ETVO

Effectively Measuring Tactile Internet With Experimental Validation

Kroep, Kees; Gokhale, Vineet; Verburg, Joseph ; Prasad, R. Venkatesha

DOI
10.1109/TMC.2023.3246659

Publication date
2024

Document Version
Final published version

Published in
IEEE Transactions on Mobile Computing

ETVO: *Effectively* Measuring Tactile Internet With Experimental Validation

Kees Kroep, Vineet Gokhale, Joseph Verburg, and R. Venkatesha Prasad

***Abstract*—The next frontier in communications is *teleoperation* – manipulation and control of remote environments with haptic feedback. Compared to conventional networked applications, teleoperation poses widely different requirements, ultra-low latency (ULL) is primary. Realizing ULL communication demands significant redesign of conventional networking techniques, and the network infrastructure envisioned for achieving this is termed as *Tactile Internet* (TI). The design of meaningful performance metrics is crucial for seamless TI communication. However, existing performance metrics fall severely short of comprehensively characterizing TI performance due to their inability to capture how well sensed signals are reproduced. We take Dynamic Time Warping(DTW) as the basis of our work and identify necessary changes for characterizing TI performance. Through substantial refinements to DTW, we design *Effective Time- and Value-Offset (ETVO)* – a new method for measuring the fine-grained performance of TI systems. Through an in-depth objective analysis, we demonstrate the improvements of ETVO over DTW. Through subjective experiments, we demonstrate that existing QoS and QoE methods fall short of estimating the TI session performance accurately. Using subjective experiments, we demonstrate the behavior of the proposed metrics, their ability to match theoretically derived performance, and finally, their ability to reflect user satisfaction in a practical setting.**

***Index Terms*—Tactile Internet, user experience, QoS, URLLC.**

I. Introduction

THE COVID-19 pandemic has made us realize the power of the Internet yet again by seamlessly connecting people located remotely through audio-video interactions. *Tactile Internet (TI)* promises to advance this level of immersion by augmenting a new modality of interaction – *haptic feedback*. This enables teleoperation, which is a primary driving force for the realization of Industry 4.0 revolution [2], [3]. One of the most well-known use cases of TI is telesurgery, where a patient in the controlled domain can be operated upon by a surgeon (*operator*) in the master domain, from a distant location as effectively as in a conventional surgery. The kinematic information (position and orientation) of the surgeon's arm is replicated by a remote robot arm (*teleoperator*), while simultaneously, haptic and video information from the controlled domain are fed back to the surgeon. This *transportation of skills* allows the surgeon to perform the medical procedure as if he/she is physically present in the controlled domain. A schematic representation of TI is shown in Fig. 1.

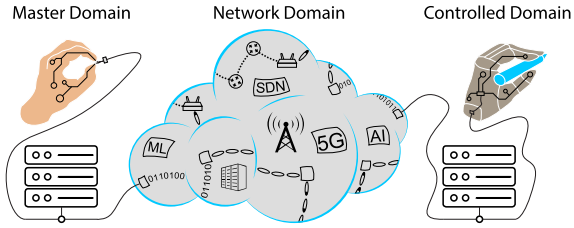

Fig. 1. A schematic representation of Tactile Internet showing the master and controlled domains.

Management of disaster-struck areas (for example, Fukushima Daiichi disaster site) remotely, and telerepair and telemaintenance of machinery, and also other sectors like education, health, and manufacturing [4] will benefit from TI. Deploying TI will significantly save time, money, lives, and in many cases, enable applications that are otherwise not feasible. Such complex applications cannot be performed by autonomous robots alone and thus require human expertise.

While real-time communications, such as audio and video conferencing, have long existed, the sense of touch brings considerable challenges beyond what is faced for audio-video applications. For example, a significant delay can be noticeable during a conference meeting, particularly during dialogue. However, speaking ability is unaffected because one's voice can be directly used as feedback regardless of technology. TI applications must provide active interactions with a feedback loop, typically involving continuous force feedback. TI applications require ultra-low latency to perform precise actions. For example, a surgeon wants to make a shallow cut in tissue. A delay could cause the surgeon to cut deeper than intended, causing catastrophic consequences. Besides delay, high reliability is typically required to ensure negligible amounts of inaccuracy. For example, an inaccuracy could cause a hard tissue to feel like a soft tissue to the surgeon, causing a misjudgment.

