TIM
A Novel Quality of Service Metric for Tactile Internet

Kroep, H. J.C.; Gokhale, V.; Simha, A.; Prasad, R. R.Venkatesha; Rao, V. S.

DOI
10.1145/3576841.3585917

Publication date
2023

Document Version
Final published version

Published in
ICCPS 2023 - Proceedings of the 2023 ACM/IEEE 14th International Conference on Cyber-Physical Systems with CPS-IoT Week 2023

TIM: A Novel Quality of Service Metric for Tactile Internet

H.J.C. Kroep
V. Gokhale
H.J.C.Kroep@tudelft.nl
V.Gokhale@tudelft.nl
Delft University of Technology
Delft, Netherlands

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

V.S. Rao
Vijay.Rao@cognizant.com
Cognizant Technology Solutions
Amsterdam, Netherlands

ABSTRACT

Tactile Internet (TI) envisions communicating haptic sensory information and kinesthetic feedback over the network and is expected to transfer human skills remotely. For mission-critical TI applications, the network latency is commonly mandated to be between 1-10 ms, due to the sensitivity of human touch, and the packet delivery ratio to be 99.99999%, failing which can lead to catastrophic outcomes. However, with humans-in-the-loop, their dexterity and adaptability to varying responses to stimuli under different network conditions, measuring the performance of a TI session only with latency and packet losses are insufficient and presents an incorrect representation of the experience of the TI application.

To develop an objective measure of the quality of TI sessions, we propose a framework that models TI applications as networked control systems, including humans-in-the-loop. We derive a closed-form expression for measuring the difference between the application performance in ideal and non-ideal network conditions. Based on Weber's law of *Just Noticeable Difference*, we provide a metric called TIM to estimate the impact of the network on haptic feedback. We implemented TIM on multiple applications on a TI testbed to show that our approach is feasible and TIM strongly follows real subjective measurements. Further, we propose a channel compensation spring based on TIM, to alleviate the network conditions' negative effects. We demonstrate the efficacy of the channel compensation spring in improving the user experience. We also present implementation notes for TI application developers.

CCS CONCEPTS

• **Networks → Network performance evaluation**.

KEYWORDS

Tactile internet, user experience, QoS, teleoperation

ACM Reference Format:
H.J.C. Kroep, V. Gokhale, A. Simha, R.R. Venkatesha Prasad, and V.S. Rao. 2023. TIM: A Novel Quality of Service Metric for Tactile Internet. In *ACM/IEEE 14th International Conference on Cyber-Physical Systems (with CPS-IoT Week 2023) (ICCPS '23), May 9–12, 2023, San Antonio, TX, USA*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3576841.3585917

ICCPS '23, May 9–12, 2023, San Antonio, TX, USA

ACM ISBN 979-8-4007-0036-1/23/05…$15.00
https://doi.org/10.1145/3576841.3585917

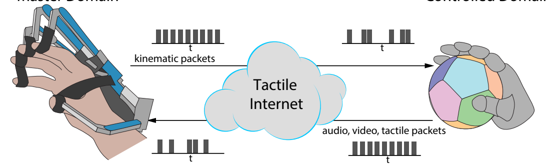


Figure 1: A typical Tactile Internet system highlighting the master and controlled domains, and network characteristics.

1 INTRODUCTION

Tactile Internet (TI) is a networking paradigm that envisions facilitating skill transfers or teleoperations between human operators and remote robots [8] over the Internet. TI is realised by connecting haptic sensors to actuators and enabling feedback from actuators to humans thus enhancing the level of immersion. TI is expected to revolutionise how we interact with remote environments.

Use-cases. As depicted in Fig. 1, an operator in the *master domain* steers a teleoperator in the *controlled domain* by transmitting the kinematic signals such as position, velocity, and torque, to the controlled domain where the teleoperator mimics the operator's actions, also with audio and video feedback to the operator. The sense of touch enables a variety of use-cases of which the most celebrated one is telesurgery – a surgeon performing delicate medical procedures over a network with the requirement of a comparable level of precision and speed to conventional surgery. Several use-cases can be conceived in the domains of industrial automation, edutainment, remotely working in hazardous environments, or even inter-personal communications – many grandparents missing sense of touch due to COVID-19.

