time, viable configurations may be overlooked due to incorrect assumptions about their performance capabilities. Scaling network support for haptic bilateral teleoperation applications becomes highly challenging without an objective metric to assess the application performance. This chapter aims to bridge this gap, providing a foundation for more informed and efficient provisioning of HBT applications.

We set out to create the first metric that captures the effect of the network on haptic feedback. We address the problem from a networked control systems viewpoint by modelling human interactions with physical objects. We use a representative model and

[1] This chapter is based on the publication titled *"TIM: A Novel Quality of Service Metric for Tactile Internet"*[35].

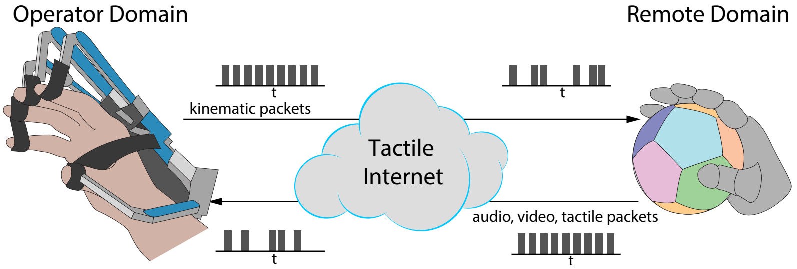


Figure 3.1: A typical Tactile Internet (TI) system highlighting the operator and remote domains and network characteristics.

derive a closed-form expression for the short-term response based on measured network conditions. Based on Weber's law of Just Noticeable Difference (JND), we propose a real-time TI metric called the Tactile Internet Metric (TIM) that estimates the amount of undesired haptic feedback the network introduces. The TIM score can determine the application requirements and, thus, the network conditions to satisfy the application. Therefore, TIM can be used to seek guarantees from the network provider. Further, we propose a method called the channel compensation spring that adjusts the application parameters to compensate for present network latency.

CONTRIBUTIONS

The contributions in this chapter are listed below.

① We provide a simplified generic framework for Tactile Internet (TI) applications by modeling them as networked control systems. This work is the first to adopt a human-in-the-loop control-theoretic approach for designing a TI metric (Section 3.3).

② We derive an expression for network-induced delay using a Markov model, enabling a deeper understanding of delay dynamics in TI systems (Section 3.3.2).

③ We theoretically derive a real-time metric, TIM, that quantifies the quality of a TI application. TIM continuously compares the application's performance under real network conditions to its performance under ideal conditions with no packet loss or delay (Section 3.4).

④ We propose a novel method to tune a channel compensation spring using TIM, allowing the system to adjust dynamically to varying network conditions. This method has been implemented and tested on two TI applications (Section 3.4.3).

⑤ We design realistic TI experiments to perform both objective and subjective evaluations of TIM. The results demonstrate that subjective user experience aligns closely with the proposed metric (Section 3.7).

3.2. RELATED WORK

In recent decades, there has been a significant body of work in the field of Networked Control Systems (NCS). An overview of control methodologies for NCS can be found in [36, 37, 38]. In particular, fundamental issues in NCS due to network-imposed constraints (such as delay, jitter, noise) are discussed in detail. The effect of network delay on control

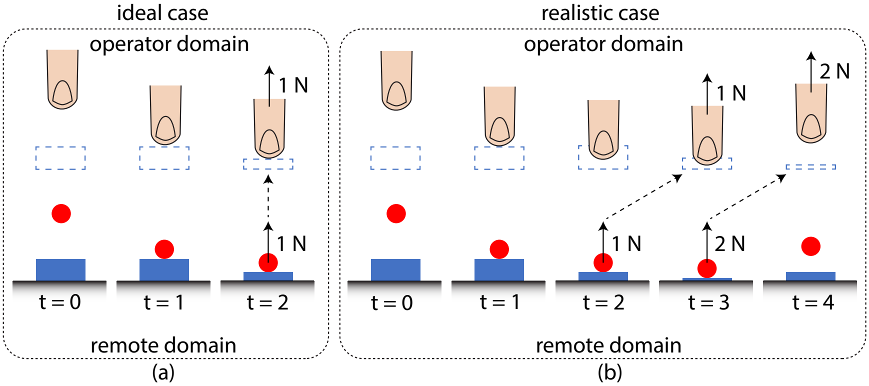


Figure 3.2: Illustration of the detrimental effects of the network on TI interaction. The operator intends to press a switch (blue block) through the teleoperator (red circle) using a tactile glove virtually (dotted blue block). (a) In the ideal case, force is experienced right at the instant of contacting the object. (b) In a realistic network case, the force feedback is transported with a variable delay and causes significant performance degradation.

system performance has been studied in [39, 40, 41, 42, 43]. In particular, the effect of delay on asymptotic stability, as well as control paradigms to overcome these effects have been studied. Delay compensation solutions based on predictive and adaptive (gain tuning) approaches are presented in [44]. While these results provide excellent tools for control system engineers to design network-resilient feedback strategies, they rely on an accurate representation of network dynamics within a mathematical framework that is compatible with traditionally used dynamical systems models. Furthermore, while these models are in the control systems domain, the results must be made consumable for networking community and engineers, i.e., to quantify the effect of network parameters on the TI performance and devise novel solutions for enhanced performance. This calls for a new design of an accurate and accessible framework, which is the focus of this work. An overview of related works to specifically network performance metrics has already been provided in Chapter 2.

3.3. THE PROBLEM AND SYSTEM MODELING

Consider a simple application of controlling a robot arm over a network to push a switch in the controlled domain using a VR headset and a tactile glove. This is schematically depicted in Figure 3.2, where the solid blue block represents the switch on the rigid platform, and the red circle is the teleoperator. The dashed blue block is the switch, as displayed via the VR headset at the operator's end. Under ideal conditions (zero latency and losses), as shown in Figure 3.2(a), the force feedback is experienced exactly when the switch is touched. In a real-world TI system over a network, as shown in Figure 3.2(b), there exists a lag between the two domains. As a result, the operator keeps pushing the virtual switch until $t = 3$ while the physical contact is made at $t = 2$. This additional penetration generates a significantly larger force manifesting as an unanticipated jerk to the operator's hand (from $t = 4$). This behavior severely hampers the user experience.

To quantify the requirements of TI applications and assess if a TI system can provide

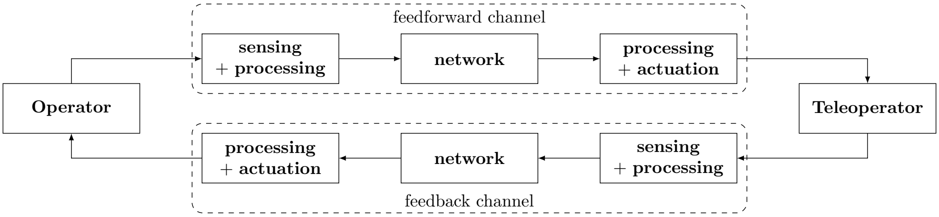


Figure 3.3: An overview of the TI system model. The feedback and feedforward channels include modules that influence the performance.

the necessary performance guarantees, we focus on the scenarios that pose the most stringent demands. If such scenarios are supported, it is reasonable to assume other scenarios with more relaxed constraints can also be supported. Generally, for TI applications, the critical scenario is whenever there is a drastic change in force feedback, such as at the interface of air and hard objects.

While it is known that TI requires low latency, there exist no tools to quantify the impact of the network characteristics on the tactile experience. Accomplishing this requires a deep understanding of the dynamics of TI systems encompassing a network and a metric to express the deviation from the ideal behavior. In this work, we aim to bridge this gap. In the following, we provide an abstract model of the TI system.

3.3.1. TI SYSTEM MODELLING

Understanding the TI system dynamics while being robust to human subjectivity requires objective models to describe the system. A TI system comprises several sensors, actuators, and a network. A complete system model involving these modules paves the way toward determining precise performance requirements, developing efficient TI solutions, and carrying out reproducible performance evaluation.

To aid in modelling, we divide the TI system into three parts: the *channel, operator,* and *teleoperator*. This is schematically depicted in Figure 3.3. The *channel* comprises all modules starting from sensors in the master domain to the actuators in the controlled domain.[2] This means that any pre- and post-processing steps, like filtering, compression, and prediction, are also part of the channel. Accordingly, we have the *feedforward channel* from the master to the controlled domain and the *feedback channel* in the opposite direction. As explained in Section 3.1, the operator is the human controller and the teleoperator is the controlled robot device. We will now model these parts using tools from both communication and control theory. First, we take up channel modelling using existing TI metrics and then we move to tactile interaction with an object.

3.3.2. EFFECTIVE DELAY

In this work, we consider effective channel delay, denoted by τ, as the overall round trip delay induced by the channel. We consider τ as the most important indicator of channel performance. An ideal channel realizes the reproduction of sensed data with zero channel delay. Besides network latency, packet loss and rate also impact the channel

[2] This is in contrast with the standard network parlance where “channel” refers only to communication links.

delay. For example, lost packets lead to missing information forcing the receiver to wait for subsequent packet arrival. This increases the overall effective delay. Similarly, packet rate influences how quickly the information is delivered to the other domain.

Effective channel delay can be determined using two types of delay indicators (a) signal-oblivious and (b) signal-aware. Signal-oblivious indicators are insensitive to the sensed signal, using indicators like network latency and packet loss. On the other hand, the signal-aware indicators consider the mismatch between the sensed and reconstructed signals for performance assessment to capture the detrimental effects of the channel. Common signal-aware indicators are position, velocity, and force. While signal-oblivious indicators are significantly easier to work with, signal-aware indicators provide a more holistic performance assessment.

In this work, we will consider both types of indicators to get a broadly acceptable delay model and we use it as the basis for the design of TIM in Section 3.4.

Signal-oblivious delay indicator ($\tau_{\mathbf{QoS}}$). We consider three QoS indicators: latency, packet loss, and packet rate. In TI literature, these performance indicators are measured and treated separately [10, 9, 12, 14]. For a given a packet rate f_t, we denote the effective delay derived from QoS metrics as τ_{QoS}. We identify three components that contribute to τ_{QoS}. First, we have the network latency τ_{network}. Second, we have the delay due to packet rate, which is half of the transmission period $\frac{1}{2f_t}$. Finally, we have the delay due to packet loss, which causes an absence of information and contributes to delay. Careful attention needs to be put to the effect of consecutive packet losses, which can contribute to significant amounts of delay. All these components together allow us to calculate τ_{QoS} as,

$$\tau_{QoS} = 2\Big(\tau_{\text{network}} + \frac{1}{2f_t} + \frac{p}{f_t r(p+r)}\Big), \tag{3.1}$$

where p is the probability of packet loss after a successful transmission and r the probability of success after a loss. Because the delay is round trip, both the feedforward and feedback channel are added together. Due to the paucity of space, we present the details of our method of finding the closed-form expression, Equation (3.1), in the online appendix.

Signal-aware delay indicator ($\tau_{\mathbf{ETVO}}$). Taking the root mean square error (RMSE) of signal-aware performance indicators such as position and velocity is insufficient as it only indicates the error between sensed and reconstructed signals without conveying anything about latency or packet loss. Further, objective QoE metrics are unsuitable for use for reasons described in Section 3.2. Recently, a framework called Effective Time-and Value-Offset (ETVO) [5] proposed simultaneously estimating both delay and error using a modified Dynamic Time Warp algorithm. ETVO can estimate instantaneous delay based on the data acquired from a real human experiment. This provides us with an alternative method to estimate the delay caused by the network in a signal-aware manner. This means that the impact of signal-aware solutions can be captured. Thus, it is prudent to adopt ETVO for the signal-aware channel modeling. We take τ_{ETVO} as the average Effective Time Offset (ETO) as derived in [5], which yields,

$$\tau_{\text{ETVO}} = \frac{1}{N}\sum_{k=0}^{N} ETO[k], \tag{3.2}$$

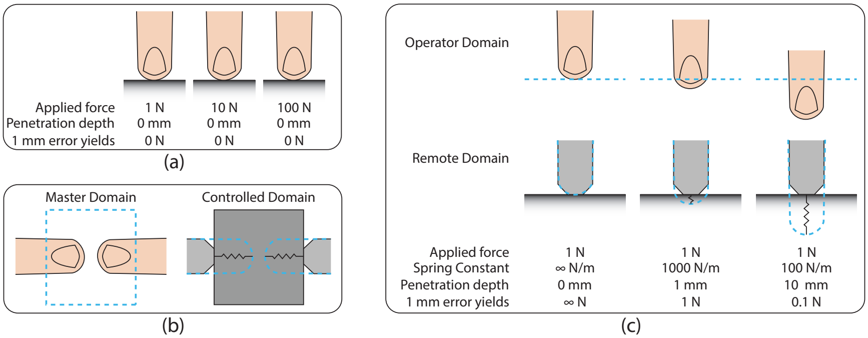


Figure 3.4: (a) Direct interaction between a finger and an infinitely rigid surface. Regardless of how hard the finger presses, it will not deform the surface. (b) An illustration of the problem arising when performing a grasping motion. To have a grip on the object through imaginary strings, the fingers need to grab tighter than the size of the object. When the spring constant is low, there is a risk of the fingers colliding against each other. (c) TI interaction over a network: When operator pushes down into the virtual surface (blue dashed line), the robot presses down on the real surface. An imaginary spring is drawn to the target position to calculate the force applied on the surface.

where N is the number of samples considered in the system.

3.3.3. TACTILE INTERACTION MODEL

Let us first consider a regular physical interaction with a finger and a highly rigid, fixed surface (Figure 3.4(a)). The finger will never penetrate the surface, irrespective of applied force. For TI applications, these surface interactions must be approximated.

Approximating surface interaction. In TI interactions, the haptic feedback is generated based on a kinematic signal. Hence, we need a way to transform the signal into haptic feedback. A standard technique is to use an imaginary spring to approximate the interactions with any surface [45]. We can use $F_s = -kx$, where k indicates the spring constant. A higher k means a stiffer spring, and vice-versa. This spring is drawn between the object's surface and the target position as received from the master domain. This yields the "penetration depth" – the depth of the target position from the surface. The force is computed as the product of the penetration depth and k. A higher k produces force corresponding to harder objects. Hence, k is an application parameter that can be tuned based on the nature of objects in the controlled domain. The choice of k can greatly impact the overall performance. An illustration is given in Figure 3.4(c). Here, three scenarios with different values of k are shown, along with the impact of effective delay. In the first case, a hypothetical spring with $k = \infty$ is shown. While this perfectly mimics the regular interactions, it produces extremely large force even for a small τ. This results in the finger being pushed away from the surface (not depicted here, but explained earlier in Figure 3.2(b)). As k reduces, the same amount of error produces a smaller force and a smoother experience. The smaller force produced for a lower k results in the target position going further below the table surface (shown with the finger crossing the blue dashed line in the controlled domain). Although this reduces the experience of feeling hard objects, it causes the system to reduce undesired force.

While reducing k is a potential solution to addressing system errors, the lower limit depends on the type of TI application. For example, an application involving grasping objects can be problematic if k is small, as the two fingers may touch each other, failing to provide a grasping experience. This undesired behavior is illustrated in Figure 3.4(b). Hence, one must balance the undesirable effects of (high or low) k to provide a realistic TI interaction experience. We assume that k is tuned in such a way that under the maximum tolerable force, interactions like grasping work as desired, leaving the performance of the tactile interaction model as the main concern.

Next, we will concisely derive a theoretical model for surface interaction. A detailed explanation, derivation, and implementation notes for Matlab are provided in the online appendix. We refer the readers to [46] for details on control-theoretic approaches. We assume the master domain to have a falling object with a specified mass with an acceleration of $\ddot{x} = -g + \frac{F_s}{m}$, where g is the gravitation constant and m is the mass of the falling object (throughout this work $\dot{a}$ indicates the time derivative of a). We will only use this model for short-term response and can therefore neglect the damping terms. The choice of a mass hitting a surface represents many interactions and can therefore be used in a wide variety of usecases. For example, consider a fingertip touching a cup from the side.

The reactive force F_s when delayed by τ can be written as a state space representation given as,

$$\begin{bmatrix} sX_1(s) - x_1(0) \\ sX_2(s) - x_2(0) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \frac{-k}{m}e^{-\tau s} & 0 \end{bmatrix} \begin{bmatrix} X_1(s) \\ X_2(s) \end{bmatrix} + \begin{bmatrix} 0 \\ -g \end{bmatrix} U(s),$$
$$Y(s) = \begin{bmatrix} 1 & 0 \end{bmatrix} X(s), \tag{3.3}$$

where $Y(s)$ is the transform of the observed position of the falling object, $U(s) = 1/s$ i.e., a transformed step function and $e^{-\tau s}$ is the Laplace equivalent of the delay τ. Note that $x_1(0) = 0$ and $x_2(0) = \dot{x}_{\text{impact}}$ are the initial values of the position and velocity, respectively. This form is a standard Laplace variant of a state-space model which yields,

$$Y(s) = \frac{\dot{x}_{\text{impact}}\, s - g}{s^3 + \frac{k}{m} s e^{-\tau s}}. \tag{3.4}$$

The position trajectories for the zero delay (ideal) case can be computed via the inverse Laplace transform as,

$$x_{\text{ideal}}(t) = \mathscr{L}^{-1}\{Y(s)\} = \frac{V}{\sqrt{\frac{k}{m}}} \sin\left(\sqrt{\frac{k}{m}}t\right) + \frac{gm}{k}\left(\cos\left(\sqrt{\frac{k}{m}}t\right) - 1\right). \tag{3.5}$$

In the case of non-zero delay, the transfer function is approximated via a rational Padé approximation known to work well in approximating delay. This converts $Y(s)$ into the form

$$\hat{Y}(s) = \frac{\beta_0 + \beta_1 s + \cdots + \beta_{n-2} s^{n-2}}{\alpha_0 + \alpha_1 s + \cdots + \alpha_{n-1} s^{n-1} + s^n} \approx Y(s), \tag{3.6}$$

where n is the order of the Padé approximant. The key point of the Padé approximation is to remove the delay term $e^{-\tau s}$. With the term approximated, we can solve the system with standard methods. From the approximation, we can create a *Controllable canonical realization*. We now obtain the position trajectory of the (approximated) delayed system by setting the initial conditions to zero and inputting an impulse to get,

$$x(t) = \mathscr{L}^{-1}\{\hat{Y}(s)\} = \hat{\boldsymbol{C}} e^{\hat{\boldsymbol{A}} t} \hat{\boldsymbol{B}}, \tag{3.7}$$

where the matrices $\hat{C}$, $\hat{A}$, $\hat{B}$ depend on $\dot{x}_{\text{impact}}$, g, m, k, and τ. For τ, we can use τ_{QoS} or τ_{ETVO} using Equation (3.1) and Equation (3.2), respectively. Due to the paucity of space, the full derivation is provided in the online appendix.

In practice, the given derivation does not need to be calculated by hand. In particular, Matlab has excellent support for these types of calculations. The code for computing Equation (3.7) is given below. Note that Line-4 directly implements Equation (3.4). A detailed explanation of the code is provided in the online appendix. The calculations are computationally inexpensive and can be easily executed in real-time.

Listing 3.1: Matlab function that calculates Equation (3.7)

```
1  function [x] = bouncingMassPade(k, tau, v_impact, m, g, t)
2  s = tf(’s’);
3  Y = (v_impact*s-g)/(s*(s^2+k*exp(-tau*s)/m));
4  Y_hat = pade(Y,6); %6th order Pade approximation
5  x = impulse(Y_approx,t); %Impulse response
6  end
```

3.4. TIM: PROPOSED OBJECTIVE METRIC FOR TI

Based on the TI system model developed in the previous section, we propose Tactile Internet Metric (TIM). TIM is an objective metric designed to measure the performance of TI sessions in real-time. TIM relies on the measured performance departure of a realistic TI system against an ideal system. To the best of our knowledge, our work is the first of its kind to propose a metric by analyzing the various components of a TI system at a fine-grained level.

3.4.1. DESIGN GOALS

Following are the design goals of TIM for it to be a useful TI metric and be widely applied across TI use-cases.

Objectivity: TIM should be independent of human subjectivity in skills and perception to provide quantifiable performance and yield reproducible measurements.

Short-term response based: TIM should be based on the short-term behavior of the interaction since we are only interested in the instantaneous tactile response.

Low complexity: Since the input to channel model (network parameters) changes in real-time, our objective metric should be computationally inexpensive. This makes it easier to deploy, analyze, and modify the metric.

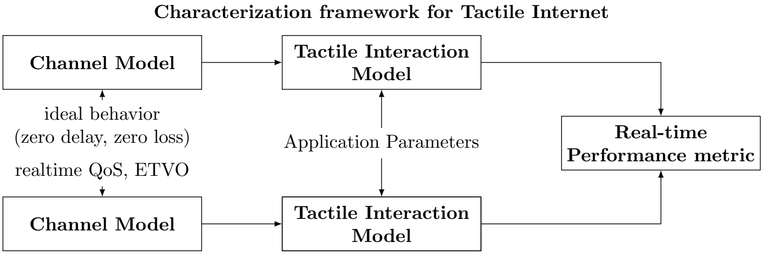

Characterization framework for Tactile Internet
Channel Model
Tactile Interaction Model
ideal behavior
(zero delay, zero loss)
realtime QoS, ETVO
Application Parameters
Real-time Performance metric
Channel Model
Tactile Interaction Model

Figure 3.5: A schematic to show how the channel and tactile interaction models are used to estimate the departure from ideal TI system behavior to yield TIM measurements.

Easily tunable: The design parameters should match the target TI usecases. Additionally, the number of design parameters should be kept at a minimum with a simple choice of values.

Monotonic behavior: The metric should be monotonically associated with TI system parameters, such as τ and k. For example, all else being equal the metric should always infer that a higher τ deteriorates performance.

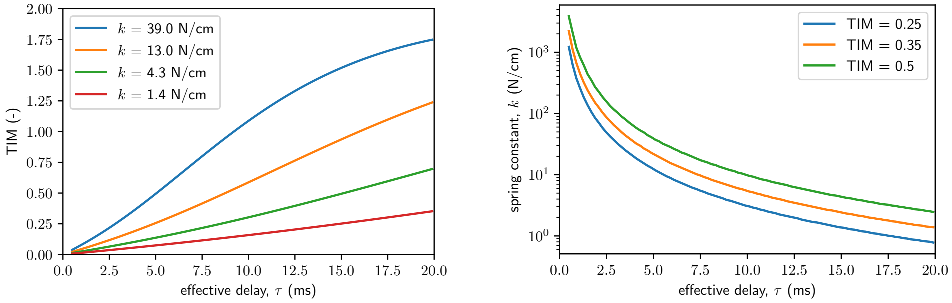
Real-time measurements: To continuously monitor the TI system performance and user experience, TIM should provide real-time measurements. This will aid in provisioning network and system resources on the fly to meet the necessary application requirements.

3.4.2. DESIGN AND ANALYSIS

Figure 3.5 shows a schematic diagram of how we leverage the TI system model to aid in the design of TIM. The tactile interaction model and its accompanying application parameters are directly based on the TI application. The ideal system behavior is considered to be the behavior when the channel is behaving perfectly. This means that any sensory data is reproduced with zero delays and loss in the other domain. It is important to note that the TI application is assumed to work well under ideal conditions as we take that as the baseline for performance evaluation.

The channel model takes direct performance indicators like QoS or ETVO in real-time. For any realistic TI system, the performance would be lower than the baseline. We take this departure from the baseline to formulate TIM. Note that the tactile interaction model should be chosen independently of subjective components like a human controller. Equation (3.7) allows us to project the trajectories of the ideal and realistic TI system for evaluating the system performance in real-time.

To estimate the effect of the TI system on user perception, we rely on Weber's law of Just-Noticeable-Difference (JND) [13]. We can use this to conclude that the system performs adequately if $\frac{\Delta I}{I} <$ JND, where I is the intensity, and JND is a threshold. In our case, we take I as the amount of force feedback, and ΔI is the difference in force feedback between the ideal and delayed response. We take the time of exit (denoted by T_{exit}) of the



(a)

(b)

Figure 3.6: (a) Plotted are TIM values for a given amount of spring constant and τ. One can see that TIM approaches zero irrespective of the k as the effective delay approaches zero. The spring constants used match those in the user study. (b) Plotted are delay and spring constant pairs that yield a constant amount of TIM. One can either determine how much delay can be tolerated for a given maximum stiffness, or the maximum stiffness that can be tolerated for a given network performance.

object from the surface in the ideal scenario, which can be derived from Equation (3.5) as,

$$T_{\text{exit}} = \frac{2(\tan^{-1}(\sqrt{\frac{k}{m}}\frac{\dot{x}_{\text{impact}}}{g}) + \pi)}{\sqrt{\frac{k}{m}}}.$$

We choose the intensity to be l^2-norm of the ideal force and ΔI as the l^2-norm between the ideal force and the delayed force. We can then derive the expression for TIM as

$$\text{TIM}(k, \dot{x}_{\text{impact}}, m, \tau) = \sqrt{\frac{\int_0^{T_{\text{exit}}} \left(x_{\text{ideal}}(t) - x(t)\right)^2 dt}{\int_0^{T_{\text{exit}}} x_{\text{ideal}}(t)^2 \, dt}}, \tag{3.8}$$

where k, $\dot{x}_{\text{impact}}$, and m can be tuned to match the target application. To reiterate, τ can be obtained from either QoS metrics (τ_{QoS}) or from ETVO (τ_{ETVO}). We can use our metric in the same way as Weber's law of JND and set a threshold below which the system performs adequately. In Figure 3.6(a), we show the TIM scores for a wide range of τ and k. It can be seen that TIM is monotonically associated with τ and k (one of the design goals). Empirically, any system that produces TIM > 1 will not be favorable for TI interaction as it produces twice the amount of ideal force feedback. One can also see how lower k can tolerate a significantly higher effective delay.