Further, different TI applications have different requirements. For example, the requirements of interpersonal communication will be more relaxed than life-critical applications like telesurgery. Hence the requirements are application specific. The IEEE TI standards committee has provided specific requirements for mission-critical TI applications: a round trip

Manuscript received 2 July 2021; revised 15 February 2023; accepted 15 February 2023. Date of publication 20 February 2023; date of current version 5 February 2024. This work is a significant extension of our earlier work in published in IEEE INFOCOM 2020 [DOI: 10.1109/INFOCOM41043.2020.9155540]. Recommended for acceptance by J. Deng. *(Corresponding author: Kees Kroep.)*

The authors are with the Embedded, Networked Systems, Delft University of Technology, 2628 CD Delft, The Netherlands (e-mail: k.kroep1@gmail.com; v.gokhale@tudelft.nl; r.r.venkateshaprasad@tudelft.nl).

Digital Object Identifier 10.1109/TMC.2023.3246659

latency of 1-10 ms and reliability of 99.99999%. However, these requirements may not apply to every case for the reasons being: ❶ The underlying application-specific requirements are not considered. ❷ The reliability requirement of 99.99999% lacks sound scientific backing by any significant study for human-in-the-loop systems. In our previous work, we objectively demonstrated a significant improvement in performance by sacrificing reliability for a lower latency [5]. We will demonstrate the aspect of the aforementioned reliability requirement as an overkill further in this paper through subjective evaluation. ❸ Protocols that may relax the constraints on the network, such as prediction and compression, are ignored. For example, with proper prediction techniques, one might be able to tolerate much higher network latency than what is prescribed. ❹ it is impractical to provide a conservative estimate of requirements (1 ms delay and 99.99999% reliability) for every TI application.

With these considerations in mind, we come to the main challenge we consider in this work: *assessment of the quality of a TI session reliably and objectively*. This is crucial for several reasons: ① estimating the quality of a session *a priori* is essential for executing mission-critical applications, ② adapting TI application parameters based on network dynamics, ③ benchmarking novel solutions at various layers of the TI protocol stack.

Measuring TI Performance. The core task of a TI system (comprising sensors, actuators, communication, and computing entities) is to communicate haptic-audio-video modalities. An ideal TI system would recreate the values at the controlled domain identical to the sensed values – in *value/magnitude* and *time* – and vice versa. However, in practice, the reconstructed signal can be degraded in time due to varying delays and in value due to losses. For fine-grained performance analysis, it is crucial to distinguish these offsets as independently as possible, and this is challenging.

Literature provides two types of approaches for measuring TI performance – Quality of Service (QoS) and Quality of Experience (QoE). The QoS metrics are based on standard network performance indicators such as delay, jitter, reliability, and throughput. In the case of TI, the emphasis is on the end-to-end delay (sensing, computation, network, and actuation), for which a commonly stated target is 1 ms [2], [3].

QoS is the de facto metric in characterizing network performance. It would therefore appear as the best candidate to evaluate and compare TI networks. However, we identify three critical problems. First, there is currently no solution to identify how different performance indicators trade off against each other. This means that the only way for QoS to conclusively state that one network is better than the other is if every performance indicator is matched or improved. Second, there is no clear notion of how much performance is good enough for a TI application. This increases the risk of significant resources being invested in improving a performance indicator without affecting the overall application performance. Finally, the QoS metrics cannot indicate how well a signal is reconstructed concerning the sensed signal, which is the core task. However, there is no known way to translate these metrics to their impact on kinematic and force feedback signals.

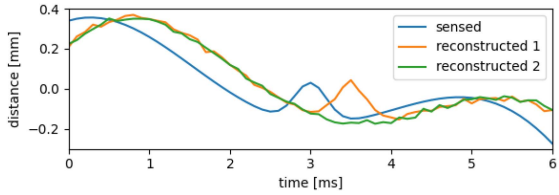

Fig. 2. Illustration of the problem of RMSE when signals have time-offset. Two possible reconstruction signals – 'reconstructed 1' and 'reconstructed 2' – are shown along with the sensed signal. While the shape of 'reconstructed 1' is identical to sensed signal, it is delayed. On the other hand, 'reconstructed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

QoE is a human-centric approach, and it focuses on an objective evaluation of user experience. Ideally, a QoE metric should closely estimate user experience without involving an extensive user study. Several QoE metrics have been designed based on sensed and reconstructed signals. The problem with such approaches is that they cannot distinguish between degradation (offset) in time and value domains. For example, the works in [6], [7] propose RMSE-based metrics. Such solutions do not consider delay and therefore face problems caused by a mismatch between two signals in time, as illustrated in Fig. 2.