The sense of touch. Transporting sense of touch brings considerable challenges even to the mature field of networking. In audio-video applications, a high latency causes a noticeable lag, but it does not hamper one's ability to converse as hearing ourselves while speaking serves as feedback. Hence we are delay-tolerant (Satellite call is a glorious example of our adaptability to long latency). On the other hand, continuous *haptic feedback* must come from the remote environment to continue effective teleoperation. As a simple example, imagine that operator is holding a heavy object and the downward force transmitted from the controlled domain cannot be taken away even for a moment, doing so would cause the operator to execute an unnecessary or even counterproductive correction.

Pitfalls in TI characterisation. In TI applications, ensuring a high-quality teleoperation experience is paramount. The operator should get a fine-grained control of the remote system without any perceivable disturbances. This demands addressing an unprecedented challenge. To enable TI applications, it is commonly considered that *Ultra Reliability and Low Latency Communications* (URLLC) is necessary. The IEEE TI standards committee [15] compiles the absolute requirements for mission-critical TI applications with a round trip latency of 1-10 ms, and a reliability (packet delivery ratio) of five 9s (i.e., 99.99999%). The committee suggests that non-compliance to these URLLC network conditions cannot guarantee the quality of TI sessions, and could lead to catastrophic outcomes. However, an understanding of the effect of not meeting the requirements, or the balance between the different indicators is unclear. For example, it is unclear if a network with 5 ms latency and 80 % packet loss (which is 20 % reliability) is better or worse than another network with 8 ms latency and 20 % packet loss.

The need for a TI-specific metric. The important question, now, is to understand the effects of network on the TI applications, specifically while facilitating haptic feedback. Several metrics exist for the performance characterization of TI solutions [6, 7, 10]. Quality of Service (QoS) metrics capture network behavior in terms of latency and packet loss. However, there is no known way to translate these metrics to their impact on haptic feedback. Additionally, a few objective Quality of Experience (QoE) metrics have also been proposed [4, 11, 21] for modeling user experience. Firstly, due to the subjectivity of human perception and response, such models are hard to be generalized. Additionally, the QoE measurements can only be collected offline. Further, TI applications have different requirements depending on the use-case being performed. None of the existing metrics (QoS/QoE) take this into consideration.

Metrics used for TI, hitherto, fall short because none of them consider the *stimuli-response* relationship associated with humans interacting with physical objects located remotely. This further compounds the issues in understanding the impact of latency and reliability in TI. Over-provisioning may help application developers but will be detrimental to the underlying network in the long run. Novel frameworks that consider these aspects are essential to establish a TI session's experience systematically. Network services cannot be provisioned successfully on a large scale without an objective metric to measure the quality. Therefore, in this work we attempt to fill this gap in provisioning TI applications.

Our approach. We set out to create the first metric that captures the effect of the network on haptic feedback. We address the problem from networked control systems view point to model human interactions with physical objects. We use a representative model and 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 Tactile Internet Metric (TIM) that estimates the amount of undesired haptic feedback introduced by the network. The TIM score can be used to 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 channel compensation spring that adjusts the application parameters to satisfy a target TIM score.

Contributions. Our contributions are enumerated below:

❶ We first provide a simplified generic framework of the TI applications. Specifically, we model TI applications as networked control systems. To the best of our knowledge, this is the first work to adopt a human-in-the-loop control-theoretic approach for designing a TI metric. (Section 3)

❷We also derive an expression for network-induced delay using a Markov model (Section 3.3).

❸Using the model, we theoretically derive a real-time metric, TIM, that can provide a numerical value to quantify the quality of a TI application. The TIM is designed in such a way that it always compares with the TI application over ideal network conditions (no packet loss and delay) (Section 4).

❹We propose a novel method to tune a channel compensation spring using TIM to adjust to the network conditions. We have implemented and tested this on two TI applications (Section 4.3).

❺Through the design of realistic TI experiments, we perform both objective and subjective evaluations of our TIM and we show that subjective user experience follows the metric (Section 6).

We also provide guidelines on how our framework can be leveraged for meaningful characterization and also how to improve further to expand its applicability.

2 RELATED WORKS

2.1 TI quality measurement

QoS metrics. In line with the URLLC requirements, the most commonly used QoS parameters for performance evaluation of TI solutions are network delay and packet loss [1, 6, 7, 10, 12]. Based on these QoS metrics, solutions to improve TI session quality are also proposed. Applications are evaluated using the same metrics. It is worth emphasizing, that apart from improving the network performance, it is crucial to also understand the impact on user experience.