Based on Equation (3.8), we can compute k for a given τ (and vice-versa) and TIM score. This is shown in Figure 3.6(b). The plotted results can be directly used to identify whether the channel and application specifications are sufficient to meet a target TIM score. For example, if the target TIM score is 0.25 and τ =10 ms, then only TI applications with $k \leq 3$ N/cm can be supported. Otherwise, a channel with lower τ should be used or k should be reduced for meeting the target TIM score.

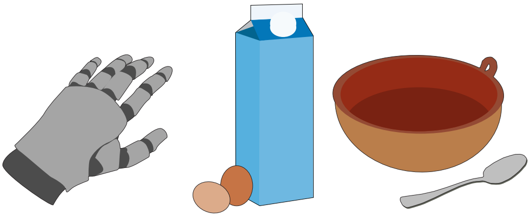


Figure 3.7: An example of a TI application that can be characterized and tuned using our proposed metric TIM.

3.4.3. Channel compensation spring

The spring constant k as modeled in Section 3.3.3 is part of the given TI application. This means that when the channel is assumed to be perfect, a spring constant of k would give the intended behavior. We can use the notion that channel disturbances are less significant for smaller spring constants to our advantage. We assume that we cannot directly meddle with the application at the endpoints. Instead, we propose the use of a virtual "channel compensation spring" with spring constant k_c. This spring is virtually added to the existing dynamics to reduce the effective stiffness and therefore lower the negative effects of the channel. This addition changes the controlled domain side of the tactile interaction model into

$$\frac{1}{k_{\text{total}}} = \frac{1}{k} + \frac{1}{k_c}, \tag{3.9}$$

where k_{total} is the resulting spring constant that determines the systems dynamics and its sensitivity to delay. For a given network performance one can look up what the required maximum k_{total} is using Figure 3.6. Then using Equation (3.9) one can derive the amount of compensation to guarantee satisfactory performance. The channel compensation spring has a relatively large effect on interactions with rigid objects when compared to soft objects. However, this addition changes the system dynamics. A separate verification is needed to make sure that the increased softness does not make the experience insufficient. If the system is found to perform insufficiently despite the added channel compensation spring, a better channel is needed to support the TI application.

3.5. Implementation Notes

This section illustrates how the proposed metric TIM can be used to characterize and improve TI systems. To provide an intuitive understanding, we take a concrete application and walk through the steps needed.

Let us take a simple example of a TI application as shown in Figure 3.7. A human operator wears a haptic glove and a head-mounted display. In the controlled domain is a robotic hand next to a table with cooking ingredients. The operator uses a TI application to prepare breakfast remotely. The robotic arm must delicately handle the milk carton, eggs, ceramic bowl, and spoon. A simple TI system (with only signal-oblivious channel modules) is in place, where each side transmits a packet after every measurement at a steady rate of 1 kHz. With this application in mind, we give the broad steps required to characterize and tune the system using TIM.

1. Identify critical interaction: In TI applications, there typically are multiple interactions with different requirements. In this case, we identify the most critical interaction as picking up an egg without breaking it. We assume that if a TI system can perform adequately well in this scenario it can provide adequate performance for the entire application. Note that in this task there is a hand with multiple fingers involved, similar to the schematic in Figure 3.4(b).

2. Build tactile interaction model: For the identified critical interaction, a tactile interaction model is built. The fingers involved in grasping the egg can be considered separately, which means that we can use the tactile interaction model provided in this work. The system should provide adequate performance when there is no channel deterioration. This value of k is supplied to the tactile interaction model.

3. Measure effective delay: The channel components, including the network, need to be captured by an existing metric for the delay model. Because of the application's simple behavior, we use QoS in real-time to measure the feedforward and feedback channel performance. τ_{QoS} can be calculated in real-time if QoS parameters can be measured in real-time.

4. Calculate TIM score and evaluate: Using the above ingredients, the corresponding TIM score is calculated. We use the concept of JND to investigate whether the network causes a significant deterioration in performance. The threshold of acceptable TIM is dependent on the application. In this case, we empirically set the threshold at 25%. If the TIM score is below the threshold, the application is adequately supported by the given TI system. If the effective delay is calculated in real-time, then TIM can also be calculated in real-time.

5. Incorporate channel compensation Spring: If the TIM score exceeds the target threshold, a channel compensation spring can be implemented, as described in Section 3.4.3. Firstly, Figure 3.6(b) should be used to determine the maximum acceptable k_{total}. Based on this, k_c is calculated according to Equation (3.9). Provided that the application supports dynamic alteration of the channel compensation spring, the compensation can be applied dynamically in real-time.

A suitable tactile interaction model must be developed in cases where the tactile interaction model deviates significantly from the critical interaction. For example, when deploying TI to move in a fluid, like swimming in water. In this case, the fluid adds dynamics not captured by the tactile interaction model supplied in this work. In such a case. a similar approach can be used as presented in this work. However, we believe the given model covers most TI use-cases with a human-in-the-loop.

3.6. EXPERIMENTAL SETUP

We evaluate TIM using two virtual environment (VE) applications to generalize our findings and also to demonstrate TIM's broad applicability.

TEST SETUPS

In the master domain, a Novint Falcon is used as a haptic device. On the controlled side, the haptic and visual rendering is done using the Chai3D physics engine. Haptic and visual frame rates are calibrated to 1 kHz and 60 Hz, respectively. The master domain houses the haptic device and a monitor. The controlled domain houses the VE, and the two domains are connected via a real network. To control the network settings, we use NetEm – a standard network emulator to tune the network latency and packet losses. In the master domain, the force is fed to the haptic device. Our experimental setups are shown in Figure 3.8.

(a) Bounce application consists of four surfaces (A, B, C, and D) with different hardness (k) to emulate different levels of bounce when interacting with them. This is shown in Figure 3.8(a). The *Bounce* application is designed to precisely match the modeled physical behavior. The VE is designed with a minimal amount of objects to ensure a consistent experience across different participants. When a particular surface is tapped, it produces a force corresponding to its k and the network characteristics.

(b) Slide application houses a cube that can be slid on the floor. A gate with an opening slightly bigger than the cube's width is positioned at the center. The participant is tasked with navigating the block through the gate. This task invites the participant to experience a more varied set of actions, such as pushing and navigating the cube accurately through the opening, than the Bounce application. This is shown in Figure 3.8(b).

In the future, additional verification of our metric is desirable with different types of haptic devices.

SETUP FOR SUBJECTIVE EVALUATIONS

In our subjective experiments, we give ample time for each participant to familiarize themselves with the TI setup under ideal network conditions – zero latency and packet loss. After this, the data collection begins. For the Bounce application, we empirically choose k from [1.4, 4.3, 13, 39] N/cm. In each experimental run, k is assigned randomly to each surface without the participant's knowledge to remove biases. The participant is informed that each surface is supposed to mimic a rigid surface. The human participant interacts with the VE surfaces and provides a subjective grade for each surface based on the experience of interaction and its similarity to a rigid surface as per Table 5.1.

The Slide application is tested on a subset of settings used for the first experiment. Participants are invited to experiment to form an opinion on how well the application operates.

The subjective study involved seventeen participants in the age group roughly between 20 and 40 years, with an average of approximately 25 years. No participant suffered from known neurological disorders. The data was collected anonymously and with consent from the participants.

We also employ Perceptual Deadband (PD) [13] – a haptic compression scheme that works by identifying perceptually insignificant samples based on a pre-defined threshold. Such samples need not be transmitted resulting in an improvement in bandwidth requirement. This enables us to measure the performance of TIM with standard haptic encoding techniques.

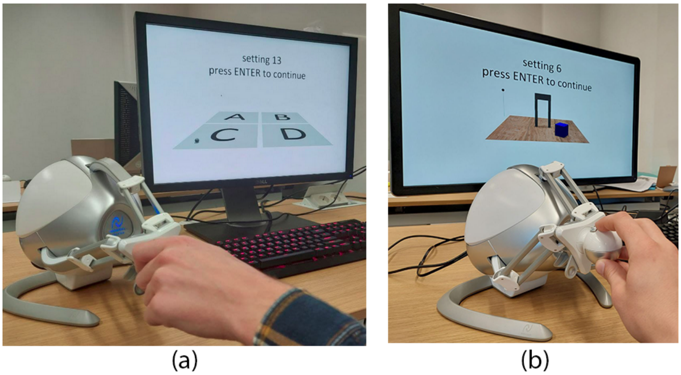


Figure 3.8: TI experimental setup in our work showing the human participant: (a) in the virtual environment interacting with surfaces A, B, C, and D that are having different spring constants indicating different types of surface hardness. (b) in the virtual environment interacting with a cube that can be pushed through a narrow gate.

Table 3.1: Description of subjective grading.

10	no perceivable impairment
8-9	slight impairment but no disturbance
6-7	perceivable impairment, slight disturbance
4-5	significant impairment, disturbing
1-3	extremely disturbing

3.7. Performance Evaluation

In this section, we first conduct an objective evaluation of the Bounce application, comparing experimental results to theoretical predictions to validate the accuracy of our models. This is followed by a subjective evaluation of the Bounce application, where user feedback is analyzed to assess the impact of network conditions on perceived performance. Finally, we extend the subjective evaluation to the Slide application, demonstrating the generalizability of our findings across different TI scenarios.

Objective evaluation of Bounce application

For objective evaluations, we secure a weight to the haptic device such that gravity pulls the device downward resulting in continuous interaction with the surfaces. This enforces continuous haptic interaction without involving human participants.

In Figure 3.9, we present the temporal variation of the haptic device trajectory as it is dropped on the VE surfaces for different combinations of k and latency and compare it against our simulations of Equation (3.7). A position below the surface yields an applied force proportional to the penetration depth and k. The force signals converge to the point where they match the gravitational pull on the attached weight. One can see that the simulations corroborate well with our real trajectory for the short-term response. Deviation increases over time because long-term effects like damping are neglected in

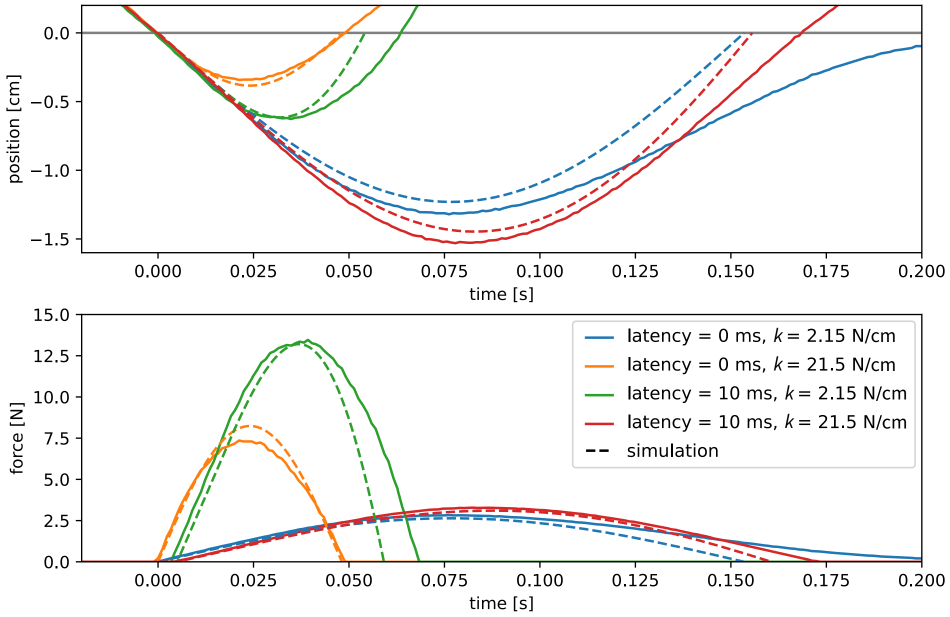


Figure 3.9: Temporal variation of haptic device trajectory and force experienced as a function of latency and k compared with the simulated measurements showing the efficacy of our theoretical approximations.

the tactile interaction model. One can see that the effect of delay on the higher k is more dramatic than a lower value, which matches our expectations.

SUBJECTIVE EVALUATION OF BOUNCE APPLICATION

The participants are asked to rate the application in terms of the user experience and the realistic nature of the surfaces. Since multiple k values are used, this can be interpreted as additions of channel compensation springs.

In Figure 3.10(a), the user grade is plotted against k and network latency. One can see that the addition of latency negatively impacts the user grade. It can be observed that lower k improves the performance in case of bad network conditions.

Inference 1. A lower k, due to soft objects or a channel compensation spring, significantly reduces the negative impact of high delay.

Further, it can be seen that lower k degrades the performance under good network conditions. Specifically, the optimal k that results in the best user experience decreases with increasing latency.

Inference 2. Network compensation should be applied only when the channel is detrimental to user experience.

From Figure 3.10(a-c) we can see a cutoff region between a TIM score of 0.25 and 0.5, where the network starts significantly affecting the user grades. Note the strong similarity between the TIM score derived from QoS and ETVO, which shows that for this type of channel, QoS is sufficiently accurate. This result can be used to derive the required τ for a given k. Likewise, we can identify the subset of TI applications, those with a sufficiently

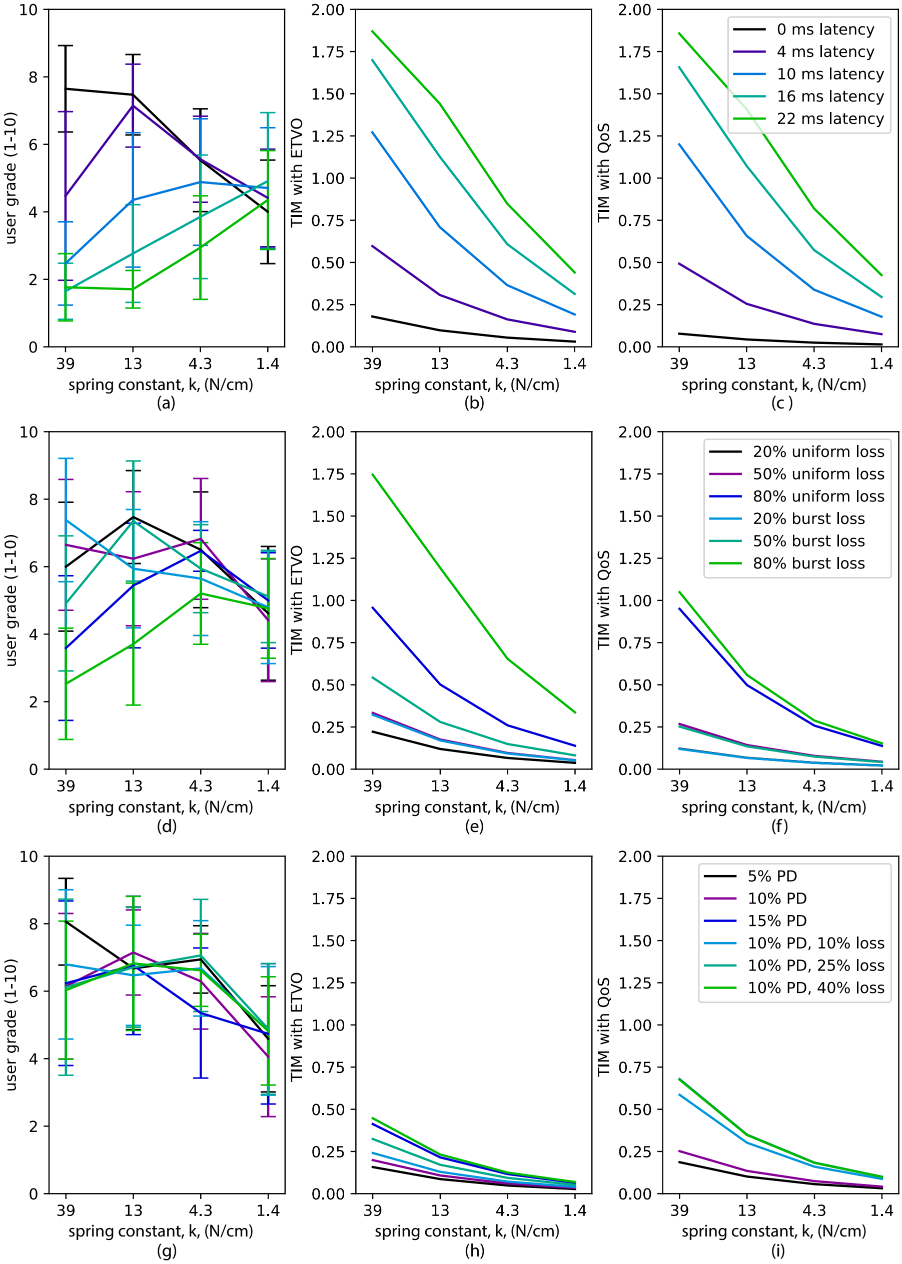


Figure 3.10: User grades and corresponding TIM scores across different network latency, packet loss, and perceptual Deadband settings by QoS and ETVO models for the Bounce application.

low k, to be supported by a given TI system performance. While these insights provide a preliminary understanding of the underlying dynamics, a more detailed analysis of TIM scores is needed for an application and channel to facilitate effective TI interaction.

Inference 3. Given a TI network, TIM can indicate the types of TI applications that can be supported. Further, given a TI application, TIM can specify the network requirements for a seamless experience.

In Figure 3.10(d-f), we sweep over the range of packet losses (both uniform and burst). It can be seen that the burst loss scenario is significantly worse than the corresponding uniform loss case in both user grades and TIM scores. This matches our expectations as consecutive losses add to the effective delay (as described) and thereby deteriorate synchronization between the master and controlled domain. One can see that ETVO recognizes that burst loss is significantly worse than uniform loss. This matches well with the user grades.

Inference 4: Through TIM scores, one can reliably distinguish the impact of uniform and burst packet losses.

In Figure 3.10(g-i), we show the results with PD and combinations of PD and packet loss. A change in PD does not significantly impact the user grade, with all of the grades being relatively close together. With ETVO, TIM shows only a marginal difference between PD values. Further, it can be seen that the worst-performing settings are combinations of uniform packet loss and PD, but even then, the hit on user experience is marginal. In all these cases, TIM reflects the user experience very well.

Inference 5: Using a signal-aware delay indicator increases the efficacy of TIM as they capture the intricacies of the tactile signal, including the effect of methods like PD.

Subjective evaluation of slide application

For the Slide application, a subset of the network settings is used in the Bounce application, as it is more time-consuming and has the risk of causing fatigue to the user. This is a more general-purpose application involving varied haptic feedback.

When comparing Figure 3.11(a-c) with Figure 3.10(a-c) we can see similar trends. The effect of the channel is most profound for a stiff system and marginal for a system with low stiffness. Simultaneously, the decrease in stiffness causes a drop in maximum user grade even at perfect network conditions. A similar observation can be made about Figure 3.11(d-f) and Figure 3.10(d-f).

Inference 6: TIM generalizes to more TI applications with multiple types of interactions.

For both applications, a packet loss of 50% only causes a significant difference in user experience for high stiffness. This suggests that, at least for this class of applications, high reliability is not a priority.

From Figure 3.11(a) and Figure 3.11(b) we can see a cutoff region between a TIM score of 0.25 and 0.5, where the network starts significantly affecting the user grades, which matches the Bounce application. Note that the user study requires more statistical significance to provide accurate TIM thresholds for TI applications. However, the presented inferences are adequate to provide a good starting point to fine-tune a specific application for a seamless user experience.

Inference 7: The choice of channel compensation spring generalizes across different types of TI applications.

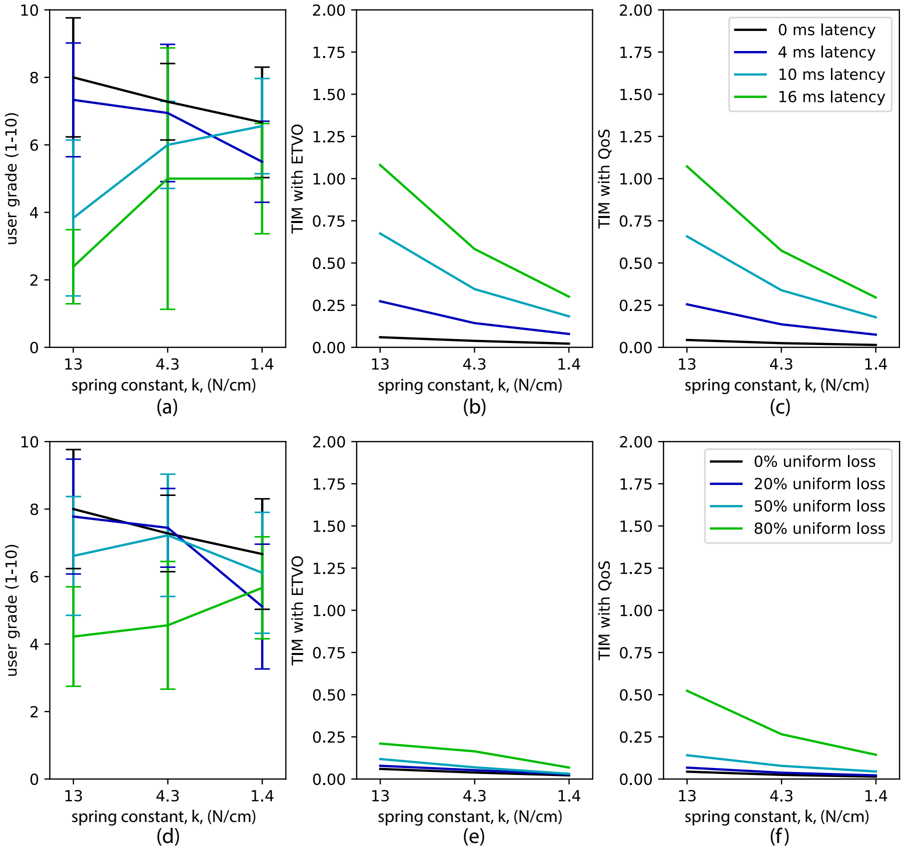


Figure 3.11: User grades and corresponding TIM scores across different network latency, packet loss, and perceptual Deadband settings by QoS and ETVO models for the Slide application.

The variety of performance evaluations presented in this work show that TIM can be used to gauge the real-time performance of the network in supporting TI applications. Further, the steps we followed in this work can be used for measuring the quality of different classes of TI applications. These insights can be used to better understand TI systems' performance, including when specialized solutions such as PD are deployed. This paves the way for a tailor-made network design for TI use-cases and allows accurate evaluation of novel solutions that are conventionally hard to quantify.

3.8. CONCLUSIONS

In this chapter, we proposed a real-time metric TIM to objectively evaluate the performance of TI sessions encompassing network parameters. Our metric is based on the dynamics of interactions with objects in conjunction with an approximated network model. The behaviour of a class of TI applications projected for the cases of both ideal and practical networks (non-zero latency and packet loss) and the difference in the trajectories was used to compute a relative norm, enabling us to evaluate the TI performance.

A novel mathematical model was developed to obtain a closed-form expression for the trajectories with varying delay, thereby allowing real-time computation of the TIM. We implemented two applications and conducted human subjective experiments on a real TI testbed. Through these human subjective experiments, we found a strong correlation between network settings, user grades, and the level of stiffness of the application. Additionally, we showed the ability of the proposed metric to indicate deterioration due to the network infirmities for a given application. We also devised a channel compensation spring that compensates for network variations measured by TIM. Several inferences were also discussed based on subjective measurements, which help in tuning the channel compensation spring. As TIM can be obtained in real-time, it opens up possibilities for better network resource management to facilitate TI applications.

This chapter provided insights into the latency requirements of Haptic Bilateral Teleoperation (HBT) and ways to manage the performance under varying network conditions. The next step is to explore how these findings can inform optimizations in network design to enhance the performance of HBT systems. This will be the focus of the next chapter.

4

MAC FOR TELEOPERATION[1]

4.1. INTRODUCTION

In this chapter, we address Sub-Question 3 as stated in Section 1.4: *How can we use insights from characterizing network performance to improve networks design for Haptic Bilateral Teleropation?* Haptic Bilateral Teleoperation (HBT) applications have extremely tight latency constraints. In Chapter 2 and Chapter 3 we proposed measuring techniques and a testbed to perform human subjective experiments for validation. In this chapter, we will use both of these contributions to improve the network and facilitate bilateral teleoperation applications.

From the experimental results provided in Chapter 2 and other supporting literature, it was found that a sub-5 ms of latency is highly desirable for tactile traffic [48]. Without such ultra-low latency (ULL) performance, the operators cannot teleoperate with the robot effectively, leading to catastrophic consequences, especially in mission-critical applications. The Tactile Internet (TI) standards [48] also recommend packet-level reliability of 99.9999% as a key requirement for seamless user interaction. However, this is only speculation, and no evidence-based substantiation of this requirement exists. On the contrary, the results in Chapter 2 and several independent studies reveal that the user experience decreases only marginally, even up to 50% packet loss when the packet rate is in the order of 1 kHz [34, 33, 49, 6]. Leveraging these insights is key to delivering high-quality performance for bilateral teleoperation applications.

Under a tight latency budget, performing optimizations at every segment of the network is crucial. This chapter focuses on first- and last-mile communications, where Wi-Fi emerges as the forefront runner due to its large-scale deployments in residential and IIoT applications. Wi-Fi is a significant latency bottleneck in shared networks, leading to high latency [50]. This will be a barrier to leveraging Wi-Fi for HBT applications, although IEEE 802.11 working groups aim to provide specific solutions to offer latency guarantees [51, 52].

[1] This chapter is based on the publication titled *"ViTaLS—A Novel Link-Layer Scheduling Framework for Tactile Internet Over Wi-Fi"*[47].