Our Contributions. We draw inspiration from a well-known algorithm, Dynamic Time Warping (DTW) [8] to identify signal similarities. Our contributions are as follows.

- We present a detailed analysis on characterizing TI sessions with DTW as a starting point and identify areas of improvement for the stated task (Section III).
- We present a concrete mathematical framework, which we call *Effective Time- and Value-Offset (ETVO)*, that extracts fine-grained time and value-offset between sensed and reconstructed signals of a TI system. To the best of our knowledge, this is the first of its kind work that comprehensively characterizes TI session performance in a system-agnostic manner. (Section IV).
- We propose two novel metrics – average effective time-offset (T_{ETVO}) and average effective value-offset (E_{ETVO}). These metrics can be jointly used to compare the performance of different TI solutions.
- Through objective analysis using a realistic TI setup, we demonstrate the effectiveness of ETVO and its improvement over DTW (Section V).
- To validate ETVO, we conduct human subjective experiments on a realistic TI setup under a wide variety of network settings. We show that the proposed metrics correlate well with the user grades (Section VI).
- Independently, we theoretically derive the expected average delay of the TI sessions and show that it corroborates well with T_{ETVO} measurements (Section VI).
- Through subjective analysis, we also demonstrate that both QoS and QoE methods are insufficient and also show where they fall short in characterizing TI sessions (Section VI).

II. Related Work

A. Quality of Service

Several modular designs of TI systems use QoS metrics for characterizing TI performance. While *Admux*, an adaptive multiplexer for TI proposed in [9], uses all of the above metrics, the multiplexing scheme in [10] focuses on throughput and delay. The haptic codec in [11] focuses on reducing the application throughput by transmitting only the perceptually significant samples. The congestion control scheme in [12] aims to contain delay and jitter within their permissible QoS limits. None of the above works consider measuring the fine-grained offsets between sensed and reconstructed signals, which is one of the crucial aspects of characterizing TI performance.

B. Quality of Experience

Subjective QoE metrics aim to capture the quality of teleoperation by involving human subjects, typically 15-20, and have them subjectively grade their experience. Some works that adopt this approach include [13], [14], [15]. Since this method is cumbersome and resource-intensive, objective QoE metrics that estimate the quality of teleoperation as experienced by the human controller have also been designed. [16] suggests a Proportional Deadband (PD) scheme for exploiting the idea that human perception has a logarithmic relationship with the haptic stimulus. They developed a framework to validate this by using the traditional Peak Signal-to-Noise Ratio (PSNR) of the reconstructed haptic signal as the QoE metric. On the same lines, [7] introduced Haptic Perceptually Weighted Peak Signal to Noise Ratio (HPW-PSNR). [6] followed up on this to propose Perceptual Mean Square Error (PMSE) that maps MSE to the human perceptual domain. Recently, [17] and [18] proposed the Haptic Structure SIMilarity (HSSIM) index and Spectral Temporal SIMilarity (ST-SIM), respectively, to improve the objective estimation of human perception.

C. Dynamic Time Warping

Our work is based on DTW, which is explained in detail in Section III. DTW has been around for a long time, and there are plenty of follow-up works that exist. We highlight a small subset of those works here. Some of the most widely recognized works to build on DTW include Edit Distance on Real sequences (EDR) [19], Edit distance with Real Penalty (ERP) [20], and Longest Common Sub-Sequence (LCSS) [21]. However, they manifest the inherent characteristics of DTW and hence do not address the existing limitations for use in characterizing TI applications.

III. 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

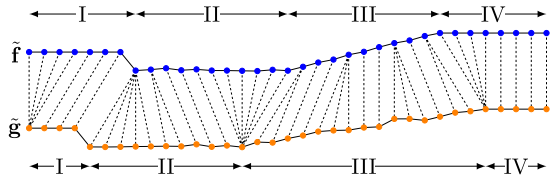

Fig. 3. Example of sample-wise alignment between signals $\tilde{\boldsymbol{f}}$ and $\tilde{\boldsymbol{g}}$ as per DTW. The dashed lines indicate the mapping between the samples.