QoE metrics. To directly capture the user's experience, it is natural to use subjective grades where a participant is asked to rate the experience on a predefined scale. This method captures the user experience in the best possible way [3, 13, 17]. However, gathering user responses over a large class of TI use-cases is cumbersome and time-consuming. Further, the subjectivity included in the user responses makes the results hard to reproduce. To alleviate this problem, objective QoE metrics based on signal errors have been used in [4, 11, 21]. While these works attempt to draw a correlation between user experience and signal deterioration, it is hard to use and compute them in real-time for a large class of TI applications. Recently, Quality of Control (QoC) has been proposed, which has a promising approach in that it considers an underlying application [19]. However, this method only captures TI system performance for a very specific set of assumptions, including a tactile latency requirement of 125 ms, which is incorrect.

2.2 Networked control systems

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 [16, 23, 26]. 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 system performance has been studied in [2, 14, 18, 20, 25]. 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

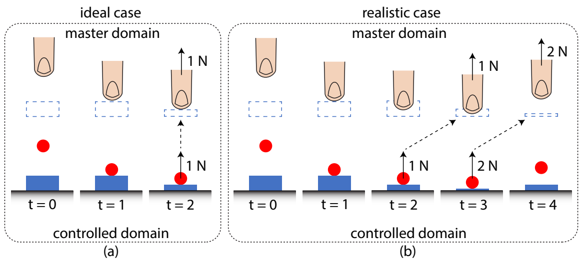


Figure 2: Illustration of the detrimental effects of the network on TI interaction. The operator intends to press a switch (blue block) through the teleoperator (red circle) using a tactile glove virtually (dotted blue block). (a) In the ideal case, force is experienced right at the instant of contacting the object. (b) In a realistic network case, the force feedback is transported with a variable delay and causes significant performance degradation.

and adaptive (gain tuning) approaches are presented in [9]. 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 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.

3 THE PROBLEM AND SYSTEM MODELING

3.1 The problem

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 Fig. 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 Fig. 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 Fig. 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 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.

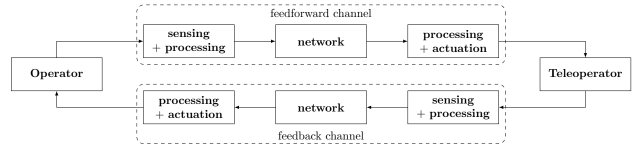


Figure 3: An overview of the TI system model. The feedback and feedforward channels include modules that influence the performance.

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.

3.2 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 Fig. 3. The *channel* comprises all modules starting from sensors in the master domain to the actuators in the controlled domain.[1] 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 Sec. 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 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 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.

[1]This is in contrast with the standard network parlance where "channel" refers only to communication links.

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 Sec. 4.

Signal-oblivious delay indicator (τ_{QoS}). We consider three QoS indicators: latency, packet loss, and packet rate. In TI literature, these performance indicators are measured and treated separately [6, 7, 10, 12]. 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,

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

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, Eq. (1), in the online appendix.

Signal-aware delay indicator (τ_{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 Sec. 2. Recently, a framework called Effective Time- and Value-Offset (ETVO) [24] 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 [24], which yields,

$$\tau_{\text{ETVO}} = \frac{1}{N}\sum_{k=0}^{N} ETO[k], \tag{2}$$

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

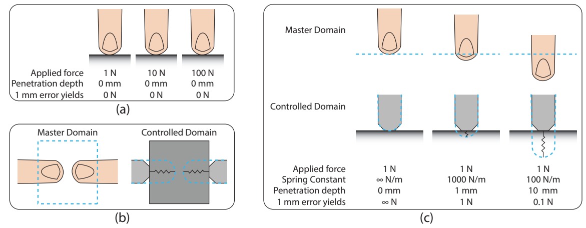


Figure 4: (a) Direct interaction between a finger and an infinitely rigid surface. Regardless of how hard the finger presses, it will not deform the surface. (b) An illustration of the problem arising when performing a grasping motion. To have a grip on the object through imaginary strings, the fingers 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.