Understanding the Quality of Service (QoS) requirements for tactile and video modalities is crucial. The tactile modality demands ultra-low latency (ULL) but can tolerate packet losses of up to 50%, as previously discussed. In contrast, video feedback allows a higher latency budget (approximately 30 ms) but has a much stricter loss tolerance of just 2% [31, 30]. Meeting these requirements necessitates efficient transmission scheduling policies. This challenge is particularly pronounced in Wi-Fi networks, where channel access uncertainties and collisions can introduce significant, unpredictable delays, severely impacting the QoS performance of HBT.

Wi-Fi 6 introduces features like Orthogonal Frequency-Division Multiple Access (OFDMA) to improve latency performance, but its potential for optimizing HBT applications remains underutilized. Existing multiplexing schemes, like VH-multiplexer [10], combine tactile and video traffic into a single stream, limiting the ability to prioritize tactile traffic. Additionally, most approaches overlook the critical role of video feedback in HBT scenarios, leaving a gap in designing effective solutions.

To address these challenges, we explore a novel approach for optimizing HBT communication over Wi-Fi 6 systems. Our approach aims to leverage the capabilities of modern Wi-Fi, such as OFDMA and Access Categories, to prioritize tactile traffic while maintaining the stringent QoS requirements of video feedback.

CONTRIBUTIONS

The contributions in this chapter are listed below.

① By taking the example of a state-of-the-art multiplexing scheme (VH-multiplexer), we provide a detailed overview of HBT communication over Wi-Fi 6 and highlight the primary limitations of the system with respect to HBT communication (Section 4.3.1).

② We propose ViTaLS as a way toward HBT communication over Wi-Fi 6/7 networks. We describe the various ingredients of ViTaLS and provide the rationale behind our design choices (Section 4.3.2).

③ We develop a mathematical model for theoretically quantifying the working of ViTaLS (Section 4.3.4). Apart from providing a formal description of ViTaLS, our model validates the custom simulator used.

④ We also present a variant of ViTaLS – ViTaLS-optimal for optimizing the queue size at Wi-Fi devices. We demonstrate that ViTaLS-optimal outperforms the VH-multiplexer through extensive objective and subjective evaluations. This makes it a promising candidate for effective HBT communication over Wi-Fi 6/7 (Section 4.4).

⑤ We provide implementation notes to serve as guidelines for vendors/implementers to deploy ViTaLS-optimal on Wi-Fi 6/7 devices (Section 4.3.5).

4.2. RELATED WORK

This section provides an overview of existing literature on Wi-Fi advancements and their applications in HBT systems.

The classical 802.11e amendment, known as Enhanced Distributed Channel Access (EDCA) [53], introduced differentiated QoS support by defining Access Categories (ACs) for prioritizing real-time traffic. EDCA operates using a channel contention mechanism with random backoff. Over the years, several enhancements have been proposed to improve its efficiency [54, 55, 56]. More recently, Wi-Fi 6 has introduced Orthogonal Frequency-Division Multiple Access (OFDMA) [57], which significantly enhances latency performance. While these advancements offer a strong foundation, the potential of optimally leveraging ACs and OFDMA for HBT communication remains an open area of exploration.

In the TI domain, application-layer strategies have been proposed for multiplexing video and tactile traffic. For instance, VH-multiplexer [10] and Dynamic Packetization Module [12] merge video and tactile traffic into a single stream, which simplifies integration but limits the ability to prioritize tactile data. Alternatively, works like [58, 9] separate video and tactile streams to utilize different ACs. This approach effectively leverages Wi-Fi's inherent QoS features, although the increased contention among ACs can present challenges when relying solely on standard Wi-Fi scheduling.

Studies on HBT communication over Wi-Fi remain relatively limited. For instance, [59] evaluates tactile latency using Hybrid Coordination Function Channel Access (HCCA), a centralized mechanism where the Access Point (AP) manages channel access. While promising in theory, HCCA has seen limited adoption due to its complexity [60]. Similarly, approaches such as the FiWi network [61] introduce scheduling algorithms for TI but do not incorporate video feedback, which forms a significant portion of HBT traffic. Latency-loss tradeoff studies [62] offer valuable insights but do not fully address the complexities of realistic HBT scenarios.

While existing methods provide meaningful contributions to HBT over Wi-Fi, challenges remain, particularly in integrating video feedback and tactile data effectively. These gaps highlight the need for further research and tailored solutions to optimize HBT communication over modern Wi-Fi systems.

4.3. The Proposed ViTaLS Framework

In this section, we first provide an overview of the Wi-Fi 6 protocol and its applicability to HBT systems. We then identify opportunities for improvement within the protocol to better support HBT applications. Building on these insights, we introduce the architecture and design of the proposed *Visual-Tactile Latency Scheduler (ViTaLS)* framework, as illustrated in Figure 4.2.

4.3.1. An analysis of Wi-Fi 6 for HBT

Consider the Industry 4.0 use case of a connected factory with the communication inside the plant enabled by Wi-Fi 6.[2] Human operators control wireless, mobile robot arms inside the factory. Given the mission-critical nature of TI applications, assuming a tightly controlled Wi-Fi 6 network serves only TI traffic is reasonable. We consider a setup where a single Wi-Fi 6 AP serves a set of Wi-Fi 6 STAs. We take the VH-multiplexer [10] as the reference TI multiplexing scheme.

Video-Haptic (VH) multiplexer

The VH-multiplexer is designed to operate in the controlled domain where video and haptic traffic are generated. Let us assume standard frame rates for video and haptic

[2] Although this work is built on top of Wi-Fi 6 specifications, we expect that it can also contribute to building a TI profile for upcoming Wi-Fi 7.

streams of 60 Hz and 1 kHz, respectively. The VH-multiplexer splits each video frame into multiple fragments at the application layer. An application layer message consists of an augmented haptic frame and a video fragment (H+V in Figure 4.1), forming a MAC Protocol Data Unit (MPDU). This prevents the transmission of large video frames from holding up haptic frames while meeting the video latency budget. These messages are sent down the network stack, where the link-layer scheduling is managed by Wi-Fi 6.

Wi-Fi 6 communication

Channel access: We show the Enhanced Distributed Channel Access (EDCA) of Wi-Fi 6 in Figure 4.1. Compared to a slower AC, a faster AC has a smaller (i) contention window (CW) range and (ii) a smaller pre-defined interval known as arbitration interframe spacing (AIFS). When the channel is busy, a device backs off by selecting a random backoff (BO) counter uniformly from [0, CW-1], where CW$\in$[CW$_{\text{min}}$,CW$_{\text{max}}$]. When the channel becomes idle for AIFS, the BO countdown starts. BO is counted down every time the channel is idle for a pre-defined interval of *slot size* denoted as T_s. When BO reaches 0, the device transmits a packet. If packets from multiple devices collide (t_5), the CW is doubled until CW$_{\text{max}}$ is reached.

Tactile-video transmission: When a STA wins contention, it transmits H+V frames in single-user (SU) mode (t_1, t_2) on uplink (UL) occupying the entire bandwidth. On the other hand, when the AP wins contention, it could employ OFDMA if there are kinematic (K) frames for multiple STAs in its buffer (t_3). This is multiuser downlink (MU-DL) transmission. Further, the AP can also provision MU-UL transmissions.[3] In MU-UL, the scheduled STAs transmit H+V frames (t_4) in allocated portions of the channel.

To summarize, the UL transmissions happen when either a STA or the AP wins the contention, whereas DL transmissions only occur when the AP wins. Therefore, when the AP wins, it is important to first transmit the K frames as they would be in the buffer since the previous AP channel access. For TI applications, the AP must perform a MU-DL first and then provision a MU-UL. After the MU-UL, the channel contention is resumed depending on new frame arrivals. We adopt this strategy of a MU-UL following a MU-DL throughout this paper.

For MU transmissions, the channel is divided into blocks of subcarriers (tones), known as Resource Units (RUs). For example, an 80 MHz channel is made up of 996 tones and can be split into two 484-tone RUs, four 242-tone RUs, eight 106-tone RUs, and so on. We adopt the RU-allocation scheme proposed in [63] for maximizing the number of STAs scheduled during a MU access. Table 4.1 summarizes the RU allocation and number of scheduled STAs. For the MU-DL, the AP can look up its buffer to determine the amount of STAs with DL data. In the case of MU-UL, the information regarding the number of STAs with UL data and the scheduled ones are exchanged using Buffer Status Report (BSR) and trigger frames, respectively.

Shortcomings of VH-multiplexer with Wi-Fi 6

We highlight the two important shortcomings of VH multiplexer.

[3] Note that we consider only OFDMA-based multiuser transmission in this work and not MIMO-based multiuser transmission.

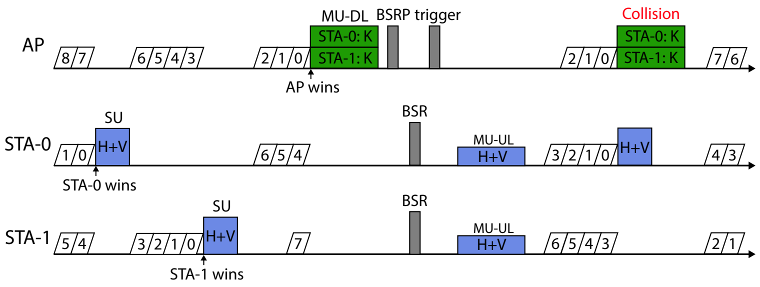


Figure 4.1: Timing diagram showing DL and UL transmission within the Wi-Fi 6 framework when using VH-multiplexer.

# STAs with data	1	2	3	4	5	6	7	≥8
# scheduled STAs	1	2	4	4	5	6	7	8
RU size (tones)	996	484	242	242	106	106	106	106

Table 4.1: Number of available and scheduled STAs in MU transmissions for 80 MHz bandwidth as per the RU-allocation scheme proposed in [63].

- Video-haptic augmentation leaves no scope to selectively transmit or drop (during congestion) frames belonging to a particular modality (haptic or video). Hence, VH-multiplexer fails to enable prioritized frame transmissions, which severely hampers QoS performance.
- Collision between augmented video-haptic frames results in a larger collision duration than when only haptic frames collide. In a collision-prone Wi-Fi network, this results in a considerable amount of wasted bandwidth.

We quantify the above claims objectively in Section 4.4.2. In order to overcome the above limitations, we propose the ViTaLS framework.

4.3.2. DESIGN

LEVERAGING WI-FI ACS

To enable transmission prioritization between tactile and video frames, we propose to leverage the different ACs of Wi-Fi. At the STAs, the haptic and video frames are assigned to AC_VO (fastest AC) and AC_VI (slower AC), respectively. This allows us to tune the scheduling mechanisms and other transmission parameters, such as retry limit and CW range, to suit the heterogeneous requirements of these modalities. However, this also poses a challenge. Each STA now has two independently contending ACs, potentially leading to higher collisions than single AC solutions [10, 12]. Although Wi-Fi offers virtual collision management between ACs within a device, AC_VI and AC_VO packets belonging to different devices can still collide. As described in Section 4.2, this can be worse than single AC solutions. To mitigate this issue, we propose to increase the CW range of AC_-VI significantly compared to that of AC_VO so that the video frames reduce their SU transmissions. Essentially, the idea is to reduce tactile-video frame collisions in a Wi-Fi 6 standards-compliant manner. At the AP, kinematic frames are enqueued in AC_VO as they

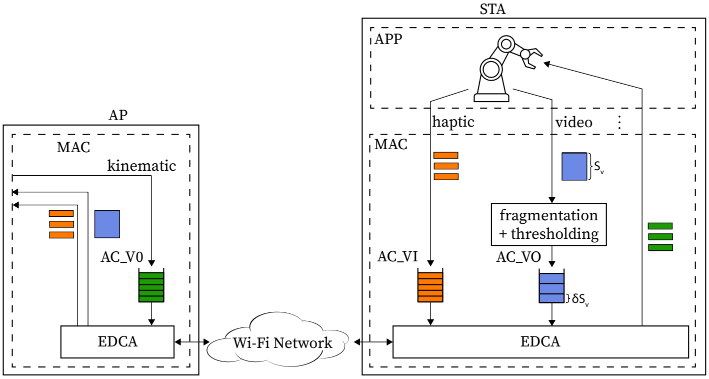


Figure 4.2: Schematic representation of ViTaLS framework depicting the different steps involved at MAC layer as well as the video-tactile data flow.

also require ULL guarantees. Haptic transmissions in SU mode are marked at t_2, t_3, and t_4 in Figure 4.3).

Scheduling video frames in MU-UL

While increasing the CW range of AC_VI favors tactile frames, it can potentially starve the video frames of channel resources, leading to video QoS violations. To address this, we leverage AP-initiated MU-UL transmissions for scheduling the video frames. The idea is to exploit the contention-free UL transmissions for scheduling high-reliability video frames. This implies that the video latency is predominantly dependent on AP channel accesses. Since the haptic and kinematic frames both use AC_VO, one can expect the AP to get channel access quite often, thereby benefiting video traffic. Further, collision-free video transmission also meets the high reliability requirement of the video stream. Moving video transmissions to the MU-UL is an important feature of ViTaLS since the collisions occur only between the tactile frames, which are typically small. This significantly reduces the wasted channel bandwidth in comparison with the VH-multiplexer.

Video fragmentation and thresholding

Transmitting a video frame as a whole results in a large MU-UL duration. As an example, consider 8 STAs, each employing MCS-9 and generating video traffic at 15 Mbps. A channel data rate of 400 Mbps results in a MU-UL duration of 6 ms. This causes significant hold-up of tactile frames, increasing their worst-case latency. To prevent this, we adopt the idea of video fragmentation from the VH-multiplexer and optimize it further for improved performance. Each video frame (of size S_v) is split into multiple fragments of size δS_v. Here, $\delta \leq 1$ denotes a parameter called *fragment threshold*. If video frames are available at the time of the MU-UL, the STA transmits a maximum of one fragment. Going back to our numerical example, $\delta = 0.33$ implies three fragments per frame, lowering the MU-UL duration to 2 ms. This reduces the worst-case tactile latency.

A small δ is favorable for containing the tactile latency, but it requires more MU-UL accesses per video frame transmission. In the case of a small number of STAs (denoted by

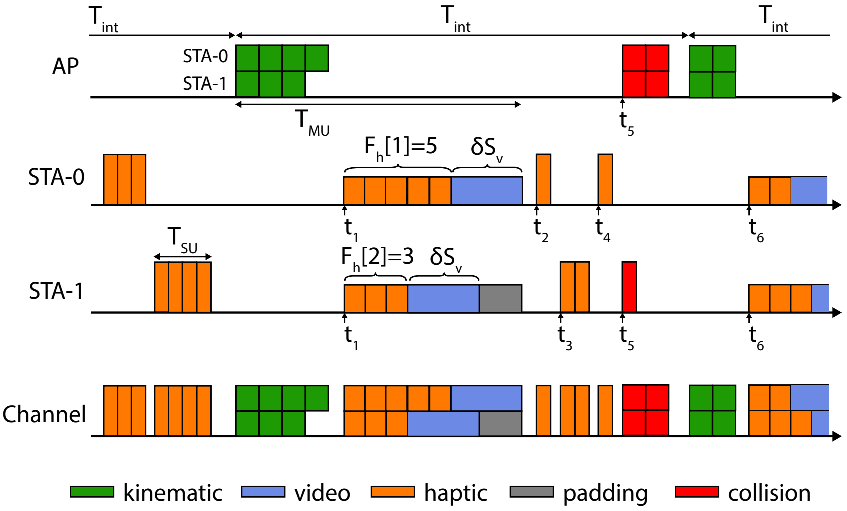


Figure 4.3: Timing diagram showing UL and DL transmissions as defined in ViTaLS framework from the standpoint of Wi-Fi devices and the channel.

N), a small δ suffices for meeting the video QoS requirements. However, higher N results in significant video latency. On the other hand, a large δ is favorable for video streams but is problematic for tactile streams. Therefore, an optimal choice of δ is important for seamless HBT interaction. We elaborate on the impact of δ on latency in Section 4.4.2.

Tactile queue sizing

As explained in Section 4.1, literature suggests that perceptual experience decreases only marginally even up to 30% tactile losses [34, 33, 49, 6]. This insight allows us to maintain a good user experience even during high load conditions. Note that the video losses should be below 2% for smooth HBT experience [31, 30]. An efficient way to achieve this is by limiting the tactile queue size (denoted by Q) at the MAC layer. When the queue is full, the older tactile frames are considered outdated and are dropped to make room for newer ones ensuring that the latest tactile information is kept intact. This puts an upper bound on the queuing latency (at the expense of loss). It is important to state the difference in Q at STAs and the AP, denoted by Q_{sta} and Q_{ap}, respectively. Q_{sta} is the maximum permissible haptic frames in the queue. Q_{ap} is the maximum permissible kinematic frames per STA. Using Q_{sta} and Q_{ap} as design parameters, we demonstrate their impact on the overall performance in Section 4.4.2.

Heterogeneous payload

Since MU-UL provides collision-free channel access to the STAs, it is beneficial to leverage MU-UL for tactile frame transmissions, when possible, without necessitating any control overhead. Prior to Wi-Fi 6, the Wi-Fi systems allowed only MPDUs belonging to the same AC to be aggregated in a packet. This was amended in Wi-Fi 6 with *Multi-Traffic Identifier Aggregated MPDU (multi-TID AMPDU)* where heterogeneous MPDUs can also be aggregated. As per multi-TID AMPDU, when a particular AC is scheduled for transmission, even

Algorithm 1 ViTaLS algorithm at STA

```
if haptic buffer is full then
    Drop oldest frame upon new frame arrival
end if
if STA-$l$ wins contention then
    Transmit $F_h[l]$ haptic data or $F_v[l]$ video data
end if
if STA[$l$] is scheduled in MU-UL then
    if both buffers are non-empty then
        Transmit multi-TID AMPDU with $F_h[l]$ haptic
        data and $F_v[l]$ video data
    else
        Transmit $F_h[l]$ haptic data or $F_v[l]$ video data
    end if
end if
```

MPDUs belonging to higher priority ACs can be aggregated. We leverage this feature in ViTaLS to piggyback haptic frames when video frames are scheduled in MU-UL (t_1 and t_6 in Figure 4.3). Note that in MU-UL transmissions, the video frames are sent only after the tactile frames, as shown in Figure 4.3. This greatly benefits the tactile latency. Further, as per Wi-Fi 6 standards, padding bits are added to synchronize MU-UL transmission across the STAs. When the video buffer is empty, only haptic frames are transmitted in MU-UL.

4.3.3. ViTaLS algorithm

With the above framework, we will now describe the ViTaLS scheduling algorithm at both STA (Algorithm 1) and AP (Algorithm 2). Let $F_h[l]$ and $F_k[l]$ denote the amount of buffered haptic and kinematic data (in bytes) belonging to STA-l, respectively.

SU transmissions: When STA-l wins the channel contention, it sends a packet comprising of either $F_h[l]$ tactile data or a video fragment depending on the winning AC. We expect negligible video transmission in SU mode as the AP wins the channel contention much more quickly then the video streams.

MU transmissions: When the AP wins the channel contention, it can, in principle, schedule up to N STAs for DL transmission depending on queued up data. Since we are employing the RU allocation proposed in [63], MU-DL transmission can accommodate up to 8 STAs based on the amount of DL data per STA. After MU-DL transmission, the AP seeks the *maximum permissible UL data* from each STA using BSRP. Let the maximum permissible video data of STA-l be denoted by $F_v[l]$ (in bytes). This is the minimum between δS_v and the video queue occupancy. Let $F_h[l]$ denote the haptic data counterpart. With this information, the AP schedules up to 8 STAs based on $F_h[l]$+$F_v[l]$. The MU-UL duration is computed as the transmission time for the STA with the highest $F_h[l]$+$F_v[l]$ and is dependent on the RU allocated for that STA and MCS used. MU-UL duration (in the form of PHY layer field *L-SIG length*) along with the RU allocation are then communicated to all STAs using the *trigger frame*. This is followed by the MU-UL transmission of multi-TID AMPDUs or haptic AMPDUs.

Algorithm 2 ViTaLS algorithm at AP

```
if queue has $Q_{\text{ap}}$ kinematic frames for STA-$l$ then
    Drop oldest frame for STA-$l$ upon new frame arrival
end if
if AP wins contention then
    Schedule STAs with highest DL data
    Send kinematic AMPDUs on allocated RUs
    if STAs have UL data then
        Compute MU-UL duration using UL data and RUs
        Schedule STAs with highest UL data for MU-UL
    end if
end if
```

4.3.4. MATHEMATICAL MODEL

We present an analytical model for theoretical estimation of the performance of ViTaLS. For the ease of analysis, we make the following reasonable assumptions. ❶ The probability of AC_VI winning the channel contention is negligible. ❷ The CW range and the number of backoff stages of AC_VO at AP and STAs are identical. ❸ There are no collisions during MU-UL as the AP broadcasts the MU-UL schedule to all STAs. ❹ All the tactile frames in the device queue are transmitted when it gets a channel access. ❺ There are no legacy (pre Wi-Fi 6) devices connected to the AP as HBT communication necessitates a tightly controlled network.

The seminal work of Bianchi [64] provides an accurate model for the throughput performance of Wi-Fi. Many later works followed up on Bianchi's work to model the latency performance of Wi-Fi [65, 66]. The work in [67] estimates throughput for OFDMA-based Wi-Fi 6 systems. Based on a per-slot analysis, the above works show that τ – packet transmission probability of a device in a slot is a constant that is dependent only on the CW parameters. As per these works, if the CW parameters of the AP and STAs are identical, they have equal τ. It is important to note that this holds good only if any of the following conditions are satisfied. ❶ Like legacy Wi-Fi systems, there is no AP-initiated UL transmission [64, 65, 66], ❷ the STAs do not reset their backoff counters after AP-initiated UL transmissions for further channel contention, as is implicitly assumed in [67, 63]. In ViTaLS, although the CW parameters of AP and STA AC_VO are identical (assumption ❷), none of the above conditions is satisfied. While condition ❶ does not hold as ViTaLS relies heavily on MU-UL transmissions, condition ❷ fails since the haptic queue of a STA is completely emptied during MU-UL transmissions (assumption ❹) leading to resetting the backoff counters. Hence, these models are not directly applicable in our case. However, a few intermediate results are useful, as we will see in the rest of this section. Therefore, capturing the above intricacies of ViTaLS requires a major departure from existing works. We take up this non-trivial exercise in the following section.

Characterizing transmissions: As explained previously, in ViTaLS the AC_VO backoff counters at the devices are reset every time AP gains channel access. To make the analysis concrete, we view the temporal axis as a continuous series of time durations between the start of consecutive, successful AP transmissions, which we call "*intervening time*".

This is denoted as T_{int} in Figure 4.3. Note that within T_{int}, there can be multiple SU transmissions, denoted by x. This includes collided as well as successful ones. To begin with, let us consider the number of transmissions from a given STA in T_{int}. For ease of analysis, we ignore the binary exponential nature of the backoff process. x can be interpreted as the number of independent backoff choices such that their cumulative sum is smaller than AP's backoff choice. We use the fact that the PDF of the sum of independent random variables is the convolution of their individual PDFs. Without loss of generality, we can think of the devices as picking a real backoff value uniformly with the distribution $f(x) = 1$, if $x \in [0,1]$, and 0 otherwise. This gives us the probability of at least n transmissions by the STA as

$$P(x \geq n) = \int_0^1 \Big(f(x) * \overset{n \text{ convolutions}}{\cdots} * f(x) \Big)\, dx = 1/(n+1)!$$

The limits of the integral denote the range of AP backoff values on the new scale. From first principles, the probability of exactly n transmissions by the STA can then be derived as

$$P(n) = P(x \geq n) - P(x \geq n+1) = (n+1)/(n+2)!$$

We can now calculate the expected number of SU transmissions per STA in T_{int} as

$$E[n] = \sum_{n=0}^{\infty} nP(n) = e - 2 \approx 0.71828. \tag{4.1}$$

This means that for every successful AP transmission, each STA transmits a mean of 0.71 packets independent of the amount of contending devices. This reveals that τ of the AP and STAs are non-identical in ViTaLS. This important finding is a significant departure from the existing works and forms the basis of our mathematical model.
Collision model: The work in [67] derives τ as

$$\tau = \left[\frac{(1 - P_c - P_c(2P_c)^m)(\text{CW}_{\min} + 1)}{2(1 - 2P_c)} + \frac{1}{2} \right]^{-1}, \tag{4.2}$$

where m is the retry limit, P_c is the collision probability of a transmitted packet. We will append the notations used so far with subscripts 'ap' and 'sta' to denote the specific parameters of AP and STA, respectively. Due to the asymmetric nature of transmissions between AP and STA derived in Equation (4.1), we can obtain the respective transmission probabilities as

$$\tau_{\text{ap}} = \tau, \text{ and } \tau_{\text{sta}} = \alpha\tau, \tag{4.3}$$

where $\alpha = E[n]$. Based on τ_{sta} and τ_{ap}, the collision probabilities of AP and STA can be expressed as

$$P_{\text{c,ap}} = 1 - (1 - \tau_{\text{sta}})^N, \tag{4.4}$$

$$P_{\text{c,sta}} = 1 - (1 - \tau_{\text{ap}})(1 - \tau_{\text{sta}})^{N-1}. \tag{4.5}$$

The closed form expressions for $P_{\text{c,ap}}$ and $P_{\text{c,sta}}$ can be obtained by solving Eqs. (4.2)-(4.5).