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.

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.

A. Mathematical Representation

DTW constructs a *warp path* that indicates a sample-wise mapping between two time-series that minimizes their cumulative euclidean distance. 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 Fig. 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:

$$\tilde{\boldsymbol{w}}_0(k) \leq \tilde{\boldsymbol{w}}_0(k+1) \leq \tilde{\boldsymbol{w}}_0(k) + 1,$$
$$\tilde{\boldsymbol{w}}_1(k) \leq \tilde{\boldsymbol{w}}_1(k+1) \leq \tilde{\boldsymbol{w}}_1(k) + 1.$$

2) Boundary:

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

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 $\tilde{\boldsymbol{g}}$ [23]. Hence, we get the error computed by DTW as

$$\text{DTW}(\tilde{\boldsymbol{f}}, \tilde{\boldsymbol{g}}) = \min_{\tilde{\boldsymbol{w}} \in \tilde{\boldsymbol{W}}} \sum_{k=0}^{K-1} \tilde{\delta}(\tilde{\boldsymbol{w}}(k)), \quad (2)$$

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 $\text{DTW}(\tilde{\boldsymbol{f}}, \tilde{\boldsymbol{g}})$ 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 [24], [25].

B. 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.

1) Boundary Conditions Cause Unrealistic Artifacts: The boundary conditions in (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 Fig. 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.

2) 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 Fig. 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 Fig. 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.

IV. 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.

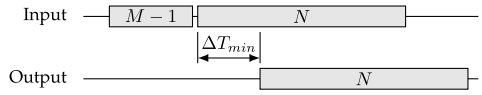

Fig. 4. Illustration of extending the input sequence by $M-1$ samples in ETVO to avoid the start and end artifacts of DTW.

A. 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.

1) Relaxation of Boundary Conditions: We first address the boundary conditions described in Section III-B1 by adjusting the mathematical structure. Let $\boldsymbol{f}$ and $\boldsymbol{g}$ denote slices of the sensed and the reconstructed signals, respectively. For ease of explanation, we use the same notations as in DTW but remove the accent ($\tilde{}$) 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 Fig. 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

$$\text{ETO}[k] = \Delta T_{\text{min}} + w[k]. \tag{3}$$

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. Fig. 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 (3). These new directions are indicated with $C_{\nearrow}$, $C_{\downarrow}$, and $C_{\rightarrow}$ indicating an

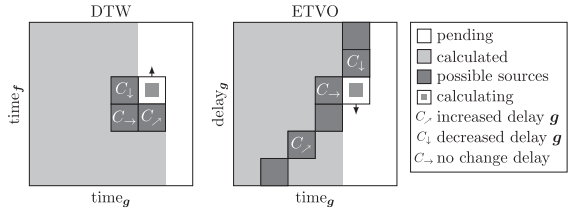

Fig. 5. Illustration of the population of $\boldsymbol{C}$ in both DTW and ETVO. Different types of changes in delay, indicated as $C_\rightarrow, C_\downarrow, C_\nearrow$ are present in both the DTW and ETVO table to show their correspondence. A key difference between DTW and ETVO is that the latter also calculates multiple steps, increasing the possible sources as indicated with dark gray squares.

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 Fig. 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

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

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 (1) is discarded for samples in $\boldsymbol{f}$.

2) Constraining Delay Adjustments: In order to mitigate the issue of unconstrained delay adjustments in DTW (described in Section III-B2), 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 Fig. 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.

$$C_\rightarrow[i,j] = \boldsymbol{C}[i-1,j],$$
$$C_\downarrow[i,j] = \min_{k \subset \mathbb{N}^+} (\boldsymbol{C}[i, j+k] + \sum_{l=1}^{k-1} \delta(i, j+l) + kP_{\text{prop}} + P_{\text{fixed}}),$$
$$C_\nearrow[i,j] = \min_{k \subset \mathbb{N}^+} (\boldsymbol{C}[i-k, j-k] + \sum_{l=1}^{k-1} \delta(i-l, j-l) + kP_{\text{prop}} + P_{\text{fixed}}). \quad (4)$$

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 (4) and is illustrated in Fig. 5.