3.4 Tactile interaction model

Let us first consider a regular physical interaction with a finger and a highly rigid, fixed surface (Fig. 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 [22]. 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 Fig. 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 Fig. 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 Fig. 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 [5] 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,

$$\begin{bmatrix} sX_1(s) - x_1(0) \\ sX_2(s) - x_2(0) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \frac{-k}{m}e^{-\tau s} & 0 \end{bmatrix} \begin{bmatrix} X_1(s) \\ X_2(s) \end{bmatrix} + \begin{bmatrix} 0 \\ -g \end{bmatrix} U(s),$$
$$Y(s) = \begin{bmatrix} 1 & 0 \end{bmatrix} X(s), \tag{3}$$

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,

$$Y(s) = \frac{\dot{x}_{\text{impact}} s - g}{s^3 + \frac{k}{m} s e^{-\tau s}}. \tag{4}$$

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

$$x_{\text{ideal}}(t) = \mathcal{L}^{-1}\{Y(s)\} = \frac{V}{\sqrt{\frac{k}{m}}} \sin\left(\sqrt{\frac{k}{m}}t\right) + \frac{gm}{k}\left(\cos\left(\sqrt{\frac{k}{m}}t\right) - 1\right). \tag{5}$$

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

$$\hat{Y}(s) = \frac{\beta_0 + \beta_1 s + \cdots + \beta_{n-2} s^{n-2}}{\alpha_0 + \alpha_1 s + \cdots + \alpha_{n-1} s^{n-1} + s^n} \approx Y(s), \tag{6}$$

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,

$$x(t) = \mathcal{L}^{-1}\{\hat{Y}(s)\} = \hat{C} e^{\hat{A}t} \hat{B}, \tag{7}$$