Due to packet retransmissions, one can think of collisions as resulting in additional data to transmit from the standpoint of the network. Further, the collisions result in bigger packets due to MPDU aggregation. Therefore, the overall data rate scales by a factor of $1/(1-P_c)$. Due to assumption ❸, the AP collisions result only in kinematic frame retransmissions. Further, MU-UL transmissions also involve padding due to the unequal haptic frames at the STAs. Essentially, the amount of padding is determined by the STA with the highest haptic queue occupancy at the time of MU-UL transmission. As an upper bound, every successful SU transmission by a STA would create a padding frame equal to the amount of transmitted haptic data. Hence, the data rates of MU and per-STA SU transmissions can be respectively expressed as

$$\begin{aligned}
D_{\mathrm{MU}} &= N\left[\left(\delta S_v + H\right) f_v + (S_h + H) f_h + \frac{(S_k + H) f_k}{1 - P_{\mathrm{c,ap}}}\right],\\
D_{\mathrm{SU}} &= \left(\frac{\alpha}{1+\alpha}\right)\frac{(S_h + H) f_h}{(1 - P_{\mathrm{c,sta}})},
\end{aligned}$$

where H denotes the header overhead per MPDU, $\alpha/(1+\alpha)$ is the ratio of haptic data transmitted by a STA to that by the AP. f_h, f_k, and f_v denote the frame rates of haptic, kinematic, and video traffic.

Fine-grained timings: Denoting channel bandwidth as B, the mean duration of each MU and SU transmission can be respectively expressed as

$$T_{\mathrm{MU}} = T^e_{\mathrm{MU}} + \frac{D_{\mathrm{MU}} T_{\mathrm{int}}}{B}, \qquad T_{\mathrm{SU}} = T^e_{\mathrm{SU}} + \frac{D_{\mathrm{SU}} T_{\mathrm{int}}}{\alpha B}, \tag{4.6}$$

where T^e_{MU} and T^e_{SU} denote the "*extra time*" per MU and SU transmission, respectively, due to control signals (TF, BSR, BSRP, etc.), PHY layer header, and other overheads (SIFS, AIFS, etc.). The expected backoff duration can be given as

$$T_b = \frac{\mathrm{CW}_{\mathrm{min}} T_{\mathrm{s}}}{2}\left[\frac{1 - (2P_{\mathrm{c,ap}})^m}{1 - 2P_{\mathrm{c,ap}}}\right]. \tag{4.7}$$

We can now express T_{int} as

$$T_{\mathrm{int}} = T_b + \alpha N T_{\mathrm{SU}} + T_{\mathrm{MU}}. \tag{4.8}$$

Substituting Eqs. (4.7) and (4.8) in Equation (4.6), we obtain the simultaneous equations

$$\begin{aligned}
-\alpha N T_{\mathrm{SU}} + \left(\frac{B}{D_{\mathrm{MU}}} - 1\right) T_{\mathrm{MU}} &= \frac{B}{D_{\mathrm{MU}}} T^e_{\mathrm{MU}} + T_b,\\
\left(\frac{B}{D_{\mathrm{SU}}} - N\right) T_{\mathrm{SU}} - \frac{T_{\mathrm{MU}}}{\alpha} &= \frac{B}{D_{\mathrm{SU}}} T^e_{\mathrm{SU}} + \frac{T_b}{\alpha}.
\end{aligned} \tag{4.9}$$

The above equations can be solved to obtain closed form expressions for T_{MU} and T_{SU} in terms of parameters of Wi-Fi and video-tactile traffic.

The parameters T_{MU} and T_{SU} significantly affect the latency performance of ViTaLS. Modeling video-tactile as a function of the above parameters is non-trivial and requires further analysis. This forms a part of our future work. Hence, in Section 4.4 we validate the estimated T_{MU} and T_{SU} through simulations.

Parameter	Value	Parameter	Value
AC_VO [$\text{CW}_{\text{min}}, \text{CW}_{\text{max}}$]	[32, 64]	MCS	9
AC_VI [$\text{CW}_{\text{min}}, \text{CW}_{\text{max}}$]	[512, 2048]	AIFS	34 μs
AC_VO Retry limit	4	SIFS	16 μs
AC_VI Retry limit	10	RTS, CTS	44 μs
Max. PPDU duration	5.4 ms	Guard Interval	0.8 μs
Block ACK	44 μs	Slot size	9 μs
BSRP, BSR, trigger	44 μs	Aggregation	MPDU

Table 4.2: Wi-Fi 6 configuration parameters used in our simulations.

4.3.5. Implementation notes

ViTaLS is compliant with Wi-Fi 6 standards except for some minor modifications at the link layer that we discuss here. Firstly, the CW range of AC_VI should be configured to a much larger value than that of AC_VO. Under tightly controlled Wi-Fi networks allowing only HBT traffic, this will not increase the video latency proportionately as the MU-UL access will satisfy the necessary video QoS. Secondly, video frames at STAs should be fragmented as per the pre-defined δ before forwarding to the physical layer. Within BSRs, the STAs must communicate to AP the amount of queued haptic frames and the permissible video size (based on δ) instead of the entire video buffer occupancy. Lastly, the MAC queues should adopt a *head-drop* scheme. This ensures that earlier frames are treated as outdated when newer ones arrive. The proposed updates to Wi-Fi 6 link layer can also serve as a basis for developing the HBT operation profile for Wi-Fi 7.

4.4. Performance Evaluation

In this section, we first describe the experimental setup for the objective and subjective evaluation of ViTaLS and then present our important findings.

4.4.1. Experimental Setup

Objective evaluation

For objective evaluation of ViTaLS, we developed a custom Wi-Fi MAC simulator written in C++. To facilitate rapid developments in the field of HBT over Wi-Fi, we have open-sourced our simulator.[4] In our simulations, we use a fixed modulation and coding scheme – MCS-9. This removes the impact of rate adaptation, enabling us to measure the performance improvement solely due to ViTaLS. This is a common approach in literature [52]. For this work, we choose a channel bandwidth of 80 MHz in the 5 GHz spectrum. The typical Wi-Fi 6 parameters are set as shown in Table 4.2.

The tactile traffic is generated at the standard rate of 1 kHz. Each kinematic and haptic frame is 480 B and 240 B, respectively, amounting to 5.8 Mbps of tactile traffic. This accounts for the sensors on the tactile glove and the robot arm. On the other hand, video frames, each of size 30 kB, are generated at 60 Hz. This corresponds to realtime 4K video or VR traffic. Accounting for the packet header overheads, we obtain an overall traffic of

[4] Wi-Fi 6 MAC simulator - `https://github.com/VinGok/Tactile-WiFi`

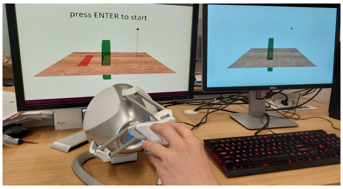
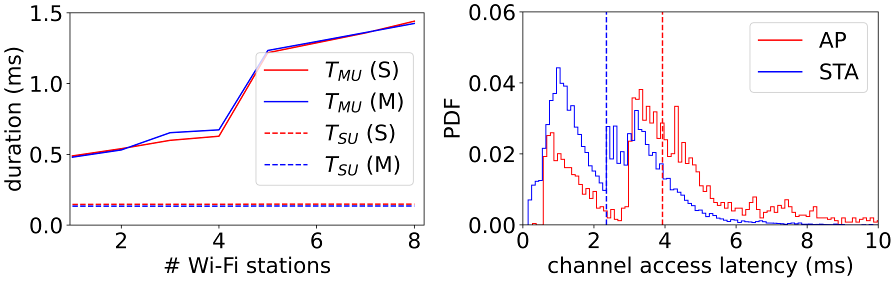


Figure 4.4: Virtual environment setup showing the haptic device with the video feedback in operator domain (left) and actual scene in the remote domain (right).



(a)

(b)

Figure 4.5: (a) Comparison of transmission durations (T_{MU} and T_{SU}) for ViTaLS, based on both our model (M) and simulations (S). (b) Asymmetric channel interaccess latency between the AP and STAs, along with their respective mean values.

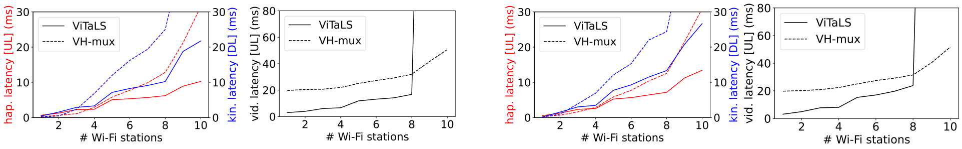
roughly 25 Mbps per operator-teleoperator pair. We take the 95th percentile latency as the worst-case latency measurement.

Subjective evaluation

For our subjective experiments, we deploy the testbed used and described Chapter 2. Since deploying custom MAC algorithms at the kernel level on real Wi-Fi devices is challenging, we leverage NetEm – a standard network emulator for network latency and packet losses. We incorporate the latency and loss characteristics obtained from the simulations in the emulator. This setup provides an easy way to assess the subjective quality of ViTaLS and the VH-multiplexer.

We leverage the same experimental setup used in Chapter 2. The task for the participant is to interact with a few VE objects and move them to a pre-determined target location, as shown in Figure 4.4. The VE runs on a remote workstation (right-side display shown for illustration) and supplies haptic and visual feedback to the operator (left-side display).

The subjective study involved 20 participants in the age group between 17 and 53 years, with an average of 25 years. Roughly half of the participants were novice users of the haptic device. Each participant interacts with the VE with the three schemes – VH-multiplexer, ViTaLS (large buffer), and ViTaLS (optimal buffer), separately enabled. The participants grade their HBT experience on a scale of 10 as follows:
10: no perceivable impairment; **8-9**: slight impairment but no disturbance; **6-7**: per-



(a) (b) (c) (d)

Figure 4.6: Comparison of latency profiles between ViTaLS and VH-multiplexer under different communication modes. (a) tactile in basic mode, (b) video in basic mode, (c) tactile in RTS/CTS mode, and (d) video in RTS/CTS mode.

ceivable impairment, slight disturbance; **4-5**: significant impairment, disturbing; **1-2**: extremely disturbing.

4.4.2. Results

Objective evaluation

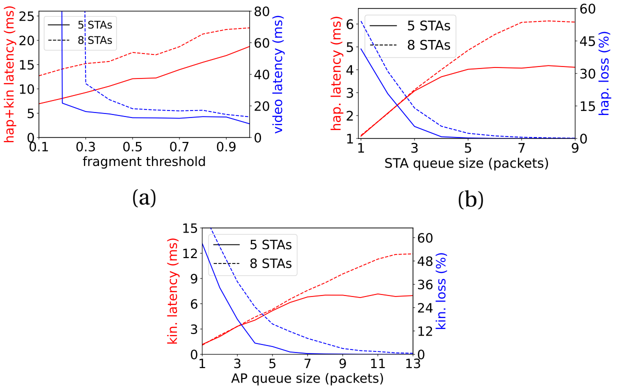
Unless mentioned otherwise, we empirically choose $\delta = 0.33$ in our simulations.

Model validation: We begin by validating our mathematical model in Figure 4.5(a). It can be seen that the estimations given by the model (M) for both T_{MU} and T_{SU} corroborate very well with the simulation measurements (S). T_{MU} increases monotonically with the amount of STAs as the load increases proportionately. On the other hand, T_{SU} remains agnostic to the load. This is because the SU transmissions occupy the entire channel bandwidth, which is significantly high in Wi-Fi 6/7 networks.

Latency measurements: We plot the PDF of the channel interaccess latency for AP and STAs along with their mean values in Figure 4.5(b) for $N = 8$ and $Q_{\text{ap}} = Q_{\text{sta}} = 50$. Note that the STA interaccess latency includes both SU and MU channel accesses. As expected, each STA gets channel access more frequently than the AP. This is because of the AP-initiated MU-UL transmissions. This implies that the haptic frames encounter significantly lower latency than the kinematic frames.[5] This is an important observation and will be utilized later in this section for optimal tactile queue sizing.

We now present the worst-case latency performance of ViTaLS and VH-multiplexer over a range of N. In Figure 4.6(a) and 4.6(b), we show the tactile and video latency profiles, respectively, for basic mode (without RTS/CTS exchange). As can be seen, ViTaLS comprehensively results in significantly lower latency overall with a peak reduction of up to 47% in the *two-way latency*, which is the sum of haptic and kinematic latency. Up to $N = 3$, where the amount of collisions is negligible, the tactile latency of ViTaLS and VH-multiplexer are comparable. However, the video latency of ViTaLS is significantly lower. For $\delta = 0.33$, three MU-UL channel accesses are required to transmit a video frame. On the other hand, the VH-multiplexer transmits a video frame over ~17 haptic frames (video frames are generated at 17 ms intervals), and thereby takes much longer. On the other hand, it can be seen that beyond $N = 8$, the video latency of ViTaLS increases drastically. This is primarily because the chosen δ cannot match the video generation and transmission rates. A higher δ is favorable at higher network loads. On the contrary,

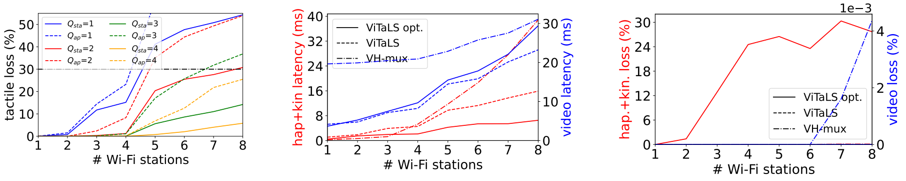
[5] To reiterate, the haptic and kinematic frames are transmitted on UL and DL, respectively.



(a) (b)

(c)

Figure 4.7: (a) Evaluation of ViTaLS depending on the system parameters. Impact of (a) fragment threshold (δ) on video-tactile latency, (b) STA queue size (Q_{sta}) on haptic latency and loss, and (c) AP queue size (Q_{ap}) on kinematic latency and loss.



(a) (b) (c)

Figure 4.8: (a) Tactile loss for different AP and STA queue sizes along with the loss threshold of 30%, (b) latency characteristics and (c) loss characteristics of ViTaLS, ViTaLS-optimal, and VH-multiplexer.

the video latency of the VH-multiplexer is still contained, as every SU transmission also carries video fragments.

As explained in Section 4.3.1, one of the reasons for the high latency of VH-multiplexer is the significant collision duration. A standard method to reduce collision duration is to use RTS/CTS. To understand if RTS/CTS improves the performance of the VH-multiplexer, we present the latency performance in the presence of RTS/CTS with an RTS threshold of 1 kB. No performance improvement is seen in both tactile and video latency, as RTS/CTS is known to be effective only when N is substantially higher [68]. In the remainder of the paper, we focus on the performance of ViTaLS up to $N = 8$ in basic communication mode.

Impact of δ: In Figure 4.7(a), we present the impact of δ on the latency characteristics of ViTaLS by varying δ in the range [0.1,1]. As can be seen, the two-way latency is a monotonically increasing function of δ since larger video fragments negatively impact the worst-case tactile latency. On the other hand, the video latency is a decreasing function of δ. Further, the minimum δ for meeting the video QoS latency increases with N due to less frequent MU-UL channel access. It is important to note the trade-off between tactile and video latency, as explained in Section 4.3.2. Further, given N, the two-way latency varies significantly over the sweep of δ. This suggests that choosing the optimal δ is crucial for a smooth HBT experience.

Impact of Q_{sta} and Q_{ap}: Naturally, higher Q_{sta} and Q_{ap} result in higher tactile latency and lower loss. This can be seen from Figure 4.7(b) and 4.7(c), respectively. For $N = 5$, the loss reaches 0 % at $Q_{sta} = 4$, at which point the haptic latency saturates at around ~4 ms. This implies that at most four haptic frames are queued up at the STAs, and higher Q_{sta} would overprovision the queue. Hence, $Q_{sta} = 4$ is sufficient to transmit all haptic frames without dropping any. Due to the higher channel inter-access latency at AP, Q_{ap} to achieve 0% loss is higher for the same value of N. Note the difference in scale of the latency axis in Figure 4.7(b) and 4.7(c). Further, the minimum Q_{sta} and Q_{ap} for achieving 0 % loss increase with N due to higher collisions and more queueing. These insights suggest that there is a large scope for controlling the queue sizes to trade-off loss for further improving the latency performance of ViTaLS.

In order to understand the optimal Q_{sta} and Q_{ap}, we present the tactile losses in Figure 4.8(a). For $N = 1$ and 2, even a small queue size of 1 frame results in no loss. For higher N, it can be seen that for the same queue size, the AP drops more frames than STAs. This is due to the asymmetric AP and STA channel interaccess behaviors (explained earlier in Figure 4.5(b)). Therefore, to achieve a target tactile loss one needs to employ different Q_{sta} and Q_{ap}, and further tune it depending on N for optimal performance (30 % loss target). For instance, $Q_{ap} = Q_{ap} = 1$ for $N = 4$, whereas $Q_{sta} = 2$ and $Q_{ap} = 4$ for $N = 8$.

ViTaLS-optimal: With the above insights, we tune Q_{sta} and Q_{ap} for each setting of N. We call this version of ViTaLS as "*ViTaLS-optimal*". We now compare the performances of ViTaLS, ViTaLS-optimal, and VH-multiplexer. In Figure 4.8(b), it can be seen that ViTaLS optimal yields a reduction of up to 82% in two-way latency compared to VH-multiplexer. The advantage of exploiting the loss threshold is clearly reflected in the latency improvement. Further, the video latency of ViTaLS-optimal also improves since dropping tactile frames reduces the network load. As seen in Figure 4.8(c), the tactile loss in case of ViTaLS-optimal reaches up to 30 %. The non-monotonic loss behavior is because the optimal Q is fine-tuned depending on N. The video loss for all schemes is negligible.

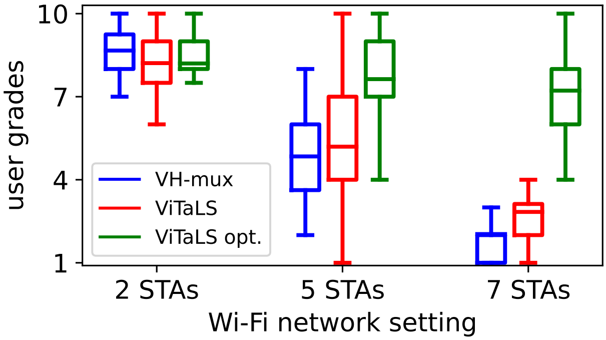


Figure 4.9: User grades confirming that ViTaLS optimal outperforms both VH-multiplexer and ViTaLS over different Wi-Fi network conditions.

While the objective performance gains of ViTaLS and ViTaLS-optimal are evident, it is also crucial to assess their quality of subjective experience to understand the perceptual artifacts that may be introduced. To this end, we now move to the subjective evaluation.

SUBJECTIVE EVALUATION

In Figure 4.9, we present the user grades for the cases of VH-multiplexer, ViTaLS, and ViTaLS-optimal under three network conditions: $N = 2, 5$, and 7. At $N = 2$, the performances of the three methods are comparable due to low collisions. At higher N, the users experience a significant disturbance with both VH-multiplexer and ViTaLS, although ViTaLS provides much better objective performance – a two-way latency of 27.5 ms with VH-multiplexer versus 17.2 ms with ViTaLS for $N = 7$. The reason for similar subjective performance between VH-multiplexer and ViTaLS despite the objective improvement is that the above latency numbers exceed the ULL budget by a significant margin. On the other hand, ViTaLS-optimal provides a significantly higher subjective performance despite the high network load due to its ability to dynamically drop frames without causing any perceptual degradation. This further substantiates the efficacy of our proposed framework.

4.5. CONCLUSIONS

This chapter investigated the less-explored problem of TI communication over Wi-Fi 6 networks. We showed conceptually and through experiments that the state-of-the-art scheduling schemes in TI fall short of satisfying the ULL requirement. To bridge this gap, we designed ViTaLS – a novel latency scheduling framework for TI. We present the ingredients of ViTaLS and provide the rationale behind our design choices, which are firmly based on the results obtained in Chapter 2. Taking VH-multiplexer, a state-of-the-art multiplexing scheme, as the baseline, we showed that ViTaLS reduces the tactile latency by about 47 %. Further, we present ViTaLS-optimal as an enhanced version of ViTaLS that employs optimal queue sizes leading to 82 % latency improvement over VH-multiplexer. Using a realistic TI testbed encompassing haptic devices and a network, we demonstrated that ViTaLS-optimal maintains a high-quality user experience even under high load conditions. At the same time, the performance of the VH-multiplexer deteriorates significantly. The proposed framework can be a strong candidate for making Wi-Fi 6 fit for TI communication. Further, ViTaLS-optimal can also be used to create a TI operation profile for Wi-Fi 7 systems.

In this chapter, we utilized insights from characterizing network performance in HBT systems to enhance network performance. However, network improvements alone are insufficient to fully address the challenge of realizing HBT. In the next chapter, we take a more drastic approach, shifting the focus from meeting performance requirements to improving the operators perception of the system performance.

5

Opportunities due to limits in operator perception[1]

5.1. Introduction

In this chapter, we address Sub-Question 4 as stated in Section 1.4: *How can we leverage knowledge about human perception to improve the user experience?* In the previous chapters, we have primarily focused on the performance of networks that facilitate a bilateral teleoperation application. Contributions have stemmed from understanding, characterizing, testing, and exploiting teleoperation performance. However, in order to make bilateral teleoperation over long distances work, there is a clear need for bolder solutions. Fortunately, there exist more areas where methods can be found to boost performance further.

In literature, there is a heavy reliance on objective metrics as the key performance indicators (KPIs), while subjective evaluations (user grades) are, most often, used only for additional validation of the objective results. An overview of the literature is provided in Section 2.2. The rationale for relying on objective metrics is reasonable and, to a large extent, justified as objective studies are controllable and repeatable and work well for humans in open-loop systems. However, some crucial limitations surface when we work with human-in-the-loop TI systems. In Chapter 3, a link was drawn between network delay and the dynamics of the underlying application. While the results were verified with human subjective experiments, the contributions were founded on a theoretical objective basis.

A fundamental goal of any system facilitating bilateral teleoperation with a human operator is to ensure a positive user experience. Achieving this goal presents both challenges and opportunities, particularly when considering the complex dynamics of human perception and system performance.

[1] This chapter is based on the publication titled *"Blind spots of objective measures: Exploiting imperceivable errors for immersive, tactile internet"*[69].

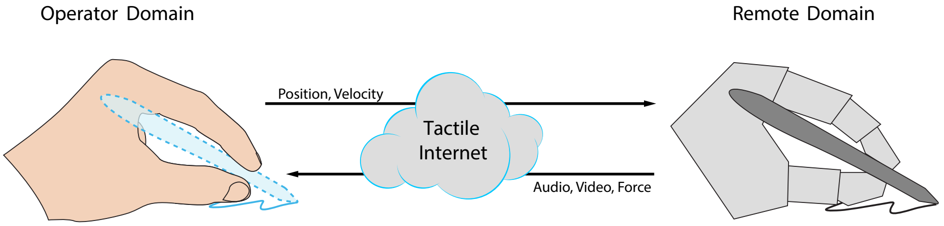


Figure 5.1: Depiction of a typical Tactile Internet (TI) system highlighting the master and controlled domains and data communication between them. In light blue, the pen is indicated in the master domain to be present only through haptic and visual feedback, while the real pen is only present in the controlled domain.

One significant challenge is that existing works often overlook the role of human perception in evaluating errors. Rather than considering how different types of errors may affect users, they treat all errors uniformly. In some cases, efforts to minimize errors by incorporating new information have inadvertently introduced perceptual artifacts, where the correction itself proves more detrimental than the original error. These artifacts can significantly degrade performance and pose serious risks, particularly in safety-critical Tactile Internet (TI) applications, where seamless and reliable interaction is crucial.

On the other hand, humans may be insensitive to certain types of errors, offering an opportunity to enhance TI performance by leveraging these imperceptible errors while easing stringent Ultra-Reliable Low-Latency Communication (URLLC) constraints. Achieving this requires deeper insights into human perception to design user-centric TI solutions.

These challenges and opportunities raise critical questions: How do different types of errors impact user experience? Can imperceptible errors be identified and exploited to improve TI communication? And are existing objective metrics sufficient to capture these effects and provide a comprehensive characterization of overall performance?

Addressing these questions opens the door to new methodologies for characterizing and developing systems that effectively support TI communication, offering both practical improvements and theoretical advancements. To this end, we take a user-centric approach, analyzing the impact of specific error types through targeted studies. Building on these insights, we introduce the *Adaptive Offset Framework (AOF)*, a novel signal reconstruction technique designed to intelligently handle errors in TI systems and enhance overall performance.

Contributions

The contributions in this chapter are listed below.

① We examine common errors in TI scenarios and introduce the concept of '*perceivability of errors*' to quantify their perceptual significance (Section 5.2).

② Based on these insights, we propose the *Adaptive Offset Framework (AOF)*, which dynamically adjusts the position offset between master and controlled domains to produce a smooth reconstruction signal without perceptual impairments (Section 5.3).

③ We implement AOF in a realistic TI setup and evaluate its performance. While objective metrics suggest underperformance, subjective measurements show a significant

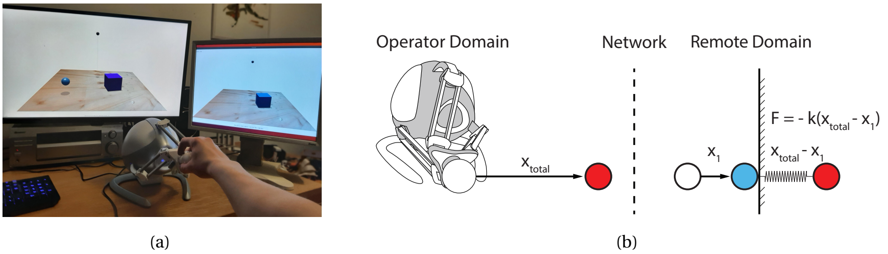


Figure 5.2: (a) TI setup in our lab showing the the user interacting with a virtual environment. The monitor on the left and right correspond to operator and remote domains, respectively. (b) Conceptual illustration of generation of haptic feedback. HIP is the white circle in the remote domain.

improvement in user experience (Section 5.5).

④ The contrast between subjective and objective metrics highlights the limitations of objective measures and demonstrates AOF's potential to improve overall TI performance (Section 5.5).