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.

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

3) 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 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

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

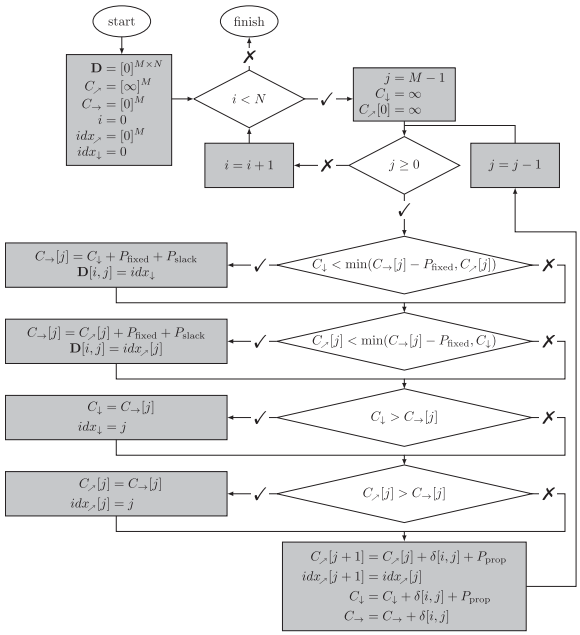

Fig. 6. Flowchart for finding the optimal way of traversing the delay, given the constraints specified for ETO.

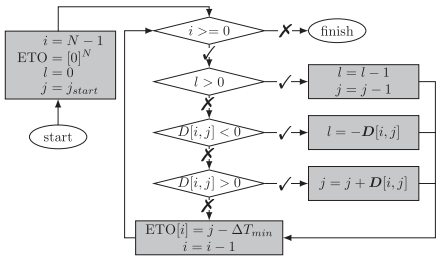

Fig. 7. Flowchart of the backtracking algorithm used to extract the ETO from direction matrix $\boldsymbol{D}$.

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

4) 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 Fig. 6.

The backtracking algorithm is shown in Fig. 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 Fig. 8.

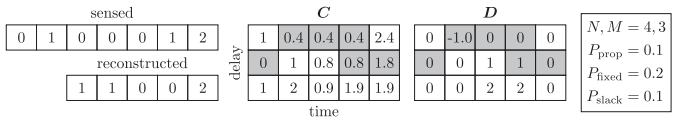

Fig. 8. Numerical example of ETVO including the direction matrix. The gray cells indicate the optimal path chosen by ETVO.

B. 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.

V. Testbed and Objective 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 [26].

A. Standard TI Testbed

A TI testbed was proposed in [26] and has been utilized to support haptic codec standardization activities [27]. The testbed simulates a TI session by having the human participant interact with a 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 [26] to realize a networked TI testbed (described next).

B. Networked TI Testbed

We extend the previous testbed by decoupling the testbed into a master domain module and a controlled domain module, each residing on a different workstation connected by a network. Fig. 9 shows an overview of the entire system. The master domain module senses the position of the haptic device. The controlled domain module houses the simulation of physics aspects. The physics simulation is a substitute for a TI application where the controlled domain module would house a real physical environment. The controlled domain module receives haptic

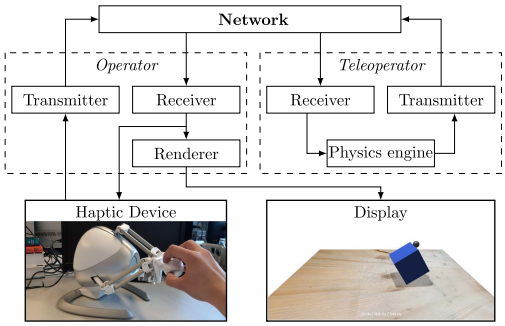

Fig. 9. A schematic overview of our experimental setup. The operator and teleoperator modules run on different computers that are not collocated. The physics engine resides in the controlled domain, resembling a real TI system using Novint Falcon haptic device.

device data through the feedforward channel and feeds it into the physics environment.