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 Eq. (1) and Eq. (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 Eq. (7) is given below. Note that Line-4 directly implements Eq. (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 1: Matlab function that calculates Eq. (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
```

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.

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.

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

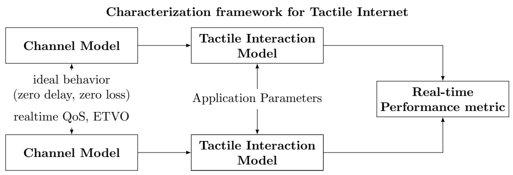


Figure 5: A schematic to show how the channel and tactile interaction models are used to estimate the departure from ideal TI system behavior to yield TIM measurements.

measurements. This will aid in provisioning network and system resources on the fly to meet the necessary application requirements.

4.2 Design and analysis

Fig. 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. Eq. (7) allows us to project the trajectories of the ideal and realistic TI system for evaluating the system performance in real-time.

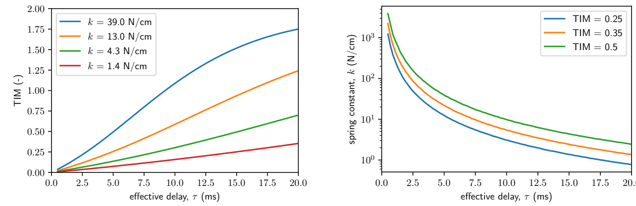
To estimate the effect of the TI system on user perception, we rely on Weber's law of Just-Noticeable-Difference (JND) [13]. We can use this to conclude that the system performs adequately if $\frac{\Delta I}{I}$ < JND, where I is the intensity, and JND is a threshold. In our case, we take I as the amount 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 object from the surface in the ideal scenario, which can be derived from Eq. (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

$$\text{TIM}(k, \dot{x}_{\text{impact}}, m, \tau) = \sqrt{\frac{\int_0^{T_{\text{exit}}} \left(x_{\text{ideal}}(t) - x(t)\right)^2 dt}{\int_0^{T_{\text{exit}}} x_{\text{ideal}}(t)^2 \, dt}}, \quad (8)$$

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 Fig. 6(a), we show the TIM scores for a wide



(a) **(b)**

Figure 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.

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 Eq. (8), we can compute k for a given τ (and vice-versa) and TIM score. This is shown in Fig. 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.

4.3 Channel compensation spring

The spring constant k as modeled in Section. 3.4 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

$$\frac{1}{k_{\text{total}}} = \frac{1}{k} + \frac{1}{k_c}, \quad (9)$$

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 Fig. 6. Then using Eq. (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

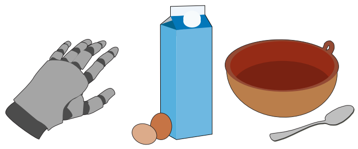


Figure 7: An example of a TI application that can be characterized and tuned using our proposed metric TIM.

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.

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 Fig. 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 Fig. 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 4.3. Firstly, Fig. 6(b) should be used to determine the maximum acceptable k_{total}. Based on this, k_c is calculated according to Eq. (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.

6 PERFORMANCE EVALUATION

6.1 Experimental setup

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

6.1.1 ***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 Fig. 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 Fig. 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 Fig. 8(b).

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

6.1.2 ***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,

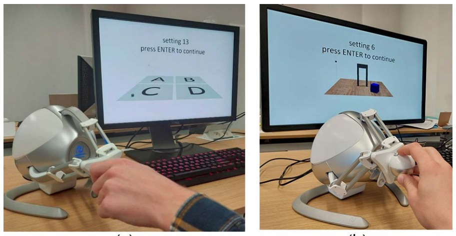


Figure 8: TI experimental setup in our work showing the human participant: (a) in the virtual environment interacting with surfaces A, B, C, and D that are having different spring constants indicating different types of surface hardness. (b) in the virtual environment interacting with a cube that can be pushed through a narrow gate.

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 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.