5.2. ERRORS IN TI AND THEIR PERCEIVABILITY

Several types of mismatch (*error*) can exist while reproducing a sensed signal in a TI system that could heavily influence the performance. To interpret these errors, we introduce the notion of *decomposition of errors* in a TI system. We consider the most common TI errors between operator and remote domains and examine their impact on user performance.

5.2.1. TYPICAL TI SETUP

We consider a typical TI setup as described in Chapter 2. A haptic device resides in the operator domain. The teleoperator, a robotic arm, resides in the remote domain in a remote physical environment. We use a Novint Falcon in the operator domain and a virtual environment (VE) in the remote domain for our experiments. The TI setup in our laboratory is shown in Figure 5.2(a). When a point on the haptic device is moved (from white to red location), the corresponding part of the teleoperator in the VE, known as *haptic interaction point (HIP)*, moves accordingly [70]. If a rigid object in the VE is at a distance of x_1 from the HIP, then the HIP applies a force F proportional to penetration depth $(x_{total} - x_1)$, when the device displacement is x_{total}, i.e., $F - k(x_{total} - x_1)$ where k is the spring constant. This is illustrated in Figure 5.2(b). The experienced force is measured with sensors and fed back to the operator through the haptic device. We opt to use a virtual physics environment to represent the remote domain. The key advantages of using a virtual physics environment are the complete access of all information in the remote domain and repeatable experimentation that provides all participants a consistent and reproducible experience. Note that physics interaction calculations in the VE reflect the general behavior in a physical environment.

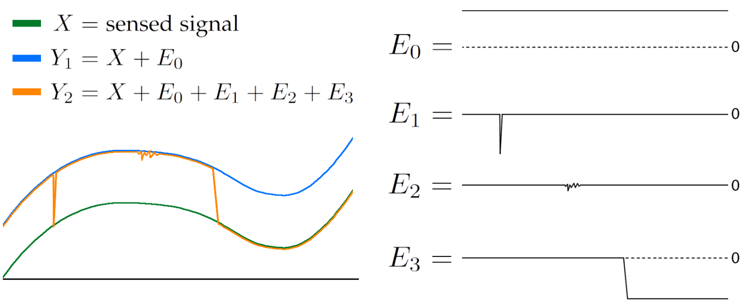


Figure 5.3: Illustration of decomposition of errors on two estimated signals Y_1 and Y_2.

5.2.2. Decomposition of Error

Consider the concept of an ideal network facilitating a TI system. Such a system must not generate any error, i.e., sensed signals in one domain are reproduced with neither temporal error (zero delay) nor spatial (position) error. Such a TI system can be described as,

$$Y[k] = X[k],$$

where $X[k]$ is the sensed sample in one domain at time k and $Y[k]$ is the estimated value at the other domain at time k. However, in practice, signal reproduction is prone to errors. Therefore, a practical TI system can be represented by,

$$Y[k] = X[k + l[k]] + E[k], \tag{5.1}$$

where $E[k]$ represents the position error at time k, and $l[k]$ is the temporal error at time k. Note that these errors are themselves dependent on k. Although in Equation (5.1) we represented the error conceptually as a whole, we can improve the analysis by considering temporal and position errors as multiple components acting simultaneously. In other words, $E[k]$ and $l[k]$ can be decomposed as,

$$E[k] = \sum_m E_m[k] \text{ , and } l[k] = \sum_n l_n[k], \tag{5.2}$$

where $E_m[k]$ is the m^{th} component of $E[k]$ and $l_n[k]$ is the n^{th} component of $l[k]$. It should be noted that in Equation (5.2) we do not define any correlation between errors for the sake of simplicity. However, error components can be correlated with the sensed signal or other error components. Therefore, there is an unlimited number of ways to decompose $E[k]$. We illustrate this concept in Figure 5.3. Here, two estimations Y_1 and Y_2 of sensed signal X are shown. Y_1 appears to be identical to X apart from a stationary offset, while Y_2 has multiple deviations. An example of how the error in Y_2 can be decomposed into multiple components is shown with E_0, E_1, E_2, and E_3. Any decomposition can be considered as long as their sum matches the total.

This notion of error decomposition allows us to isolate errors and examine their impact on the operator separately. The objective of this work is not to extract error components from the signal but to consider some common errors that provide a scope for improving TI performance. We now consider some common errors that occur in TI communication and examine their impact on the user experience.

5.2.3. Perceivability of errors

We now consider a few types of errors that offer us the most interesting opportunities to improve the TI performance. We examine the *perceivability of errors* which is the impact of an error on the user experience. To this end, we consider three specific types of errors.

Stationary offset

A mapping between the operator and remote domains is defined for reproducing the operator's actions. This causes any unique location in the operator domain to point to a unique location in the remote domain. The choice of this mapping is heavily dependent on the application. There can be an application where the operator spans the entire workspace of the teleoperator and another application where the operator is interested in fine-grained movements in a limited portion of the workspace. In any case, the operator learns the deployed mapping by interacting with the TI setup. Let us consider the former scenario in which each point in the teleoperator's workspace is uniquely mapped to a point in the operator's workspace and vice-versa. Let us suppose there is a stationary offset of 2 cms in the mapping, and the operator intends to pick and place an object in the remote domain. If the operator can perform all actions as intended, the offset does not pose any issues. On the other hand, if the teleoperator (HIP in case of VE) is a few centimeters away from the object while the operator has reached the workspace edge and can not move any further, then the offset starts to make a negative impact, and this is undesirable.

The operator realizes the offset only due to the presence of *reference points*. In the above example, the workspace edge acts as the reference point. This is illustrated in Figure 5.4. Here, the blue cube indicates the workspace of the haptic device. An example of a good map from the operator to the teleoperator's workspace is the green cube in the remote domain. However, a stationary offset results in the red cube being the teleoperator's workspace. If the operator intends to touch the remote object (vase), the teleoperator can never really allow that since it can access only a portion of the object. However, the interaction would have been satisfactory if the object resided in the green and red cubes' overlapping regions. The reference point, in this case, is a combination of the workspace and the desired area of operation.

From the above examples, it can be observed that reference points play a significant role in governing the perceivability of stationary offsets. If the offset is small with respect to the reference points, then the offset can be considered imperceivable to the operator. However, the same offset can be a significant error for objective metrics.

Velocity Scaling error

When the operator performs an action, if the teleoperator moves considerably faster or slower than the operator, then it becomes perceivable. For example, if the operator moves the hand, and the teleoperator barely moves, this will be highly perceivable. However, small variations in the scale of the velocity of the teleoperator's movements will be imperceivable.

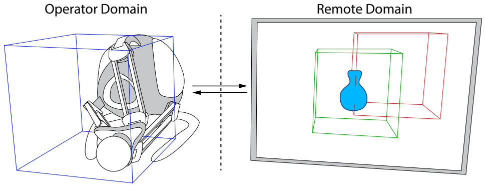


Figure 5.4: Illustration of the notion of reference points. The blue cube indicates the operator's workspace, whereas the green and red cubes indicate the desired workspace and stationary offset workspace of the teleoperator, respectively. In the offset case, the object (blue vase) can never be fully accessed, leading to performance issues.

DELAY-INDUCED POSITION ERRORS

Due to the inherent TI delay (network, processing, among others), there would be a lag in replicating the operator's actions, leading to position errors. The presence of haptic feedback strongly determines how perceivable these position errors are. When the HIP is distant from the objects in the VE, there is no haptic feedback. Hence, position errors corresponding to even a few milliseconds between operator and remote domains will not cause any disturbances in the operator's ability to teleoperate. On the other hand, if the HIP is in the vicinity of or in contact with VE objects, these position errors could create undesirable haptic feedback. For example, there is a sharp transition between free space and hard object since the force rises rapidly from zero (free space) to a considerable value (on the object's surface). Hence, even minor position errors can harm the user experience. Suppose the operator is transitioning from free space to hard object. If the force feedback is delayed, then the operator would have applied a large force to the object before force feedback is experienced. The large penetration generates a high force that could impair the operator's teleoperation ability. Hence, minor delay-induced position errors could be highly detrimental.

5.2.4. BLIND SPOTS IN OBJECTIVE MEASURES

So far, we have explored multiple errors and their perceivability for a human operator. To find opportunities that are not explored in the state-of-the-art in TI, we focus on errors that either have (i) *a large impact on objective measures but a small impact on the user experience* or (ii) *a small impact on objective measures but a large impact on the user experience.*

Stationary offset and velocity scaling error belong to the former category, and delay-induced position error belongs to the latter. Clearly, in these cases, contradictory inferences are drawn by objective measures and user experience. Since user experience is the KPI in TI systems, any measure that does not agree with it manifests severe shortcomings with respect to performance characterization. Hence, we argue that there exist *blind spots* in objective measures, which is their inability to characterize TI performance properly. As an example, we take the stationary offset discussed in Section 5.2.3. Objective measures based on network parameters are agnostic to the underlying data. Therefore, there is no way to identify any error term. However, that does not mean those network parameters are not useful. An increased delay and increased information loss will undoubtedly deteriorate the system's performance. However, it does have blind spots to pinpoint the performance more accurately.

A simple objective measure that is signal-aware is RMSE. A stationary offset will cause a significant increase in the RMSE, which can marginalize other deterioration like high-frequency noise. This is a blind spot within RMSE, causing it to significantly drop in effectiveness when any form of stationary offset is present. Another example is the delay-induced position errors described in Section 5.2.3. Here we identify that force feedback significantly impacts the consequence of an error. Objective measures based on network parameters or RMSE can identify a delay or an error but do not consider their effect on the physical environment. This concept is another blind spot in objective measures.

We now leverage the insights gained on perceivability of errors to improve user experience. To this end, we propose *Adaptive Offset Framework (AOF)* for reconstructing a smooth kinematics signal in the remote domain.

5.3. ADAPTIVE OFFSET FRAMEWORK (AOF)

In Sec. 5.2, a small stationary offset was deemed almost imperceivable, with the caveat that the offset should be sufficiently small with respect to potential reference points. This observation provides us with a range of stationary offsets that can be maintained indefinitely without affecting the user experience. This range can be deployed as an *adaptive offset*. The adaptive offset can be deployed just before the estimation is used. With this, we get

$$Z[k] = Y[k] + A[k], \tag{5.3}$$

where $Z[k]$ is the reconstructed value at time k and $A[k]$ the state of the adaptive offset at time k. $Y[k]$ is the estimation as defined in Equation (5.1). Whenever errors are identified in the system, they can be absorbed into the adaptive offset instead of correcting for them directly. The error can then be handled at a later time. This enables us to address these errors at the opportune moment.

The adaptive offset by itself does not provide an improvement to the user experience. We introduce *shaping functions* that modify the adaptive offset to improve the user experience. Their intended purpose is to mask *errors with a small impact on objective measures but a large impact on the user experience* so that with minimal dependence on the adaptive offset, they improve the user experience. At the same time, we need to prevent the offset from ever-increasing. To this end, we introduce *decay functions* whose primary goal is to shrink and contain the offset, by utilizing *errors with a large impact on objective measures but a small impact on the user experience*. Multiple shaping and decay functions can be active simultaneously. We define the adaptive offset as

$$A[k] = A[k-1] + \sum_{p} S_p[k] + \sum_{q} D_q[k], \tag{5.4}$$

where $S_p[k]$ is the contribution of the p^{th} shaping function at time k, and $D_q[k]$ the contribution of the q^{th} decay function at time k. The interconnection between the different modules of AOF is shown in Fig. 5.5.

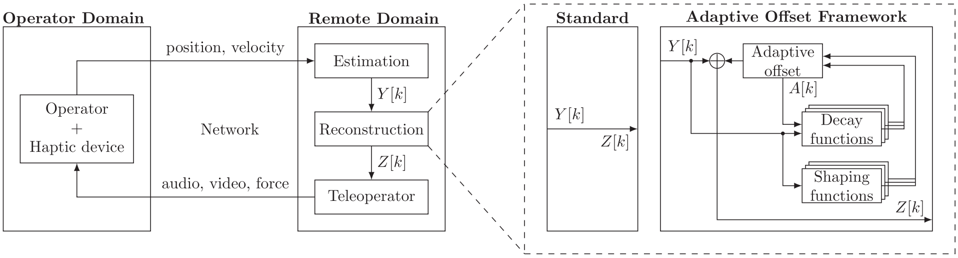


Figure 5.5: Block diagram representation of the proposed AOF solution. Also shown is the contrast with standard reconstruction methods in TI literature.

5.3.1. SHAPING FUNCTION FOR ONE-SHOT CORRECTION ERROR

An offset can be built up when new information is not received at the controlled domain. When new information eventually arrives, the standard method is to adjust to the new information as soon as possible, causing all offset to be removed in one shot. We define this change as a correction. The benefit of the correction is clear. It removes all of the position offset in the system by going to the intended position. However, the problem is that these corrections can result in short spikes in the velocity that were not present in the original signal. We call this spike in velocity a *one-shot correction error*. The one-shot correction error is experienced as an impulse by the user. If this happens in the vicinity of a physical object, a large spike in force feedback can be experienced. These effects are highly perceivable, especially when the spike in force feedback is sufficiently large. When the corrections are sufficiently large, these can be perceived even visually. The effects are even more pronounced in the presence of delay and packet losses.

The one-shot correction error can be removed entirely by subtracting an equal amount of the offset from the buffer when the correction is performed. Instead of a high spike in velocity, the adaptive offset is altered. To do this, the correction needs to be calculated before it can be committed to the buffer. When calculating a new reconstruction after a new packet was received, one should not calculate the correction for the upcoming step but the correction needed in the previous step. That way, the signal maintains its velocity correctly. $S_{\text{corr}}[k] = Y[k-1] - X[k-1]$, where $k-1$ indicates the time of the previous estimation and the arrival of the latest packet. S_{corr} is the shaping function that targets the correction error.

Depending on the quality of the network, removing the corrections can create a large amount of pressure on the adaptive offset. Therefore an option to tune the aggressiveness of the shaping function is useful. One way to use the same concept less aggressively is to introduce a threshold τ_{corr}. Any corrections smaller than a certain amount are deemed acceptable, but anything larger could hurt the user experience. This also potentially works well when packet loss is present, which can sometimes cause potentially harmful corrections. The resulting shaping function is defined as,

$$S_{\text{corr}}[k,\tau] = \begin{cases} 0 & \text{if } \|S_{\text{corr}}[k,0]\| < \tau_{\text{corr}}, \\ S_{\text{corr}}[k] - \tau_{\text{corr}}\hat{S}_{\text{corr}}[k] & \text{otherwise,} \end{cases} \tag{5.5}$$

where τ_{corr} is the threshold and $\hat{S}_{\text{corr}}$ the unit vector of S_{corr}.

5.3.2. SHAPING FUNCTION FOR DELAY

Inevitably there exists a delay between the master and controlled domain. This delay can be caused by the Round Trip Time (RTT), information loss, or other methods. Multiple ways can be considered to decrease this problem. For example, future predictions can be considered. However, prediction does not come for free and runs the risk of instability. For this work, we will consider a different approach to suppress the effects of delay.

In order to decrease the adverse effects of delay on the user experience, a more specific error needs to be found. We can use a concept discussed previously, where position errors are more perceivable when interacting with physics objects. A delay allows the operator to move through solid objects like walls without feeling force feedback for a short period. When the force feedback does arrive, the operator has already moved deep into the object, causing the device in the controlled domain to apply a lot more force to the physical object than the operator intended.

There is a way for the controlled domain to recognize instances when the force feedback in the system and the force feedback experienced by the operator have a mismatch. When the force feedback changes, the controlled domain knows this before the operator in the master domain. The system can keep track of the information experienced by the operator, and with that information available, a shaping function can be crafted.

Let $F_{\text{controlled}}[k]$ be the sensed force in the controlled domain at time k. Then we take $F_{\text{master}}[k]$ as the force feedback experienced by the operator in the master domain at time k. The difference in force can be calculated as

$$F_{\text{difference}}[k] = F_{\text{controlled}}[k] - F_{\text{master}}[k]. \tag{5.6}$$

We now make the assumption that the effect of $F_{\text{difference}}$ would result in an amount of velocity, would the operator have experienced it. We can then proactively apply the effects of that velocity by modifying the buffer. The resulting shaping function becomes

$$S_{\text{delay}} = f_{\text{delay}}(F_{\text{difference}}), \tag{5.7}$$

where S_{delay} is the shaping function, and f_{delay} is a function indicating the amount of velocity as a result of the force difference.

With the shaping function stated above, the kinematic data is slowed down as it moves through a rigid object. Of course, this "slowing down" only happens at the receiver. The operator is not affected. The goal is to suppress a potentially unintended spike in force. However, there is a high risk of a positive feedback loop: the pressure is lessened because of the sensed increase in force. Then the operator feels less force as a result and slows down less. Once again, the sensed force is increased. With this loop, the wall can appear very weak.

To make sure the effect of S_{delay} is as desired, f_{delay} should be chosen appropriately. Additionally there is a consideration in how the $F_{\text{controlled}}$ and F_{master} are obtained. We propose to deploy a shift register at the receiver that keeps track of recent force measurements. Through communication a Round Trip Time (RTT) can be obtained. Based on the RTT, F_{master} can be chosen from the shift register. We then choose $f_{\text{delay}}(F_{\text{difference}}) = C_{\text{delay}} \cdot F_{\text{difference}}$, where ${}_{\text{delay}}C$ is a constant that linearly correlates the difference in force with a velocity. For a fixed RTT, this naturally balances the offset created by S_{delay}. For an infinitely long session, if it ends with a period of zero force, the cumulative effect of S_{delay} will be zero. This property reduces the pressure of S_{delay} on the adaptive offset. Still C_{delay} needs to be chosen carefully so it has a noticeable effect, but not so high that it makes the walls feel flimsy.

5.3.3. Decay functions

Besides shaping functions, we need adequate decay functions to handle the pressure shaping functions apply to the adaptive offset. The combined effort of all deployed decay functions should handle the pressure provided by all deployed shaping functions.

Velocity scaling Decay function

As shown in Sec. 5.2, a stationary scaling error is almost imperceivable. By extension, slightly scaling the velocity at run-time is also hard to perceive. This provides an excellent opportunity to shrink the adaptive offset.

Any time the pointer moves, the component in the direction of the adaptive offset can be slightly scaled, either increasing or decreasing the movement slightly. When there is a non-zero velocity in the system, this function provides a steady shrinking of the adaptive offset. We first need to project the movement and we can project the velocity onto the adaptive offset.

$$\dot{X}_{\text{projection}}[k] = \frac{\dot{X}[k] \cdot A[k]}{\|A[k]\|} \frac{A[k]}{\|A[k]\|} \tag{5.8}$$

where $\dot{X}_{\text{projection}}[k]$ is the projected velocity. Now we scale the projection depending on it matches or opposes the direction of the adaptive offset.

$$D_{\text{scaling}}[k] = \begin{cases} C \cdot \dot{X}_{\text{projection}}[k] & \text{if } \frac{\dot{X}[k] \cdot A[k]}{\|A[k]\|} \geq 0, \\ \frac{-C}{1+C} \cdot \dot{X}_{\text{projection}}[k] & \text{otherwise.} \end{cases} \tag{5.9}$$

Here D_{scaling} is the decay function based on scaling velocity, and C_{scaling} a scalar indicating the strength. To choose a good value for C_{scaling} we can use the same concept deployed in Perceptual Deadband. A concept called Just Noticeable Difference (JND) indicates how much a velocity can differ before the operator notices it. Typically this value is stated as 10%. Here the same value can be used for C_{scaling}.

An alternative method is to make C_{scaling} dependent on the size of the adaptive offset. The idea is that there is not a strong need for the decay function to act for a small adaptive offset, but when the size is relatively big, the offset should be suppressed more strongly.

$$C_{\text{scaling}}[k] = \frac{2\|A[k]\|}{B_{\max}}, \tag{5.10}$$

where $B_{\max}$ is the maximum size of the buffer. A $C_{\text{scaling}} = 2$ means that all velocity in the direction of the offset will be completely nullified. It is possible to choose different functions for C_{scaling}, but exploring other options remains future work.

With the above described shaping and decay functions, we arrive at a specific implementation of the Adaptive Offset Framework, which can be expressed as

$$A[k] = A[k-1] + S_{\text{corr}}[k,\tau] + S_{\text{delay}}[k] + D_{\text{scaling}}[k]. \tag{5.11}$$

There is sufficient scope to explore other functions to further improve the user experience.

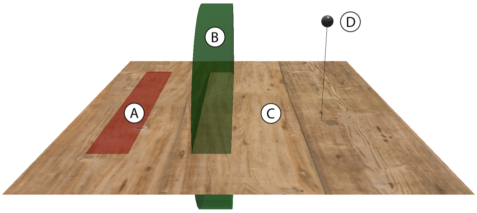


Figure 5.6: A snapshot of the *FollowMe* game used for the performance evaluation of AFO. The target 'A' needs to be tracked by moving the green object indicated as 'B' using the haptic device. 'C' is a rigid surface like a table and 'D' is the HIP.

5.4. EXPERIMENTAL SETUP

As explained in Section 5.2.1, in this work, we use a VE setup for our experimentation. Note that AOF's working and performance evaluation also applies to remote physical environments. Virtual physics exist within conventional game engines. Conventional game engines calculate physical parameters linked to the frame rate, typically 60 Hz. This rate is insufficient as haptic signal requires 1 kHz update rate. The TI testbed proposed by Bhardwaj et al. [28] solves this problem. The game engine used is Chai3D, which has a physics engine detached from the frame rate, and runs physics calculations at the required 1 kHz.

As shown in Figure 5.2(a), in our the master domain houses the haptic device and a monitor. The controlled domain houses the VE, and the two domains are connected via a real network. In the master domain, the force is fed to the haptic device. The kinematic data of the dynamic objects are used to update virtual copies of those dynamic objects stored locally. The rendering engine then produces frames locally.

We develop a VE game that we call *FollowMe* where the task is for the user to track a continuously moving target using the haptic device. A snapshot of the FollowMe game is shown in Figure 5.6. The demo was designed with minimal VE objects to ensure a consistent experience across different participants. The only objects are a rigid, immovable surface ('C' in the figure) and a slider ('B' in the figure) that can be moved using the haptic device.

In order to test the efficacy of AOF, we perform experiments under a wide variety of different network behaviors. We use Netem – a standard network emulation tool to consistently and precisely emulate various network conditions. A consistent behavior is desirable as it helps reduce variance in performance and isolate the effects of AOF.

Table 5.1: Description for subjective grading.

10	no perceivable impairment
8-9	slight impairment but no disturbance
6-7	perceivable impairment, slight disturbance
4-5	significant impairment, disturbing
1-3	extremely disturbing

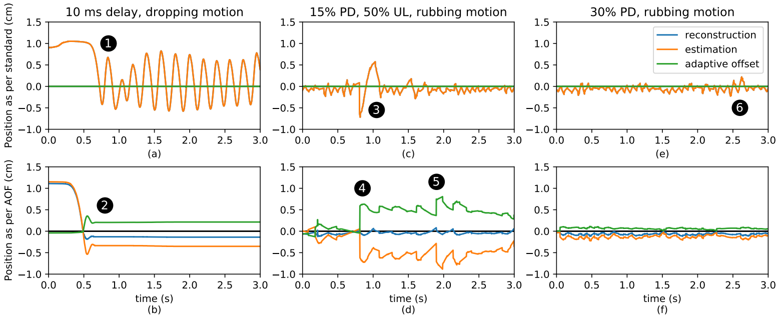


Figure 5.7: Demonstration of the working of AOF. The rigid surface is placed at a height of 0 cm. The force feedback is inversely proportional to the penetration depth below 0 cm, and zero otherwise. In (a) and (b) a dropping motion is performed and in (c), (d), (e), and (f) a rubbing motion is performed.

We consider network delay, uniform packet loss, and bursty packet loss settings. Bursty packet loss is induced using the Gilbert-Elliot model.

We also deploy Perceptual Deadband (PD) [13] – a state-of-the-art compression scheme for haptics signals. PD works by estimating the perceptually insignificant samples. The transmitter can avoid sending such samples leading to improvement in application bandwidth requirement. For example, a PD of 15 % implies that a sample is transmitted only if the percentage change in magnitude with respect to the previous transmitted sample is higher than 15 %.

Experimental Procedure

The goal of the experiment is to investigate the effect of different forms of performance degradation on the user experience. In this regard, we consider three specific tasks: ① *Pushing* the slider at a steady rate. This motion helps in recognizing subtle disturbances due to PD and packet loss. ② *Dropping* the HIP on the surface from a height as there is a sharp transition in generated force. This motion is like resting a hand on a table. ③ *Rubbing* the rigid surface. There is both a steep transition in force and smooth motion. Participants are requested to experiment with all three actions to get a more inclusive idea of the user experience in a more realistic scenario. Participants are given time to familiarize themselves with the application, typically five minutes.

Participants are presented with ten sets of network settings in random order. Once a setting is chosen, it is given twice – once with AOF and the other with standard behavior in a random order, as explained in Figure 5.5. Hence, there are 20 different scenarios altogether. For every setting, the target travels a predefined trajectory for 20 seconds. At the end of each setting, the participant grades the experience as per Table 5.1. The subjective study involved fifteen participants in the age group between 20 and 64 years, with an average of 32 years. No participant suffered from known neurological disorders.

5.5. Performance Evaluation

In this section, we examine the behavior, objective performance, and subjective evaluations of AOF. First, we illustrate AOF's functionality through examples highlighting

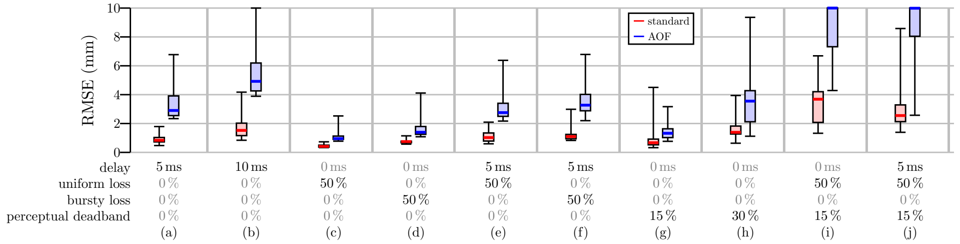


Figure 5.8: RMSE is based on logged data for each experiment. The RMSE values are capped at 10 mm, but the last two columns values go higher than the limit. One can see that the proposed AOF consistently scores worse than the usual method of immediate corrections as per the objective measure (RMSE).

its ability to address common teleoperation challenges. Next, we analyze AOF's performance using objective metrics. Finally, we evaluate user-perceived performance through subjective grading.

5.5.1. AOF BEHAVIOR ANALYSIS

To illustrate the working of AOF, we first present some examples, shown in Figure 5.7 . Each data set is created within the experimental setup. At 0 cm on the vertical axis, there is a rigid surface extending downwards. We plot the teleoperator position in the vertical axis while TI interaction is being conducted. We measure the generated force based on reconstructed position as explained in Section 5.2.1. Figure 5.7(a) and Figure 5.7(b) correspond to a *dropping motion* where the operator attempts to put drop the device down on the rigid surface while being subjected to 10 ms of RTT. Figure 5.7(c), Figure 5.7(d), Figure 5.7(e), and Figure 5.7(f) correspond to a *rubbing motion* where the operator attempts to rub the device over the rigid surface. The difference between AOF and standard behavior is described in Figure 5.5. We will explain some of the key performance benefits of AOF from these examples.

1. Suppression of oscillations. In Figure 5.7(a), one can see that the dropping motion causes significant oscillations with the standard reconstruction method. This effect is indicated by marker ❶. Due to the delay, the user enters the surface without feeling the force feedback, thus penetrating deeper before the force feedback arrives. The force feedback is larger than desired as the penetration depth is larger. This causes the user to be pushed out of the surface quicker. When the operator continuously applies downward force, this causes oscillations. This is typical of TI because of the delay in the network, which otherwise would not be physically possible. Figure 5.7(b) corresponds to AOF and the function $S_{\text{delay}}[k]$ is active. $F_{\text{difference}}$ is linearly proportional to a velocity added to the adaptive offset. The velocity slows down when the user moves into the surface, and the force feedback has not yet been experienced. Consequently, the surface is penetrated less deep than before, and a lower force is generated. Consequently, the operator is forced less aggressively out of the surface. When the operator applies constant downward pressure, the forces in opposite directions are identical. Therefore, the operator can lay on the surface comfortably, as seen at marker ❷.

2. Suppression of large one-shot corrections. In Figure 5.7(c) and Figure 5.7(d), there is

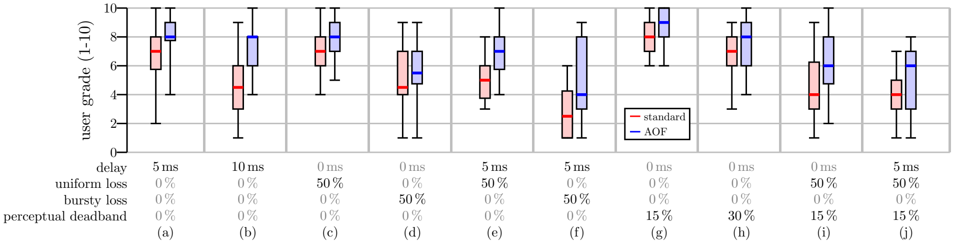


Figure 5.9: Subjective user grades showing that the proposed AOF provides a significantly higher user experience.

15% PD and 50% uniform packet loss. The standard behavior, to always apply corrections in one shot whenever new information is received, is demonstrated by the marker ❸. The operator is immediately forced out of the surface due to a one-shot correction. Since the delay is negligible, oscillations are absent. The amount of information loss is very high. Therefore the size of the corrections can be so large that when canceled out by $S_{\text{corr}}[k]$, the change in adaptive offset does not go unnoticed. An example is shown by the marker ❹. An upside of canceling the correction is that no significant undesired force is generated onto the surface. This means that both the surface and the operator do not experience a sudden spike in measured force. If there would be a TI application with a delicate object, this is undoubtedly an improvement over the standard method. A second example of a correction being suppressed is shown at marker ❺.

3. Suppression of small one-shot corrections. In Figure 5.7(e) and Figure 5.7(f), 30% PD is used. This scenario results in smaller and more consistent corrections than in the previous scenario. In Figure 5.7(e), one can see that the estimation appears consistent. However, there is a consistent high-frequency component. This directly results in a noticeable high-frequency component in the measured force and a distinctly recognizable deterioration for the operator. The high-frequency force can push the operator out of the surface completely, as is seen at the marker ❻. In Figure 5.7(f), a combination of $S_{\text{delay}}[k]$ and $S_{\text{corr}}[k]$, suppress the high frequency signal and produce a more smooth experience. The reconstruction is consistently just below the surface with a smaller variance than seen in Figure 5.7(e). Here, AOF helps the user move over the surface more smoothly without clear downsides. Only a marginal amount of adaptive offset is used to accomplish this feat.

5.5.2. OBJECTIVE ANALYSIS

There are multiple objective measures to consider, and among these are multiple traditional network performance parameters like packet loss or transmission delay. However, in comparing AOF and the standard behavior, identical network behavior is used. Therefore network parameters will not provide an insight into the difference between AOF and the standard behavior.

Alternatively, some methods look at the underlying data. Multiple objective measures have been proposed over the years, but none of the proposed methods address the blind spots we highlight in this work. As a representative of these methods, we use RMSE. The data is plotted in Figure 5.8. One can see that the reconstructions produced by AOF pose a significant increase in RMSE for every network scenario. Note that for the two rightmost columns, the displayed RMSE is capped at 10 mm, but some of the measured RMSE is well over that value. We use this data to make several observations.

1. AOF creates an overall deterioration in RMSE. Based on RMSE, AOF is outperformed by standard behavior for every scenario tested. This is expected, as AOF actively maintains an adaptive offset, which RMSE will take strong note of. We elaborate further in Section 5.2.4.

2. Delay overshadows effects from information loss. Figure 5.8(a), Figure 5.8(e), and Figure 5.8(f) have the same 5 ms delay, but with no loss, uniform loss and bursty loss, respectively. One can see that the observed RMSE is consistent between these methods. However, it is reasonable to expect that adding uniform and burst loss would deteriorate the system performance.

3. Significant information loss dominates RMSE for AOF. Figure 5.8(i) and Figure 5.8(j) both have a combination of PD and uniform loss. This combination significantly impacts RMSE, especially when AOF is included. Because PD removes most redundancy in the communication, all network loss drops affect packets of significant importance. Because of this, the number of significant corrections is numerous. With the presence of $S_{\text{corr}}[k]$ this causes a significant impact on the adaptive offset and thus the RMSE.

5.5.3. SUBJECTIVE ANALYSIS

1. AOF creates an across-the-board improvement. As per Figure 5.9, AOF improves the user grade significantly for every network scenario compared to the standard method yielding an average of up to three points (on a scale of ten). Note that we expect more improvements by further tuning the shaping and decay functions. A wider variety of tasks and a significantly more extensive data set should give a more accurate view of AOF's improvements and limitations.

2. Significant improvement for delay. In Figure 5.9(a) and Figure 5.9(b), the network is only affected by delay. Because there is no information loss $S_{\text{corr}}[k]$ is inactive, leaving $S_{\text{delay}}[k]$ as the only active shaping function. One can see that the effect of the user grade is significant, where a 10 ms delay with AOF scores better than 5 ms delay with the standard method. This is a significant result, as TI demands extremely low delay. This suggests that AOF can potentially relax the stringent delay requirement. However, more research is needed to verify this conclusively.

3. Comparing shaping functions. We consider different scenarios in Figure 5.9(c), Figure 5.9(d), Figure 5.9(g), Figure 5.9(h), and Figure 5.9(i) where no delay is added to the network. In these scenarios, $S_{\text{delay}}[k]$ is marginally active, leaving $S_{\text{corr}}[k]$ as the main shaping function. While for each scenario the inclusion of AOF poses an improvement, the benefits are smaller than any of the scenarios where delay is present. This suggests that, while $S_{\text{corr}}[k]$ is beneficial, $S_{\text{delay}}[k]$ is even more so. This can be partly explained, because the effect of $S_{\text{corr}}[k]$ has the same irregular and one-shot nature, as the correction errors it targets. This also puts pressure on the adaptive offset and the decay functions at play.

4. Uniform versus Bursty loss. Figure 5.9(c), Figure 5.9(d) have uniform loss and bursty loss respectively. The average packet loss is identical, which means that the number of packets dropped is identical between the methods. The only difference is the distribution.

A bursty loss distribution causes an average difference in user grade of three points. This significant difference illustrates the disruptive effect of bursty loss on TI applications. The observation is in line with the expectations. A bursty loss model risks longer periods without transmissions, causing the performance to take a large dip at irregular intervals. Uniform loss is fundamentally more consistent and provides a more convincing user experience.

5. Uncovering blind spots of objective results. When considering RMSE as a measure for performance, it would seem that AOF is a deterioration of standard behavior. However, when considering the user grades shown in Figure 5.9, AOF provides across-the-board improvement. There are several blind spots at play for this result to happen. First, the concept that a stationary offset is almost imperceivable is not being considered. Instead, the adaptive offset has a massive impact on RMSE. Secondly, the concept that velocity scaling is almost imperceivable is similarly not considered. Causing all efforts by $D_{\text{scaling}}[k]$ to increase RMSE. In both these cases, the blind spots are the lack of consideration for the imperceivability of these errors. Secondly, RMSE does not notice that the high-frequency corrections are mostly nullified. The intentional compensation in velocity, which leads to less unstable force feedback, is not considered either. RMSE does not consider force feedback or any additional information related to the environment. In both these cases, the blind spots are the lack of understanding that specific differences improve the user experience significantly. This is further explained in Section 5.2.4. With this, we demonstrate the blind spots present in RMSE and currently available objective measures. Additionally, we demonstrate how we successfully exploit this underutilized potential from the perceivability of errors with AOF to improve the user experience.

5.6. Conclusions

Haptic Bilateral Teleoperation presents fresh challenges due to a human-in-the-loop with haptic feedback in teleoperation. Generally, stringent requirements in terms of latency and reliability are often stated. However, by curating the experience tailor-made for a human operator and exploiting the limited human perception, we can significantly relax the stringent requirements for TI while maintaining a satisfying performance. In this work, we examined how errors can be classified based on their perceivability and impact on the user experience. We proposed the Adaptive Offset Framework (AOF) to exploit perceivable and imperceivable errors by modifying the adaptive offset to improve the user experience. Subjective experiments confirmed that AOF improves user experience in every network configuration. Specifically, we show that AOF significantly enhances the user grade, up to 3 points (on a scale of 10) compared to the standard reconstruction method. We compared these results with objective analysis and demonstrated multiple blind spots in objective measures that led to an incorrect characterization of the performance of the TI application. We believe that the concepts explored in this work can provide numerous additional opportunities to improve the user experience, further relaxing the TI system requirements.

In this chapter, we examined alterations aimed at enhancing the operator's perception. However, the methods presented here do not leverage knowledge of the application's physical behavior to improve the experience further. This aspect will be explored in the next chapter.

6

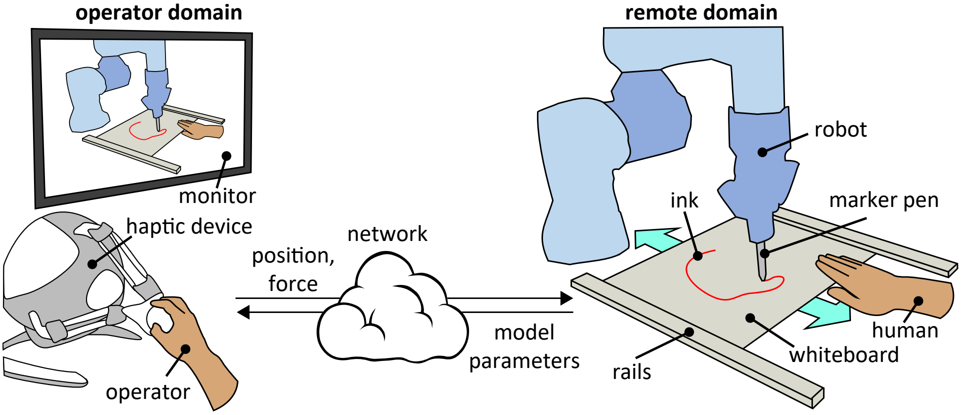
Model Mediated Teleoperation with Operator Intent[1]

6.1. Introduction

In this chapter, we address Sub-Question 5 as stated in Section 1.4: *How can we relax the delay requirement with alternative feedback mechanisms?* In a Haptic Bilateral Teleoperation (HBT) system, the operator must receive two essential types of feedback: visual and haptic. This chapter focuses specifically on active force feedback as the haptic feedback mechanism. As discussed in Chapter 2 and Chapter 3, this type of feedback requires a latency of less than 10 ms, with an ideal target of less than 1 ms. However, this does not necessarily mean the network itself must achieve such stringent low-latency requirements. An alternative approach is to predict force feedback rather than relying solely on real-time measurements and communication. While accurately predicting the magnitude of force feedback can be challenging, ensuring correct timing is significantly easier and provides a practical way to meet the demanding 1 ms latency requirement. This concept is utilized in Model Mediated Teleoperation.

Unlike acti, Model Mediated Teleoperation (MMT) starts with the assumption of considerable network delays – instead of trying to decrease the latency directly – and tries to mitigate its impact on the transparency and stability of the system [71, 72, 73]. An MMT system consists of two primary components: (a) the operator and (b) the remote environment. At the operator's end, a comprehensive model is constructed to replicate the characteristics of the remote environment. The operator uses a haptic device to transmit actions to the remote robot, receiving instantaneous feedback based on the local

[1] This chapter is based on the manuscript titled *"Utilizing Operator Intent for Haptic Teleoperation Under High Latencies"*, which has been accepted for publication in *IEEE Transactions on Mobile Computing* and is expected to appear in 2025.



model of the remote environment. A high-level overview of this framework is presented in Figure 6.1.

On the remote side, the robot follows the received commands and simultaneously gathers sensor data, such as force, position, and audio-visuals. This information facilitates real-time estimation of the model parameters of the remote environment. Instead of transferring all the sensory data from the remote end to the operator domain, only the model parameters are sent. The digital twin in the operator domain is then updated using these parameters. While MMT effectively addresses significant network delays, spanning several seconds [71, 74], 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 chapter, 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.

Contributions

The contributions in this chapter are listed below.

① We introduce key design principles for MMT solutions that prioritize operator intent.

② To enhance the scalability of MMT solutions, we advocate leveraging modern physics engines rather than relying on handcrafted models.

③ We present a comprehensive framework tailored for MMT solutions operating in complex and dynamic environments, incorporating an imitation controller.

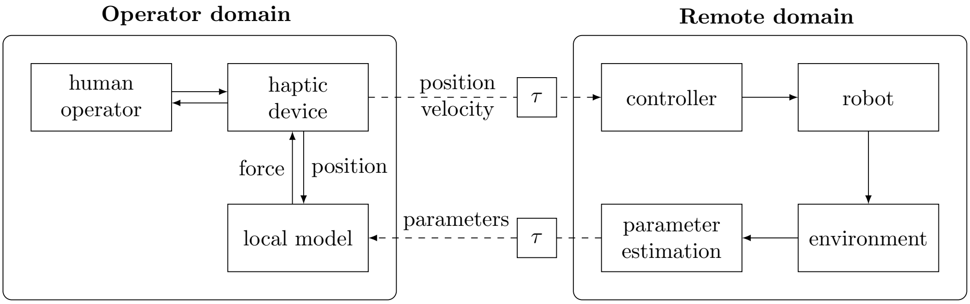


Figure 6.1: General structure of a Model-Mediated Teleoperation system.

(4) The proposed framework and design principles are demonstrated through a practical application where a robot arm is guided to draw on a whiteboard that is in motion.

(5) We implement the drawing application on a system where the operator and remote environment are separated by a distance of 8000 km.

(6) A user study highlights the efficacy of our approach, showing significant improvements in user experience under network latencies of up to 1 s, with a 3-point increase on a 7-point Likert scale.

6.2. RELATED WORK

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 [75]. Several methods have been explored to reduce the model jump effect, including delaying model updates and alerting operators about impending updates [76, 77, 74]. 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 [74]. Finally, MMT struggles with dynamic environments with moving objects. Xu et al. initiated the advancement of MMT to accommodate movable objects [78]. They adopted a model-based approach tailored to a particular scenario, limiting its broader applicability.

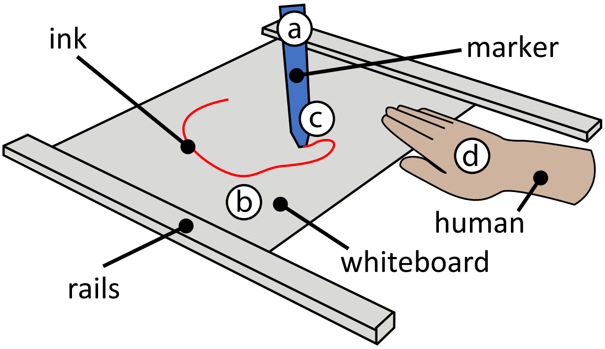


Figure 6.2: Illustration of an application where a marker draws on a whiteboard while a human moves the whiteboard between two rails.

Discerning human intent

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 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 [79, 80, 81]. 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 [82, 80, 83]. 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 [84].

6.3. Design Principles for human centered MMT

In this section, we present design principles to make a system for user centered bilateral teleoperation. Instead of attempting to design a system that does not have any mismatch between the operator and remote side, we propose an approach where the mismatch is embraced, and solutions are placed to address the consequences of these mismatches by prioritizing operator intent and perception. In Section 6.5, we validate these design principles with a concrete application and user study.

6.3.1. 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 interacting with an imperfect simulation of that environment. Discerning this intent is straightforward when the environments at the operator and the remote domain align. Direct use of operator trajectory and force suffices. However, when substantial latency, imperfect simulations, or inaccurate measurements cause mismatches, interpreting the operator's intent becomes complex, given its subjective nature and the need to evaluate multiple indicators.

We identify four aspects to consider operator intent. Operator's behavior, object's behavior, hard transitions, and reactions, are outlined below.

Operator's behavior The primary indicator of operator intent is their trajectory. From this trajectory, further insights can be obtained. For example, quick and purposeful movements might signal urgency, whereas measured, careful actions suggest precision and caution. Beyond just the global trajectory, it is also valuable to assess the operator's position relative to nearby objects.

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.

For example, in Figure 6.2, the marker (a) is directly attached to the end-effector. One can consider the movement of the marker, the movement relative to the whiteboard, and the amount of force applied to the whiteboard as the operator's behavior.

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.

For example, in Figure 6.2, the whiteboard (b) experiences force applied by the operator. While the trajectory of the whiteboard is altered, it is not initiated by the operator and therefore not their intent.

Hard transitions In many situations, even a slight change in position or force can make a significant difference. For example, in Figure 6.2 at (c), the marker touches the whiteboard, a small increase in height would stop the ink from being deposited. It is often assumed that the person handling these situations is aware of these hard transitions. Therefore, if someone were to imitate the action and not follow through, it would be immediately noticeable. These hard transitions can also apply to the state of an object. For example, when using a soldering iron, it is highly relevant whether the soldering tin has melted or not. Since these hard transitions are highly context-specific, general-purpose solutions for bilateral teleoperation have difficulty addressing them.

Reactions The final consideration for operator intent is whether the action of an operator is a response to an event in the remote environment. For example in Figure 6.2 (d), an individual in the remote domain is manipulating the whiteboard. If the operator wants the drawing to stay consistent despite this movement, they will have to react to it, and manipulate the marker to compensate for the motion.

6.3.2. 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 [6, 35, 18]. 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.

6.3.3. Operator's experience of the remote environment

A fundamental aspect of haptic bilateral teleoperation is ensuring that the operator can experience interactions occurring within the remote environment. There are types of interactions that have already been experienced in the operator domain in the form of instantaneous force feedback. Those forces therefore do not have to be relayed a second time.

This does pose a challenge when there are also *remotely initiated interactions*. For example, in Figure 6.2, if the individual in the remote domain would push against the marker, this would be an interaction involving force feedback that has not yet been experienced by the operator. In order to be able to experience the remote environment fully, remotely initiated interactions need to be separated from already experienced interactions and relayed to the operator.

6.3.4. 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

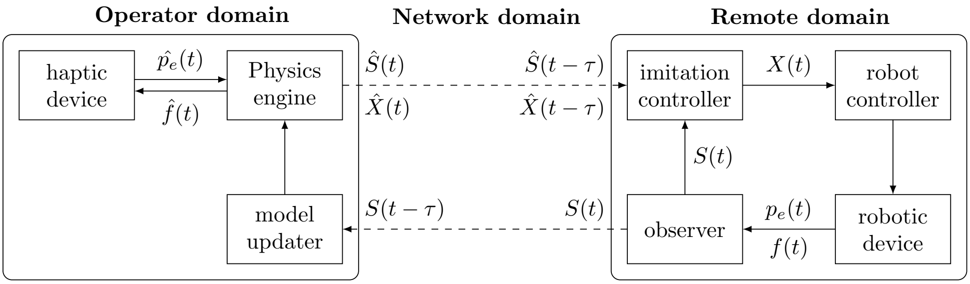


Figure 6.3: Illustration of our proposed framework that aims to extend MMT to work with complex and dynamic environments by considering operator intent.

detect minor inaccuracies, provided the simulated interaction is plausible.

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

6.4. A FRAMEWORK FOR MMT WITH OPERATOR INTENT

In this section, we outline a framework for designing a system for bilateral 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 Figure 6.3. 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 provided as priors or are observed in the remote domain in realtime.

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

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

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 [76, 77].

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 [6]. 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.

6.5. 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.

Because the network link has an average latency of 179 ms, an approach with local predictive force feedback is required. Without predictive force feedback, the operator can unintentionally crush the marker against the whiteboard without applying any force. We follow the design considerations stated previously to design a sound control strategy based on operator intent.

Firstly, we identify the hard transitions in the application. 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 mm can cause a significant difference.

Secondly, we identify the importance of the operator's behavior. A key observation is that when drawing, the relative position from the marker to the whiteboard is important. A mismatch in the whiteboard position between the operator and remote domain can lead to the robot drawing on a different part of the whiteboard than the operator intended. When making a drawing this can lead to a crooked image. Similarly, the marker should only be pressed with force against the whiteboard if the operator also used force to press the marker against the whiteboard in the operator domain. In this application, the only dynamic object is the whiteboard, which can only be manipulated in the remote domain. Therefore, in this specific case, the object's behavior is irrelevant to discern operator intent.

Finally, there are the operator's reactions to events in the remote environment. In this case, the key event caused in the remote environment is the constant motion of the whiteboard. In order to make the drawing, the operator has in mind, we need to compensate for the motion of the whiteboard. The trajectory of the operator is directly influenced by the movement in the remote domain, and if due to network delay, this compensation is misaligned with the actual whiteboard movement the drawing will not match.

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

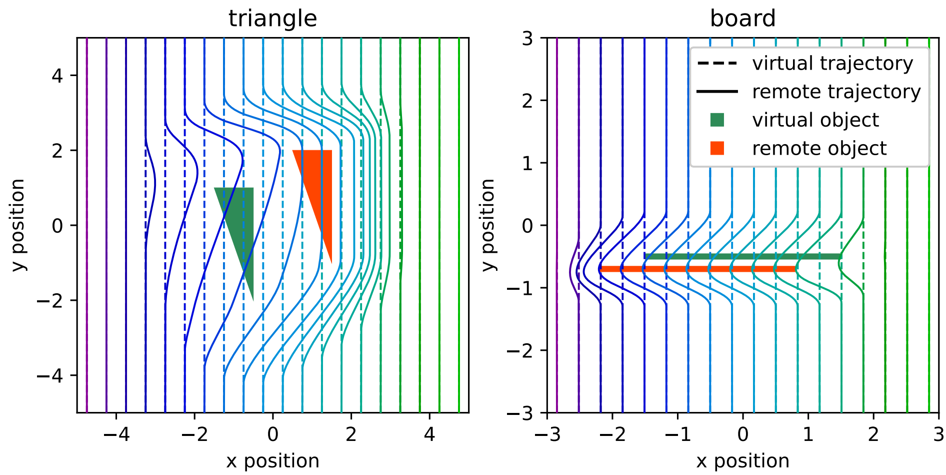


Figure 6.4: Illustration of the behavior of the imitation controller design. For a given set of virtual trajectories of $\hat{\boldsymbol{p}}_p$ in the operator domain, the modified trajectory of $\boldsymbol{p}_p$ in the remote domain is shown.

in the form of a mismatch between the actual and the intended drawing. Whenever the marker is not directly touching objects, however, small differences in position and velocity are hard to distinguish. This is particularly true when there are no clear points of reference signaling to the operator what the correct position would be. 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.

Based on these insights, 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.

6.5.1. Translation between absolute and relative trajectory

In this section, we describe how to design an imitation controller that respects operator intent in our specific application. Given the control signal from the operator domain $\hat{X}$, a modified version of the control signal, X_{mod}, is expressed such that it improves the realization of operator intent in specific situations. Next, a transition factor α is chosen to smoothly transition between $\hat{X}$ and X_{mod}. We get

$$X(t) = \alpha \hat{X}(t-\tau) + (1-\alpha) X_{\text{mod}}(t). \tag{6.1}$$

Note that $\hat{X}(t-\tau)$ includes the communication delay τ to update $\hat{X}$ in the remote domain.