Table 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

6.2 Objective evaluation

For objective evaluations, we use only Bounce application. 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 Fig. 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

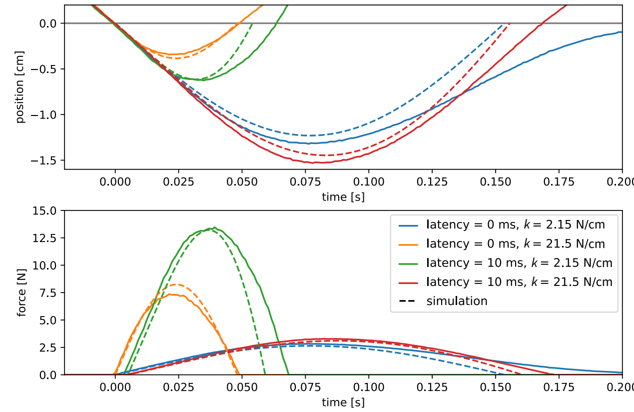


Figure 9: Temporal variation of haptic device trajectory and force experienced as a function of latency and k compared with the simulated measurements showing the efficacy of our theoretical approximations.

Eq. (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 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.

6.3 Subjective evaluation

6.3.1 ***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 Fig. 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 Fig. 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 low k, to be supported by a given TI system performance. While these insights provide a preliminary understanding of the underlying dynamics,

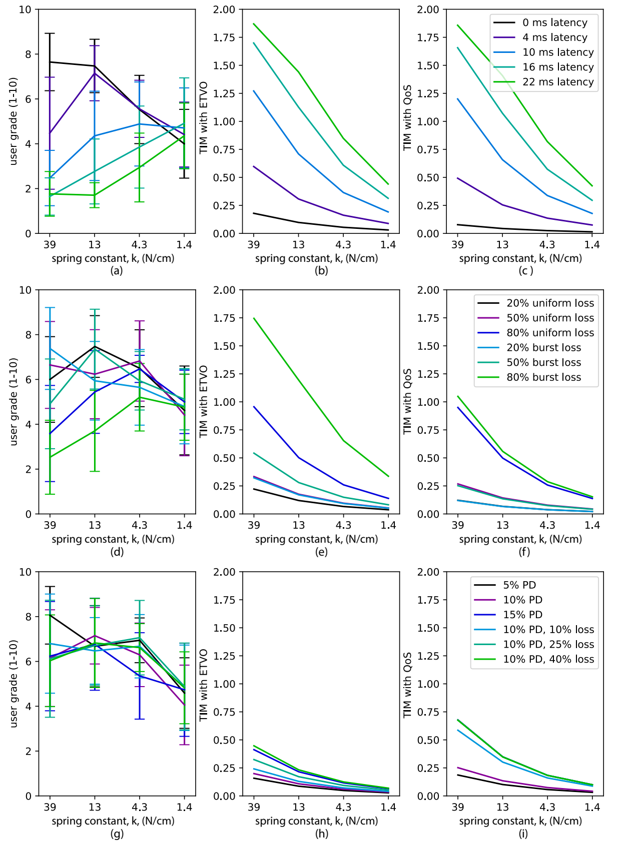


Figure 10: User grades and corresponding TIM scores across different network latency, packet loss, and perceptual Deadband settings by QoS and ETVO models for the Bounce application.

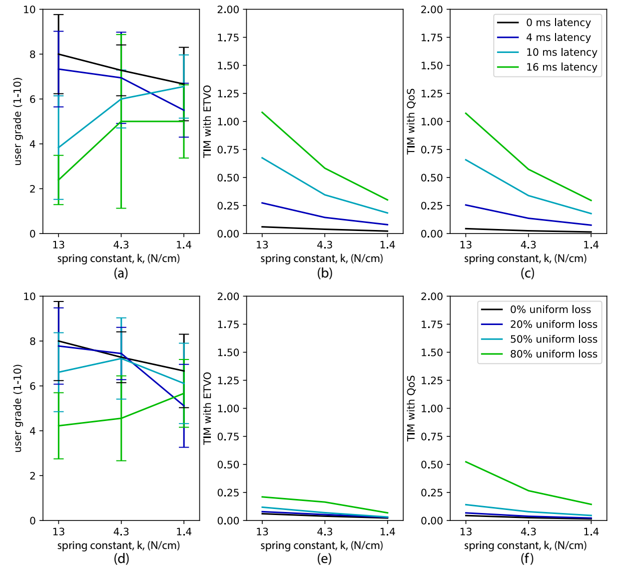


Figure 11: User grades and corresponding TIM scores across different network latency, packet loss, and perceptual Deadband settings by QoS and ETVO models for the Slide application.

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 Fig. 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 Fig. 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.

6.3.2 ***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 Fig. 11(a-c) with Fig. 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 Fig. 11(d-f) and Fig. 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 Fig. 11(a) and 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.

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.

7 CONCLUSIONS

Tactile Internet (TI) applications will be the next frontier for networked applications. In this work, 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 behavior 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 which enabled us to evaluate the TI performance.

A novel mathematical model was developed to obtain 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. We found a strong correlation between network settings, user grades, and the identified spring constant through human subjective experiments. 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 as 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.

REFERENCES

[1] Konstantinos Antonakoglou, Maliheh Mahlouji, and Toktam Mahmoodi. 2021. Bilateral Teleoperation Performance Model for Network Resource Management. *IEEE Access* 9 (2021), 29106–29117.
[2] Mohsen Barforooshan, Milan Stefan Derpich, Photios A Stavrou, and Jan Østergaard. 2020. The effect of time delay on the average data rate and performance in networked control systems. *IEEE Trans. Automat. Control* (2020).
[3] Amit Bhardwaj, Subhasis Chaudhuri, and Onkar Dabeer. 2014. Design and analysis of predictive sampling of haptic signals. *ACM Transactions on Applied Perception (TAP)* 11, 4 (2014), 1–20.
[4] Rahul Chaudhari, Eckehard Steinbach, and Sandra Hirche. 2011. Towards an objective quality evaluation framework for haptic data reduction. In *2011 IEEE World Haptics Conference*. IEEE, 539–544.
[5] Ben M Chen, Zongli Lin, and Yacov Shamash. 2004. *Linear systems theory: a structural decomposition approach.* Springer Science & Business Media.
[6] Burak Cizmeci, Xiao Xu, Rahul Chaudhari, Christoph Bachhuber, Nicolas Alt, and Eckehard Steinbach. 2017. A Multiplexing Scheme for Multimodal Teleoperation. *ACM Trans. Multimedia Comput. Commun. Appl.* 13, 2, Article 21 (April 2017), 28 pages. https://doi.org/10.1145/3063594
[7] Mohamad Eid, Jongeun Cha, and Abdulmotaleb El Saddik. 2010. Admux: An adaptive multiplexer for haptic–audio–visual data communication. *IEEE Transactions on Instrumentation and Measurement* 60, 1 (2010), 21–31.
[8] Gerhard P Fettweis. 2014. The tactile internet: Applications and challenges. *IEEE Vehicular Technology Magazine* 9, 1 (2014), 64–70.
[9] Mahmoud Gamal, Nayera Sadek, Mohamed RM Rizk, and Ahmed K Abou-elSaoud. 2016. Delay compensation using Smith predictor for wireless network control system. *Alexandria Engineering Journal* 55, 2 (2016), 1421–1428.
[10] Vineet Gokhale, Jayakrishnan Nair, and Subhasis Chaudhuri. 2017. Congestion Control for Network-Aware Telehaptic Communication. *ACM Trans. Multimedia Comput. Commun. Appl.* 13, 2, Article 17 (March 2017), 26 pages. https://doi.org/10.1145/3052821
[11] Rania Hassen and Eckehard Steinbach. 2018. HSSIM: An objective haptic quality assessment measure for force-feedback signals. In *IEEE International Conference on Quality of Multimedia Experience.*
[12] Peter Hinterseer, Sandra Hirche, Subhasis Chaudhuri, Eckehard Steinbach, and Martin Buss. 2008. Perception-based data reduction and transmission of haptic data in telepresence and teleaction systems. *IEEE Transactions on Signal Processing* 56, 2 (2008), 588–597.
[13] Peter Hinterseer, Eckehard Steinbach, Sandra Hirche, and Martin Buss. 2005. A novel, psychophysically motivated transmission approach for haptic data streams in telepresence and teleaction systems. In *IEEE International Conference on Acoustics, Speech, and Signal Processing.*
[14] Sandra Hirche, Tilemachos Matiakis, and Martin Buss. 2009. A distributed controller approach for delay-independent stability of networked control systems. *Automatica* 45, 8 (2009), 1828–1836.
[15] Oliver Holland, Eckehard Steinbach, R Venkatesha Prasad, Qian Liu, Zaher Dawy, Adnan Aijaz, Nikolaos Pappas, Kishor Chandra, Vijay S Rao, Sharief Oteafy, et al. 2019. The IEEE 1918.1 "Tactile Internet" Standards Working Group and its Standards. *Proc. IEEE* 107, 2 (2019), 256–279.
[16] Magdi S Mahmoud and Mutaz M Hamdan. 2018. Fundamental issues in networked control systems. *IEEE/CAA Journal of Automatica Sinica* 5, 5 (2018), 902–922.
[17] Evelyn Muschter, Andreas Noll, Jinting Zhao, Rania Hassen, Matti Strese, Basak Guelecyuez, Shu-Chen Li, and Eckehard Steinbach. 2021. Perceptual Quality Assessment of Compressed Vibrotactile Signals through Comparative Judgment. *IEEE Transactions on Haptics* (2021).
[18] Payam Naghshtabrizi and Joao P Hespanha. 2006. Stability of networked control systems with variable sampling and delay. In *Allerton Conf. on Communication, Control, and Computing*. Citeseer.
[19] Kurian Polachan, TV Prabhakar, Chandramani Singh, and Deepak Panchapakesan. 2019. Quality of control assessment for tactile cyber-physical systems. In *IEEE International Conference on Sensing, Communication, and Networking (SECON).*
[20] Asif Šabanović, Kouhei Ohnishi, Daisuke Yashiro, Nadira Šabanović, and Eray A Baran. 2010. Motion control systems with network delay. *Automatika* 51, 2 (2010), 119–126.
[21] N Sakr, ND Georganas, and J Zhao. 2007. A perceptual quality metric for haptic signals. In *2007 IEEE International Workshop on Haptic, Audio and Visual Environments and Games*. IEEE, 27–32.
[22] Kenneth Salisbury, Francois Conti, and Federico Barbagli. 2004. Haptics rendering: Introductory concepts. *IEEE computer graphics and applications* 24, ARTICLE (2004), 24–32.
[23] Yodyium Tipsuwan and Mo-Yuen Chow. 2003. Control methodologies in networked control systems. *Control engineering practice* 11, 10 (2003), 1099–1111.
[24] Joseph P Verburg, HJC Kroep, Vineet Gokhale, R Venkatesha Prasad, and Vijay Rao. 2020. Setting the Yardstick: A Quantitative Metric for Effectively Measuring Tactile Internet. In *IEEE INFOCOM.*
[25] Zhichun Yang and Daoyi Xu. 2007. Stability analysis and design of impulsive control systems with time delay. *IEEE Trans. Automat. Control* 52, 8 (2007), 1448–1454.
[26] Lixian Zhang, Huijun Gao, and Okyay Kaynak. 2012. Network-induced constraints in networked control systems—A survey. *IEEE transactions on industrial informatics* 9, 1 (2012), 403–416.