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

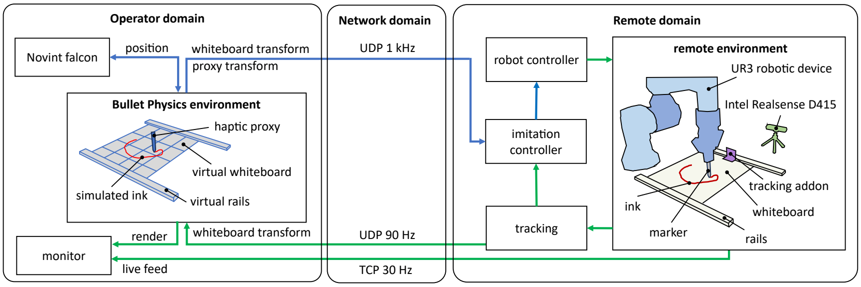


Figure 6.5: Schematic overview of experimental setup. In the figure, we lay down the different components of our setup, showcasing how they relate to the proposed teleoperation framework and how data flows through the system. Blue arrows indicate high-frequency communication of 1 kHz, while green arrows indicate a medium-frequency communication of approximately 60 Hz.

force $\hat{\boldsymbol{f}}$. The control signal in the operator domain is thus $\hat{X} = \{\hat{\boldsymbol{p}}_p, \hat{\boldsymbol{f}}\}$. 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,

$$X_{\text{mod}}(t) = \{\hat{\boldsymbol{p}}_p(t-\tau) + \boldsymbol{p}_o(t) - \hat{\boldsymbol{p}}_o(t-\tau), \hat{\boldsymbol{f}}(t-\tau)\}, \tag{6.2}$$

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.

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

$$g(x) = \begin{cases} 0 & \text{if } x \leq 0, \\ 1 & \text{if } x \geq 1, \\ 3x^2 - 2x^3 & \text{otherwise.} \end{cases} \tag{6.3}$$

Next, we 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 denote $|\cdot|$ as the l^2 norm of a vector. 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

$$\hat{\boldsymbol{p}}_{\text{min}} = \underset{\boldsymbol{p} \in P}{\arg\min}(\boldsymbol{p} + \hat{\boldsymbol{p}}_o - \hat{\boldsymbol{p}}_p). \tag{6.4}$$

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 $\frac{d}{dt}\hat{\boldsymbol{p}}_p(t-\tau)\cdot\frac{d}{dt}\boldsymbol{p}_p(t) > 0$.

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. (6.3) 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.

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

$$\alpha = \begin{cases} g\left(\frac{|\hat{\boldsymbol{p}}_{\text{min}} - \frac{1}{2}\text{proj}_{(\boldsymbol{p}_o - \hat{\boldsymbol{p}}_o)}\hat{\boldsymbol{p}}_{\text{min}}|}{|\boldsymbol{p}_o - \hat{\boldsymbol{p}}_o|}\right) & \text{if } \hat{\boldsymbol{p}}_{\text{min}} \cdot (\boldsymbol{p}_o - \hat{\boldsymbol{p}}_o) > 0, \\ g\left(\frac{|\hat{\boldsymbol{p}}_{\text{min}}|}{|\boldsymbol{p}_o - \hat{\boldsymbol{p}}_o|}\right) & \text{otherwise,} \end{cases} \tag{6.5}$$

where $\text{proj}_{\boldsymbol{b}}\boldsymbol{a}$ is the projection of $\boldsymbol{a}$ onto $\boldsymbol{b}$. Finally, we can use Eq. (6.2), (6.3), and (6.5) in (6.1) 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 Figure 6.4.

6.6. 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 Figure 6.5.

In the operator domain, the Bullet-Physics engine provides the local simulation [85]. 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 kHz. The physics engine is decoupled from the rendering engine so that the physics engine can run at 1 kHz while the OpenGL-based renderer runs at 60 Hz. This ensures a smooth and responsive haptic response from the physics engine with sub 1 ms computational delay. A photo of the operator domain is shown in Figure 6.6(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 mm until fully compressed. A ROS2 environment communicates directly with the UR3 [86]. 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 Hz. A photo of the remote domain is shown in Figure 6.6(b).

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

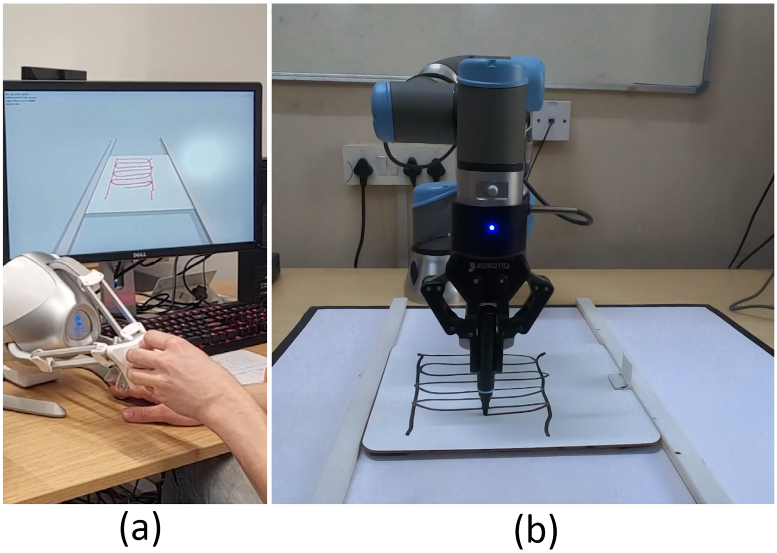


Figure 6.6: The experimental setup used in the user study. (a) the operator domain and (b) remote domain.

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.

6.6.1. CONTROLLER DESIGN

Conventionally, a robotic device in such applications is manipulated with a compliance controller that makes use of an accurate force-torque sensor [87]. 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 [87]. The applied force $\boldsymbol{f}$ is converted into a positional offset. We can calculate the target position fed to the position controller as

$$\boldsymbol{p}_{\text{target}} = \boldsymbol{p}_p + \frac{1}{k_s}\boldsymbol{f}, \tag{6.6}$$

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.

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 mm above the whiteboard. Consequently, drawing in the remote domain will only occur when the operator applies force. Only then is the displacement caused by $\frac{1}{k_s}\boldsymbol{f}$ enough for the marker to apply pressure to the whiteboard. To match this behavior in the virtual domain, if enough force is applied to the whiteboard, ink is deposited accordingly.

In the user study, two control strategies are investigated. In the first control strategy, only the operator's behavior is considered. The operator's trajectory and applied force directly lead to a target position using Eq. (6.6). In the second control strategy, the mismatch in the whiteboard position between the operator and remote domains is considered. Here, Eq. (6.1) and Eq. (6.6) are used to create a target position that tracks the whiteboard when it is in its vicinity.

6.6.2. 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. 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 whiteboard's position. Consequently, a high-precision, 90 Hz sampling rate tracking solution with minimal computational lag was realized. There are a multitude of alternatives that can be used for tracking an object that is restricted to a specific 1 DoF motion, but a low latency measurement is highly beneficial when using the measurement to have the robot compensate for the whiteboard's movement.

In this setup, the 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.

6.6.3. NETWORK

The operator and remote domains are separated by an approximate distance of 8000 km. Kinematic and force data from the operator, as well as kinematic data from the whiteboard, are relayed over a UDP channel at 1 kHz and 90 Hz, respectively. Both feedforward and feedback packets carry a 100-byte payload. Additionally, the packets contain a sequence number so that only the most recent packets are used, while out-of-order packets are ignored. A video stream from the remote domain is forwarded to a secondary computer in the operator domain at a maximum rate of 30 Hz. The Round-Trip Time (RTT) for the UDP link was assessed over 10 hours, and the results are depicted in Figure 6.7. 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.

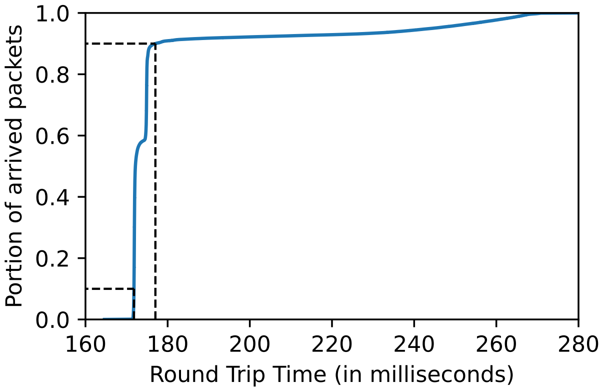


Figure 6.7: Cumulative distribution of end-to-end network latency measured over 15 hours. 80 % of the packets have latencies between 172 ms and 177 ms. The average latency is 179 ms.

6.6.4. 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 Figure 6.6. 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 second is added. Participants are tasked with rating each scenario in the following four aspects. Each aspect is rated on a Likert scale with seven points.

- *Picture matching*: How well does the final image in the remote domain match what you looked at while drawing?
- *Controllability*: How much control do you have over the drawing in the remote domain?
- *Immersion*: Do you feel like you are present at the remote location and all the things happening there are experienced by you?

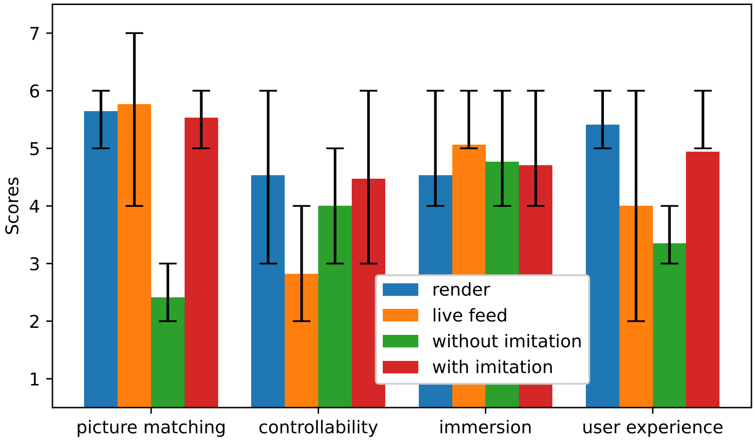


Figure 6.8: The results of the user study. Four scenarios are rated on four categories using a 7-point Likert scale.

- *Overall experience*: This rating reflects the user's overall experience, taking into account factors such as picture matching, controllability, and immersion. These aspects are prioritized based on the user's personal preferences.

The user study was performed with 20 participants.

6.7. PERFORMANCE ANALYSIS

Figure 6.8 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.

6.8. Conclusion

In this chapter, we 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 built this application on a system where the operator and remote domain are 8000 km apart with an average end-to-end network latency of 165 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 s.

By combining the advancements presented in this chapter with those from the previous chapter, a broader vision for realizing HBT systems in the near future takes shape. This vision will be outlined in the next and concluding chapter.

7

Conclusions and Vistas

Haptic Bilateral Teleoperation (HBT) could enable people to manipulate remote environments as if they were physically present. People would be able to perform intricate repairs in hazardous locations and facilitate the sharing of specialized skills, such as calligraphy, all over the world.

Achieving this vision requires more than simply constructing a HBT system. After all, what would such a system even look like? Key questions arise: How feasible is the realization of these systems? What performance indicators are critical and what are their requirements? A practical implementation requires not only technical advancements in areas like robotics and networking, but also a deep understanding of the human operator's role as an integral part of the system.

In this work, we set out to chart a path toward the practical application of HBT over long distances. The aim was to provide a human operator with a satisfactory experience while performing a task remotely. The research goal was thus stated as follows.

How to realize haptic bilateral teleoperation across long distances?

To address this research question, we explored ways to improve both the understanding and performance of HBT systems. The contributions span from network design and control methods, to system architecture and human experience. Below is a summary of the key contributions in this thesis.

7.1. Summary

Characterizing kinematic data transmissions - Chapter 2. Traditional Quality of Experience Metrics struggle with the effects of latency. This chapter introduced a novel metric called Effective Time- and Value-Offset (ETVO) for assessing the quality of kinematic data streams over networks. The method does not consider typical network performance indicators, but directly compares the measured time sequence to the reproduced one after transmission. This metric distinguishes between noise and latency-induced variations, offering a more precise analysis of network impacts on the system.

In this work, we found that there is a considerable difference in priority between latency and loss of information. Where less than 10 ms delay already yields a considerable decrease in user satisfaction, we demonstrated that a large packet loss is barely noticeable for kinematic data sent at a packet rate of 1 kHz. Moreover, we demonstrated that a jitter buffer is detrimental to performance, and that packets arriving out of order should be treated as lost data for best performance.

Characterizing force feedback transmissions - Chapter 3. In this chapter, we examined the feedback loop between operator actions, robotic response, and the resulting force feedback. It was shown that it is not the time difference caused by network latency that the operator perceives, but rather the amplification in force feedback, which can significantly alter the user's experience. We introduced the Tactile Internet (TI) Metric (TIM), a method to determine the network performance required for specific teleoperation tasks. Furthermore, we proposed the channel compensation spring, a mechanism that adjusts the system to mitigate the negative effects of latency, reducing its impact on the application. We demonstrated that the channel compensation spring is effective across all levels of network latency, successfully compensating for approximately 4 ms of delay while maintaining a satisfactory user experience.

MAC for teleoperation - Chapter 4. In this chapter, we designed the ViTals MAC protocol, optimizing the transmission of teleoperation data streams, by accounting for the different requirements of kinematic data and video traffic. This is achieved by leveraging the functionality of the separate voice and video access categories in WiFi 6. By tuning this mechanism we can optimize the network that is facilitating haptic and video traffic, while being realizable with existing architecture.

We demonstrated that ViTals utilizes the different requirements of each type of data traffic, and that this leads to an improvement of the overall user experience. The ETVO algorithm and a user study have both been used to confirm the efficacy of ViTaLS.

Improving User Experience with Deliberate Alterations - Chapter 5. This chapter focused on enhancing user perception by masking network-induced alterations with nearly unnoticeable deliberate changes. The Adaptive Offset Framework was proposed to improve user experience by exploiting gaps in human perception. In particular, we utilized the fact that high frequency noise and abrupt changes are profoundly more noticeable than low frequency differences. In this work, we demonstrated this method to yield a considerable improvement in user experience, when subject to network latency.

Model Mediated Teleoperation with Operator Intent - Chapter 6. In this chapter, we worked with Model Mediated Teleoperation (MMT), a strategy to bypass latency issues through predictive interactions using local physics simulations. While MMT reduces latency requirements, it presents challenges that grow with the system's complexity. This chapter broadens the capability of MMT to handle interactions with objects in motion. A method is proposed to reproduce a precise interaction with an object by considering not only the motions of the operator directly, but also the motions relative to objects in close proximity, and have the robot compensate for it. The method is implemented on an application that enables an operator to make a drawing on a surface that is in motion, which would not be possible without such adjustments.

We demonstrated a working HBT interaction on a setup with the operator located in the Netherlands and the remote domain in India, where the operator was able to make a drawing remotely on an object in motion. A user study proved the efficacy of this method.

7.2. KEY INSIGHTS

Across the different contributions presented in this work, there are several key insights that will help develop future HBT systems.

Low latency at the cost of loss of information and throughput. The three most critical Network Performance Indicators are latency, throughput, and reliability. For an application like HBT, throughput is typically a lower priority since the amount of transmitted information is not necessarily large. This is well-documented in the literature, and the TI requirements do not emphasize throughput [1, 88]. Reliability and latency are considered as more significant. The effects of reliability on the kinematic modality were examined in Chapter 2, Chapter 3, and Chapter 4, where we consistently found that reliability has minimal impact. Our simulations showed functional systems even with up to 40% packet loss. This outcome is expected, given the strong temporal correlation of the kinematic modality. At a transmission rate of 1 kHz, the loss of a packet has little effect on the signal quality. The key takeaway is that networks supporting such data streams should prioritize low latency above all else, including reliability.

The human factor. Defining precise objective requirements for an HBT system is inherently challenging due to the complexities of human perception. The primary goal is to provide the operator with a seamless, satisfying experience while allowing him to effectively manipulate the remote environment. This introduces both difficulties and opportunities. On the one hand, objectively measuring system performance becomes complex, as human experience is subjective, highly variable, and not easily quantified. Chapter 2 and Chapter 3 delve into this challenge in detail, providing multiple contributions to better characterize the performance of HBT systems.

On the other hand, the human brain is highly adaptable, actively working to create a coherent, positive experience, and willing to compensate for sensory gaps or imperfections. This adaptability allows HBT systems to function effectively, even when there are significant discrepancies between the remote environment and the operator's perception. Chapter 5 and Chapter 6 explore ways to leverage this adaptability, focusing on providing the operator with the impression of experiencing the remote environment rather than striving for complete accuracy. Both chapters demonstrate how this approach can enhance user experience in the presence of network latency and, ultimately, reduce the requirements of HBT systems.

The key takeaway is that there is a significant opportunity to relax system requirements by leveraging human perception. This could bring the latency requirement up from 1 ms to over 100 ms. Therefore, a redefinition of the TI is necessary to align its stated requirements with its envisioned purpose.

Predict feedback with low latency requirement. In an HBT system, the operator must receive at least two types of feedback: visual and haptic. For the purposes of this work, we focused solely on active force feedback as haptic feedback. As demonstrated in Chapter 2 and Chapter 3, this type of feedback demands a latency of less than 10 ms, and ideally less than 1 ms. However, this does not necessarily imply that the network itself must always meet such low-latency requirements. Instead, the key takeaway is that direct measurement of force feedback should be avoided. As explored in Chapter 6, a more promising solution is to predict force feedback rather than measure it in real time. More broadly, this leads to the following conclusion: feedback with a latency requirement under 10 ms should be predicted rather than directly measured and transmitted over the network.

7.3. The Path Forward

With the presented work and insights, what is our recommended path for the future of HBT systems, and how do the contributions outlined here fit into that vision? Traditionally, one might expect a broad outlook, offering ideas to expand the research scope and explore new applications. However, in this case, we present a more specific recommendation, taking off from the previous section.

Future HBT systems should combine predictive force feedback with live video transmission. Force is well suited for prediction since the data involved in predicted force is relatively minimal compared to video, and human operators are unlikely to notice small inaccuracies in force feedback. By focusing on predictive force, we open up possibilities for overcoming time constraints that currently limit several promising applications of HBT systems. Leveraging predictive force feedback can directly address the challenge of achieving the low latency necessary for smooth, effective operations.

In contrast, video feedback offers a different kind of challenge. Predicting visual feedback is highly complex, with difficulty growing exponentially as environments become more intricate. For example, predicting how liquids will behave and appear, particularly when they are manipulated, presents a significant challenge. There is also the possibility of displaying events to the operator that may never actually happen. However, live video transmission is not bound by the same strict latency requirements as force feedback, with a requirement of approximately 100 ms. This flexibility allows us to sidestep the considerable challenges that come with predicting video, offering a more reliable alternative for visual feedback.

Given this specific approach, how can the research presented here contribute to the future of HBT systems? Figure 7.1 illustrates how the findings from earlier chapters can inform and enhance future HBT systems that leverage live video transmission with predictive force feedback.

The network could be optimized to support the transmission of kinematic data, live video, and model parameters that drive the simulation. The ViTaLS MAC protocol, as outlined in Chapter 4, is well-suited for this purpose, efficiently managing both large video packets and small, frequent kinematic data while optimizing for low latency. By integrating this with other network enhancements, we can further optimize the system

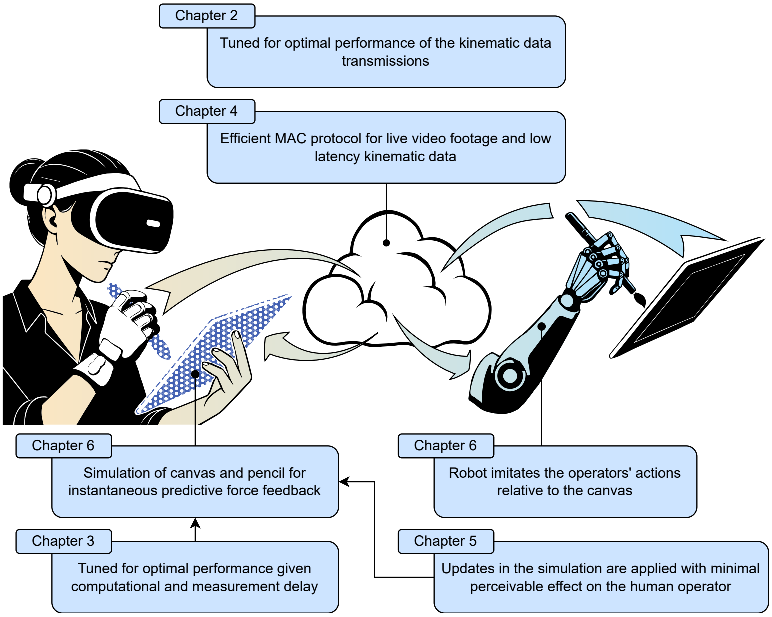

Chapter 2
Tuned for optimal performance of the kinematic data transmissions
Chapter 4
Efficient MAC protocol for live video footage and low latency kinematic data
Chapter 6
Simulation of canvas and pencil for instantaneous predictive force feedback
Chapter 6
Robot imitates the operators' actions relative to the canvas
Chapter 3
Tuned for optimal performance given computational and measurement delay
Chapter 5
Updates in the simulation are applied with minimal perceivable effect on the human operator

Figure 7.1: A HBT system that combines live video and kinematic data transmissions with predictive force feedback. The system uses contributions from every chapter in this thesis.

and validate its performance using the ETVO metric from Chapter 2, ensuring an efficient and reliable setup for future simulations.

Predictive force feedback is generated by a local physics simulation, as described in Chapter 6. The TIM metric from Chapter 3 demonstrates a direct relationship between system delay and the dynamics of stiff spring interactions, where increased delay can make rigid interactions more difficult. Delays can arise from various sources, including local computation and I/O. By using the TIM metric, we can fine-tune the maximum stiffness of the simulation that produces the predictive force feedback.

The operator domain simulation should be continuously updated with measurements from the remote environment. Since the operator receives only force feedback, their perception of the simulation state is limited. The Adaptive Offset Framework from Chapter 5 offers a way to distinguish between noticeable and unnoticeable changes, enabling updates to the local simulation without impacting the operator's experience. This approach allows for smoother adjustments and enhances the overall performance of the system.

Lastly, the robot should precisely replicate the operator's actions. Since the operator operates within a simulation, there is additional context beyond just the kinematic motions. As shown in Chapter 6, this extra information can be leveraged to fine-tune the robot's behavior, enhancing its movements in relation to the objects the operator interacts with. For instance, when the operator in the virtual domain grasps an object by its handle, the robot can adjust to pick up the corresponding object by the handle as well, even if the object has shifted or rotated slightly. This enables the robot's actions to align more closely with the operator's intended actions.

7.4. CHALLENGES AHEAD

We have discussed a future outlook for HBT systems. Specifically, one that involves the use of live video and predictive force feedback. For such a system, what are the challenges ahead?

Ongoing efforts to reduce latency in networking, along with protection against interference, will be greatly beneficial, and enable HBT systems to exist together with other forms of traffic that inhabit the internet.

Investments in real-time physics engines, particularly from the video game industry, are pushing the boundaries of realism and complexity. These advances can be applied to teleoperation, enabling support for more complex interactions.

The growing automation industry is driving improvements in sensor technologies and real-time data processing, providing more accurate and detailed representations of remote environments, which will directly benefit HBT.

The most pressing challenge ahead is reliance on prediction, which introduces significant challenges. The robot must react to its environment before knowing how the operator will respond to it. Meanwhile, the operator experiences predicted force feedback for an event that has not yet occurred and may unfold differently. For instance, the operator might feel the force of grabbing a bottle that, in reality, has already fallen over. These challenges become more complex as latency increases. Addressing these issues is essential, as the effectiveness of teleoperation systems depends on their ability to manage the complications of these predictions.

7.5. RETROSPECTIVE GLANCE

We started our research effort from a network vantage point, and steadily included more aspects of the system into our efforts, including the human operator interacting with the system. Throughout this research, our understanding has evolved significantly, driven by the emergence of new ideas, challenges, and solutions.

At the outset, the stringent latency demands of the TI, particularly the 1 ms latency requirement, appeared unattainable. The fundamental constraints imposed by the laws of physics made achieving such performance over long distances next to impossible. However, as our research progressed, new concepts reshaped our perspective. Exploring human perception revealed that many physical barriers could be mitigated, suggesting the feasibility of a functional system that offers the operator a satisfying experience. The introduction of Model Mediated Teleoperation (MMT) further fueled this optimism, though practical constraints such as rendering complex visual environments suggest that such systems may initially be feasible only in highly confined scenarios. For instance, simulating liquids remains notoriously challenging. Yet, even here, new opportunities emerge: the goal is not perfect accuracy but providing plausible force feedback to create the perception of interacting with a liquid to a human operator, a task that may be achievable through simple estimations.

Significant challenges and yet-undiscovered solutions lie ahead, but none appear insurmountable. At the conclusion of this research, we are optimistic about the future of HBT and its eventual realization. When the time comes, this technology could bring about positive change and fundamentally reshape how we interact with the world.

Bibliography

[1] Gerhard P Fettweis. "The Tactile Internet: Applications and challenges". In: *IEEE Vehicular Technology Magazine* 9.1 (2014), pp. 64–70.

[2] Chong Li et al. "5G-Based Systems Design For Tactile Internet". In: *Proceedings of the IEEE* (2018), pp. 1–18. ISSN: 0018-9219. DOI: 10.1109/JPROC.2018.2864984. URL: https://ieeexplore.ieee.org/document/8452975/.

[3] Joachim Sachs et al. "Adaptive 5G low-latency communication for tactile Internet services". In: *Proceedings of the IEEE* 107.2 (2018), pp. 325–349.

[4] Kwang Soon Kim et al. "Ultrareliable and low-latency communication techniques for tactile internet services". In: *Proceedings of the IEEE* 107.2 (2018), pp. 376–393.

[5] Joseph P Verburg et al. "Setting the Yardstick: A Quantitative Metric for Effectively Measuring Tactile Internet". In: *IEEE INFOCOM*. 2020.

[6] HJC Kroep et al. "Etvo: Effectively measuring tactile internet with experimental validation". In: *IEEE Transactions on Mobile Computing* (2023).

[7] Martin Wollschlaeger, Thilo Sauter, and Juergen Jasperneite. "The future of industrial communication: Automation networks in the era of the internet of things and industry 4.0". In: *IEEE industrial electronics magazine* 11.1 (2017), pp. 17–27.

[8] Martin Maier et al. "The tactile internet: vision, recent progress, and open challenges". In: *IEEE Communications Magazine* 54.5 (2016), pp. 138–145.

[9] Mohamad Eid, Jongeun Cha, and Abdulmotaleb El Saddik. "Admux: An adaptive multiplexer for haptic–audio–visual data communication". In: *IEEE Transactions on Instrumentation and Measurement* 60.1 (2010), pp. 21–31.

[10] Burak Cizmeci et al. "A Multiplexing Scheme for Multimodal Teleoperation". In: *ACM Trans. Multimedia Comput. Commun. Appl.* 13.2 (Apr. 2017), 21:1–21:28. ISSN: 1551-6857.

[11] P. Hinterseer, E. Steinbach, and S. Chaudhuri. "Perception-Based Compression of Haptic Data Streams Using Kalman Filters". In: *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*. Vol. 5. May 2006, pp. V–V. DOI: 10.1109/ICASSP.2006.1661315.

[12] Vineet Gokhale, Jayakrishnan Nair, and Subhasis Chaudhuri. "Congestion Control for Network-Aware Telehaptic Communication". In: *ACM Trans. Multimedia Comput. Commun. Appl.* 13.2 (Mar. 2017), 17:1–17:26. ISSN: 1551-6857.

[13] Peter Hinterseer et al. "A novel, psychophysically motivated transmission approach for haptic data streams in telepresence and teleaction systems". In: *Proceedings.(ICASSP'05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005*. Vol. 2. IEEE. 2005, pp. ii–1097.

[14] Peter Hinterseer et al. “Perception-based data reduction and transmission of haptic data in telepresence and teleaction systems”. In: *IEEE Transactions on Signal Processing* 56.2 (2008), pp. 588–597.

[15] Cagatay Basdogan et al. “An experimental study on the role of touch in shared virtual environments”. In: *ACM Transactions on Computer-Human Interaction (TOCHI)* 7.4 (2000), pp. 443–460.

[16] Zhenhui Yuan et al. “User quality of experience of mulsemedia applications”. In: *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 11.1s (2014), p. 15.

[17] N Sakr, ND Georganas, and J Zhao. “A perceptual quality metric for haptic signals”. In: *2007 IEEE International Workshop on Haptic, Audio and Visual Environments and Games*. IEEE. 2007, pp. 27–32.

[18] Rahul Chaudhari, Eckehard Steinbach, and Sandra Hirche. “Towards an objective quality evaluation framework for haptic data reduction”. In: *2011 IEEE World Haptics Conference*. IEEE. 2011, pp. 539–544.

[19] Rania Hassen and Eckehard Steinbach. “HSSIM: An objective haptic quality assessment measure for force-feedback signals”. In: *2018 Tenth International Conference on Quality of Multimedia Experience (QoMEX)*. IEEE. 2018, pp. 1–6.

[20] Stan Salvador and Philip Chan. “Toward accurate dynamic time warping in linear time and space”. In: *Intelligent Data Analysis* 11.5 (2007), pp. 561–580.

[21] Lawrence R Rabiner and Bernard Gold. “Theory and application of digital signal processing”. In: *Englewood Cliffs, NJ, Prentice-Hall, Inc., 1975. 777 p.* (1975).

[22] Hiroaki Sakoe and Seibi Chiba. “Dynamic programming algorithm optimization for spoken word recognition”. In: *IEEE transactions on acoustics, speech, and signal processing* 26.1 (2003), pp. 43–49.

[23] Lei Chen, M Tamer Özsu, and Vincent Oria. “Robust and fast similarity search for moving object trajectories”. In: *ACM SIGMOD international conference on Management of data*. 2005.

[24] Lei Chen and Raymond Ng. “On the marriage of lp-norms and edit distance”. In: *Proceedings of the International conference on Very large data bases*. VLDB Endowment. 2004.

[25] Michail Vlachos, Dimitrios Gunopoulos, and George Kollios. “Discovering similar multidimensional trajectories”. In: *icde*. IEEE. 2002, p. 0673.

[26] Donald J Berndt and James Clifford. “Using dynamic time warping to find patterns in time series.” In: *KDD workshop*. 1994.

[27] Diego F Silva and Gustavo EAPA Batista. “Speeding up all-pairwise dynamic time warping matrix calculation”. In: *Proceedings of the 2016 SIAM International Conference on Data Mining*. SIAM. 2016, pp. 837–845.

[28] Amit Bhardwaj et al. “A Candidate Hardware and Software Reference Setup for Kinesthetic Codec Standardization”. In: *2017 IEEE International Symposium on Haptic, Audio and Visual Environments and Games (HAVE)*. 2017, pp. 53–58. DOI: 10.1109/HAVE.2017.8240353.

[29] Eckehard Steinbach et al. “Haptic codecs for the tactile internet”. In: *Proceedings of the IEEE* 107.2 (2018), pp. 447–470.

[30] Tahir Nawaz Minhas et al. “Mobile video sensitivity to packet loss and packet delay variation in terms of QoE”. In: *International Packet Video Workshop*. IEEE. 2012.

[31] James Nightingale et al. “The impact of network impairment on quality of experience (QoE) in H. 265/HEVC video streaming”. In: *IEEE Transactions on Consumer Electronics* 60.2 (2014), pp. 242–250.

[32] Seokhee Lee, Sungtae Moon, and JongWon Kim. “A network-adaptive transport scheme for haptic-based collaborative virtual environments”. In: *ACM SIGCOMM workshop on Network and system support for games*. 2006.

[33] Jae-young Lee and Shahram Payandeh. “Forward error correction for reliable teleoperation systems based on haptic data digitization”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2013.

[34] Markus Rank et al. “Predictive communication quality control in haptic teleoperation with time delay and packet loss”. In: *IEEE Transactions on Human-Machine Systems* 46.4 (2016), pp. 581–592.

[35] Kees Kroep et al. “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.

[36] Lixian Zhang, Huijun Gao, and Okyay Kaynak. “Network-induced constraints in networked control systems—A survey”. In: *IEEE transactions on industrial informatics* 9.1 (2012), pp. 403–416.

[37] Magdi S Mahmoud and Mutaz M Hamdan. “Fundamental issues in networked control systems”. In: *IEEE/CAA Journal of Automatica Sinica* 5.5 (2018), pp. 902–922.

[38] Yodyium Tipsuwan and Mo-Yuen Chow. “Control methodologies in networked control systems”. In: *Control engineering practice* 11.10 (2003), pp. 1099–1111.

[39] Asif Šabanović et al. “Motion control systems with network delay”. In: *Automatika* 51.2 (2010), pp. 119–126.

[40] Payam Naghshtabrizi and Joao P Hespanha. “Stability of networked control systems with variable sampling and delay”. In: *Allerton Conf. on Communication, Control, and Computing*. Citeseer. 2006.

[41] Sandra Hirche, Tilemachos Matiakis, and Martin Buss. “A distributed controller approach for delay-independent stability of networked control systems”. In: *Automatica* 45.8 (2009), pp. 1828–1836.

[42] Zhichun Yang and Daoyi Xu. “Stability analysis and design of impulsive control systems with time delay”. In: *IEEE Transactions on Automatic Control* 52.8 (2007), pp. 1448–1454.

[43] Mohsen Barforooshan et al. “The effect of time delay on the average data rate and performance in networked control systems”. In: *IEEE Transactions on Automatic Control* (2020).

[44] Mahmoud Gamal et al. “Delay compensation using Smith predictor for wireless network control system”. In: *Alexandria Engineering Journal* 55.2 (2016), pp. 1421–1428.

[45] Kenneth Salisbury, Francois Conti, and Federico Barbagli. “Haptics rendering: Introductory concepts”. In: *IEEE computer graphics and applications* 24.ARTICLE (2004), pp. 24–32.

[46] Ben M Chen, Zongli Lin, and Yacov Shamash. *Linear systems theory: a structural decomposition approach*. Springer Science & Business Media, 2004.

[47] Vineet Gokhale et al. “ViTaLS—A Novel Link-Layer Scheduling Framework for Tactile Internet Over Wi-Fi”. In: *IEEE Internet of Things Journal* 10.11 (2023), pp. 9917–9927. DOI: 10.1109/JIOT.2023.3235433.

[48] Oliver Holland et al. “The IEEE 1918.1 “Tactile Internet” Standards Working Group and its Standards”. In: *Proceedings of the IEEE* 107.2 (2019), pp. 256–279.

[49] Jing Qin et al. “Effect of packet loss on collaborative haptic interactions in networked virtual environments: An experimental study”. In: *Presence* 22.1 (2013), pp. 36–53.

[50] Changhua Pei et al. “WiFi can be the weakest link of round trip network latency in the wild”. In: *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*. IEEE. 2016, pp. 1–9.

[51] Toni Adame, Marc Carrascosa-Zamacois, and Boris Bellalta. “Time-sensitive networking in IEEE 802.11 be: On the way to low-latency WiFi 7”. In: *Sensors* 21.15 (2021), p. 4954.

[52] Gaurang Naik et al. “Can Wi-Fi 7 Support Real-Time Applications? On the Impact of Multi Link Aggregation on Latency”. In: *IEEE International Conference on Communications (ICC)*. 2021.

[53] Stefan Mangold et al. “IEEE 802.11 e Wireless LAN for Quality of Service”. In: *Proc. European Wireless*. Vol. 2. 2002, pp. 32–39.

[54] Che-Yu Chang et al. “QoS/QoE support for H. 264/AVC video stream in IEEE 802.11 ac WLANs”. In: *IEEE Systems Journal* 11.4 (2015), pp. 2546–2555.

[55] Katarzyna Kosek-Szott, Marek Natkaniec, and Lukasz Prasnal. “IEEE 802.11 aa intra-AC prioritization-A new method of increasing the granularity of traffic prioritization in WLANs”. In: *2014 IEEE Symposium on Computers and Communications (ISCC)*. IEEE. 2014, pp. 1–6.

[56] Guosong Tian, Seyit Camtepe, and Yu-Chu Tian. “A deadline-constrained 802.11 MAC protocol with QoS differentiation for soft real-time control”. In: *IEEE Transactions on Industrial Informatics* 12.2 (2016), pp. 544–554.

[57] Leonardo Lanante, Chittabrata Ghosh, and Sumit Roy. “Hybrid OFDMA random access with resource unit sensing for next-gen 802.11 ax WLANs”. In: *IEEE Transactions on Mobile Computing* 20.12 (2020), pp. 3338–3350.

[58] Hussein Al Osman et al. “Alphan: Application layer protocol for haptic networking”. In: *2007 IEEE International Workshop on Haptic, Audio and Visual Environments and Games*. IEEE. 2007, pp. 96–101.

[59] Ye Feng et al. “A feasibility study of IEEE 802.11 HCCA for low-latency applications”. In: *IEEE Transactions on Communications* 67.7 (2019), pp. 4928–4938.

[60] Dave Cavalcanti et al. “Extending accurate time distribution and timeliness capabilities over the air to enable future wireless industrial automation systems”. In: *Proceedings of the IEEE* 107.6 (2019), pp. 1132–1152.

[61] Amin Ebrahimzadeh and Martin Maier. “Delay-constrained teleoperation task scheduling and assignment for human+ machine hybrid activities over FiWi enhanced networks”. In: *IEEE Transactions on Network and Service Management* 16.4 (2019), pp. 1840–1854.

[62] Vineet Gokhale et al. “Toward Enabling High-Five Over WiFi: A Tactile Internet Paradigm”. In: *IEEE Communications Magazine* 59.12 (2021), pp. 90–96.

[63] Davide Magrin et al. “Validation of the ns-3 802.11 ax OFDMA Implementation”. In: *Proceedings of the Workshop on ns-3*. 2021, pp. 1–8.

[64] Giuseppe Bianchi. “Performance analysis of the IEEE 802.11 distributed coordination function”. In: *IEEE Journal on selected areas in communications* 18.3 (2000), pp. 535–547.

[65] Ilenia Tinnirello and Giuseppe Bianchi. “Rethinking the IEEE 802.11 e EDCA performance modeling methodology”. In: *IEEE/ACM transactions on networking* 18.2 (2009), pp. 540–553.

[66] Sungkwan Youm and Eui-Jik Kim. “Latency and jitter analysis for ieee 802.11 e wireless lans”. In: *Journal of Applied Mathematics* 2013 (2013).

[67] Boris Bellalta and Katarzyna Kosek-Szott. “AP-initiated multi-user transmissions in IEEE 802.11 ax WLANs”. In: *Ad Hoc Networks* 85 (2019), pp. 145–159.

[68] Oran Sharon and Yaron Alpert. “The combination of QoS, aggregation and RTS/CTS in very high throughput IEEE 802.11 ac networks”. In: *Physical Communication* 15 (2015), pp. 25–45.

[69] H. J. C. Kroep, V. Gokhale, and R. Venkatesha Prasad. “Blind Spots of Objective Measures: Exploiting Imperceivable Errors for Immersive Tactile Internet”. In: *2022 ACM/IEEE 13th International Conference on Cyber-Physical Systems (ICCPS)*. 2022, pp. 01–10. DOI: 10.1109/ICCPS54341.2022.00011.

[70] Mandayam A Srinivasan and Cagatay Basdogan. “Haptics in virtual environments: Taxonomy, research status, and challenges”. In: *Computers & Graphics* 21.4 (1997), pp. 393–404.

[71] Probal Mitra and Günter Niemeyer. “Model-mediated telemanipulation”. In: *The International Journal of Robotics Research* 27.2 (2008), pp. 253–262.

[72] Xiao Xu et al. “Model-mediated teleoperation: Toward stable and transparent teleoperation systems”. In: *IEEE Access* 4 (2016), pp. 425–449.

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

[74] Jingzhou Song et al. “Model-mediated teleoperation with improved stability”. In: *International Journal of Advanced Robotic Systems* 15.2 (2018), p. 1729881418761136.

[75] Bert Willaert, Hendrik Van Brussel, and Günter 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.

[76] Probal Mitra, Diana Gentry, and Gunter 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.

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

[78] Xiao Xu, Sili Chen, and Eckehard 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.

[79] Fanny Ficuciello, Luigi Villani, and Bruno Siciliano. “Variable impedance control of redundant manipulators for intuitive human–robot physical interaction”. In: *IEEE Transactions on Robotics* 31.4 (2015), pp. 850–863.

[80] Leonel Rozo et al. “Learning physical collaborative robot behaviors from human demonstrations”. In: *IEEE Transactions on Robotics* 32.3 (2016), pp. 513–527.

[81] Yanan Li and Shuzhi Sam Ge. “Human–robot collaboration based on motion intention estimation”. In: *IEEE/ASME Transactions on Mechatronics* 19.3 (2013), pp. 1007–1014.

[82] An T Le et al. “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.

[83] Youssef Michel et al. “Bilateral teleoperation with adaptive impedance control for contact tasks”. In: *IEEE Robotics and Automation Letters* 6.3 (2021), pp. 5429–5436.

[84] Ahmed Hussein et al. “Imitation learning: A survey of learning methods”. In: *ACM Computing Surveys (CSUR)* 50.2 (2017), pp. 1–35.

[85] Erwin Coumans and Yunfei Bai. *PyBullet, a Python module for physics simulation for games, robotics and machine learning*. http://pybullet.org. 2016–2021.

[86] Steven Macenski et al. “Robot Operating System 2: Design, architecture, and uses in the wild”. In: *Science Robotics* 7.66 (2022), eabm6074. DOI: 10.1126/scirobotics.abm6074. URL: https://www.science.org/doi/abs/10.1126/scirobotics.abm6074.

[87] Stefan Scherzinger, Arne Roennau, and Rüdiger 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.

[88] Nattakorn Promwongsa et al. “A comprehensive survey of the tactile internet: State-of-the-art and research directions”. In: *IEEE Communications Surveys & Tutorials* 23.1 (2020), pp. 472–523.

ACKNOWLEDGEMENTS

Before I began my PhD, I found myself a little lost. Up until that point, the decisions shaping my academic path had been guided almost entirely by curiosity. I had a tendency to pursue whatever seemed most like magic to me. This was why, when faced with a choice between mechanical and electrical engineering, I chose the latter. I could picture how a car or a large piece of machinery could work, but "what is up with a mobile phone?". My approach worked. The world felt full of possibilities, each waiting for the right idea to reveal itself. I learned to look at any piece of technology and imagine, at least in broad strokes, how it might work. Following that same instinct, I pursued a master's thesis in quantum computing, convinced it was the next great unknown on my journey. Yet this time, instead of the liberating sense of discovery I had come to expect, I felt scientifically confined. The demanding cryogenic conditions required for quantum computing imposed painstaking, costly, and drawn-out design cycles. My usual style of throwing mud at the wall to see what sticks had no room to breathe in that environment. I was in dire need of a big change.

It was at this point that I was approached by Dr. R. Venkatesha Prasad, colloquially known as VP. He contacted me and directly offered me a PhD position. “I like your ideas. I want to explore them and see what happens!” he said, even giving me several weeks to think it over. Rationally, it felt like a strange decision to set aside all the knowledge I had built in other domains and start anew in the field of Networking. Yet intuitively, it seemed exactly the change I had been searching for, so I accepted the offer. The old saying is to trust your gut, and in this case it could not have been more right. I had always been good at coming up with unexpected ideas, but research demands far more than that. I was hot tempered, impatient, and inexperienced. VP created a safe environment where I could bring my chaotic, youthful energy, learn to work effectively with others, and refine my skills at the highest level. These years under the guidance of VP have been an invaluable experience, and I believe I have matured enough to also be able to thrive outside such safe spaces. More than that, I now find myself in a position to be the one providing that kind of environment for others.

As I grew more comfortable working with others, I quickly realized how much more fulfilling I found collaboration compared to working alone. I had already sensed this from earlier experiences, but during my PhD it became increasingly clear that fostering a healthy and productive collaborative environment was something I deeply cared about. In fact, it gradually grew into one of the most meaningful aspects of my work. Most of these collaborations turned into genuinely happy and productive partnerships, though there were, of course, challenging moments too. Still, I can say with confidence that I do not regret working with any of the students I supervised. Each of them taught me valuable lessons and contributed in their own way to my growth as a researcher and mentor. A few of them deserve a special mention. Joseph Verburg was the first master's student I supervised, and at the time, he was arguably a more experienced engineer than I was. He might tell you the ideas were mostly mine, but I firmly believe he played a pivotal role in kickstarting my PhD. His contributions helped me make the leap into the networking domain with the academic quality needed to publish in respected venues. Phu Nguyen, another student I had the privilege to work with, identified a key shortcoming in my early work. A form of technical debt I had yet to address. He dedicated his master's thesis to tackling this problem. The solution turned out to be far more complex than initially expected, ultimately requiring the combined efforts of both myself and several master students after him. Still, in guiding Phu and through his fresh perspective, I uncovered some of the most important insights presented in this dissertation. The challenge was inherited by Deniz Yıldırım who made great advancements, and finally, there was the brilliant team of Koen Wösten and Stijn Coppens. Both are incredibly talented engineers who devoted their master's theses to actually making the system work completely end-to-end, bringing their own expertise to fill gaps in my own knowledge and skill set. Supervising the two of them was one of the most fun and rewarding experiences I've had throughout my PhD. This collaboration is etched into my brain as a model for the kind of collaborative spirit I hope to foster in all future undertakings. Beyond these special mentions, I am also deeply grateful to all the other master students I had the pleasure to supervise throughout these years: Naveen Jakka, Jelger Lemmers, Gijsbert Maan, Tamas Mayer, David Zwart, Teun Buijs, Kilian van Berlo, Koen Peelen, Berkin Zeybekoğlu, Sarthak Singhal, Joris Gravesteijn, and Quinten van Opstal. Each of them brought their own unique perspectives, skills, and energy to the work we shared, and I learned something new from each of them.

Beyond the students I had the privilege to supervise, I was also fortunate to engage with many peers and colleagues throughout these years. The university can be a remarkable place not just for acquiring academic skills, but for learning about the world. At TU Delft, people from all corners of the globe come together in pursuit of knowledge, and the vast majority are unafraid to challenge ideas. This creates a lively brewing pot of different eloquently phrased perspectives. Through countless such conversations, I learned a great deal from my colleagues in both the Networked Systems and Embedded Systems groups. Among my colleagues are two people I collaborated with that left a great mark on me as both a researcher and a person. Dr. Vijay Rao is sharp, methodical, and not easily convinced. Precisely the kind of person you want nearby when the other two voices in the room are VP and me, each lost in our own clouds, believing our simple ideas can topple mountains. His ability to bring us back to earth and refocus us on what actually matters cannot be overstated. Then there is Dr. Vineet Gokhale. Over the course of many collaborations, I came to deeply value his patience and generous spirit. He was there to discuss my ideas even when I was not yet capable of articulating them clearly, and he always brought a positive energy to our work. This was especially important during the long periods of isolation brought on by the pandemic. Finally, I would like to thank my promotor, Prof. Koen Langendoen. He is sharp, methodical, not easily convinced, and perhaps a touch cynical. While my research style might resemble VP's more, I am also Dutch, and place great value on Koen's perspective. His feedback, particularly on this dissertation, was indispensable. Without his guidance, dear reader, you would have been subjected to some rather incoherent lines of thought present in my early drafts.

My mother used to quote Simone Weil: “To be rooted is perhaps the most important and least recognized need of the human soul.” She is right, of course, at least for me. Having a place and a people where I truly belong has always felt like both a refuge in difficult times and the very thing that gives meaning to everything else. Throughout my PhD, there was always a home with my mother, Loes Hegger, and my sister, Désirée Kroep. Together with my father, when he was still alive, they taught me what it means to belong. Temporary places to belong were also generously offered by those I shared a home with during these years. Lukas, Mees, Piyush, Mathan, Jorge, and Pavlos were not just roommates, but companions that I have fond memories off. Then there were the people who provided homes of a different kind, through their listening ear, valuable insights, or the simplicity of enjoying life together. There are too many to name here, so I will have to thank you all in person. Let that be a welcome excuse to spend more time together.

And then, at the very tail end of my PhD, I stumbled upon the home I had been searching for all along when I fell in love with my now wife, Karin Merckens-Kroep. Together we had a son, Iep, and through him I have come to understand what this academic journey has truly prepared me for. I now have the task of creating a safe space for this little one to explore the world with his own chaotic, young energy. We will collaborate on the imaginative, poorly formulated ideas he dreams up, and I will be there to challenge him with new perspectives. But most important of all, I will be someone he can always come home to.

To what comes next,
Kees Merckens-Kroep

List of Publications

The following publications resulted from the research presented in this dissertation:

1. J. P. Verburg, **H. J. C. Kroep**, V. Gokhale, R. Venkatesha Prasad, and V. Rao. *Setting the Yardstick: A Quantitative Metric for Effectively Measuring Tactile Internet.* In *Proceedings of IEEE INFOCOM*, 2020.

2. **H. J. C. Kroep**, V. Gokhale, J. Verburg, and R. Venkatesha Prasad. *ETVO: Effectively Measuring Tactile Internet With Experimental Validation. IEEE Transactions on Mobile Computing*, 2023.

3. V. Gokhale, **H. J. C. Kroep**, V. S. Rao, J. Verburg, and R. Yechangunja. *TIXT: An Extensible Testbed for Tactile Internet Communication. IEEE Internet of Things Magazine*, vol. 3, no. 1, pp. 32–37, 2020.

4. **H. J. C. Kroep**, V. Gokhale, A. Simha, R. Venkatesha 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)*, pp. 199–208, 2023.

5. V. Gokhale, **H. J. C. Kroep**, R. Venkatesha Prasad, B. Bellalta, and F. Dressler. *ViTaLS—A Novel Link-Layer Scheduling Framework for Tactile Internet Over Wi-Fi. IEEE Internet of Things Journal*, vol. 10, no. 11, pp. 9917–9927, 2023.

6. **H. J. C. Kroep**, V. Gokhale, and R. Venkatesha Prasad. *Blind Spots of Objective Measures: Exploiting Imperceivable Errors for Immersive Tactile Internet.* In *Proceedings of the ACM/IEEE 13th International Conference on Cyber-Physical Systems (ICCPS)*, pp. 1–10, 2022.

7. **H. J. C. Kroep**, P. Makridis, J. Huidobro, K. Wösten, D. Choudhary, N. Gnani, T. V. Prabhakar, S. Coppens, K. van Berlo, and R. Venkatesha Prasad. *Utilizing Operator Intent for Haptic Teleoperation Under High Latencies.* Accepted for publication in *IEEE Transactions on Mobile Computing*, to appear in 2025.

8. **H. J. C. Kroep**, S. Coppens, K. Wösten, A. Bhattacharjee, and R. R. Venkatesha Prasad. *Breaking the Latency Barrier: Practical Haptic Bilateral Teleoperation over 5G.* In *Proceedings of the ACM/IEEE 16th International Conference on Cyber-Physical Systems (with CPS-IoT Week)*, pp. 1–11, 2025. Runner-up Best Paper Award.

“Surprisingly, it is possible.
We will be able to reach out
and touch something
anywhere in the world.”