As explained in Section I, a realistic TI application is characterized by kinematic information communicated from the master to the controlled domain and haptic-video information back to the master domain. However, one needs to consider the following aspects. ❶ 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 ($\sim$30 ms) but highly sensitive to losses ($<$2%) [28], [29]. ❷ It is known in the literature that asynchrony between haptic/kinematic and video frames can be highly detrimental to the user experience [30]. This implies that users may notice disturbances if the video display is not properly synchronized with the haptic display. 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 controlled 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 master domain. The kinematic information is used to update the visual display of VE in the master 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.

Network Emulation. 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 master and the controlled domains are connected to the university (shared) network and use Ethernet links to connect to a network switch. NetEM is switched on at the master domain

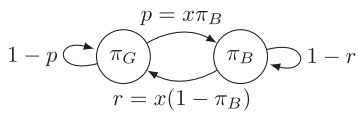

Fig. 10. Gilbert Elliot model and inclusion of scalar x that allows one to change the distribution between bursty and uniform behavior without affecting the average packet loss. π_G and π_B are the average probability of successful and failed packet transmissions, respectively. p and r are the chance of switching states.

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, 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. Fig. 10 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 master domain, which is included in the packets. This adds redundancy to the system which improves performance in most cases.

C. Objective Analysis

The modifications to the basic DTW algorithm proposed in Section IV 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 Fig. 11(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

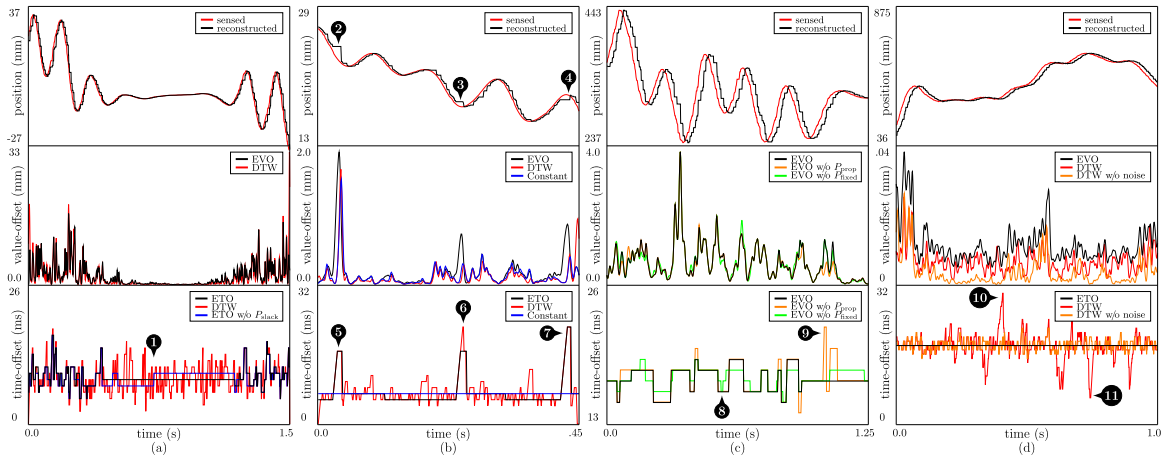

Fig. 11. Comparison of the performances of DTW and ETVO frameworks using a wide variety of experimental setups showing the effects of (a) P_{slack}, (b) uniform packet losses and perceptual deadband (PD) scheme, (c) P_{prop} and P_{fixed}, and (d) addition of noise to the sensed signal.

both edges because of the start and end artefacts. This behavior can be seen in the other examples as well.

In Fig. 11(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 Fig. 11(b). We show the results in Fig. 11(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.

Fig. 11(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.

VI. 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 Fig. 12, 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.

A. Network Emulation

During the experiments, users experience several instances of the same scenario while subjected to different emulated network settings as described in Section V-B. To perform an extensive

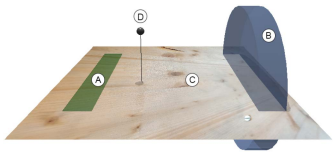

Fig. 12. A snapshot of *target tracking* game developed for the subjective performance evaluation of ETVO. 'A' is the moving target that needs to be tracked by the slider indicated with 'B'. 'C' is a plane that serves as a rigid floor. 'D' is the cursor that represents the position of the Novint Falcon 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.

TABLE I
CORRELATION BETWEEN USER GRADE AND USER OPINION

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

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 V-B.

B. 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 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 I.

C. 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.

D. Performance Analysis

The data from all participants is aggregated and presented in Fig. 13. 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 VI-A.

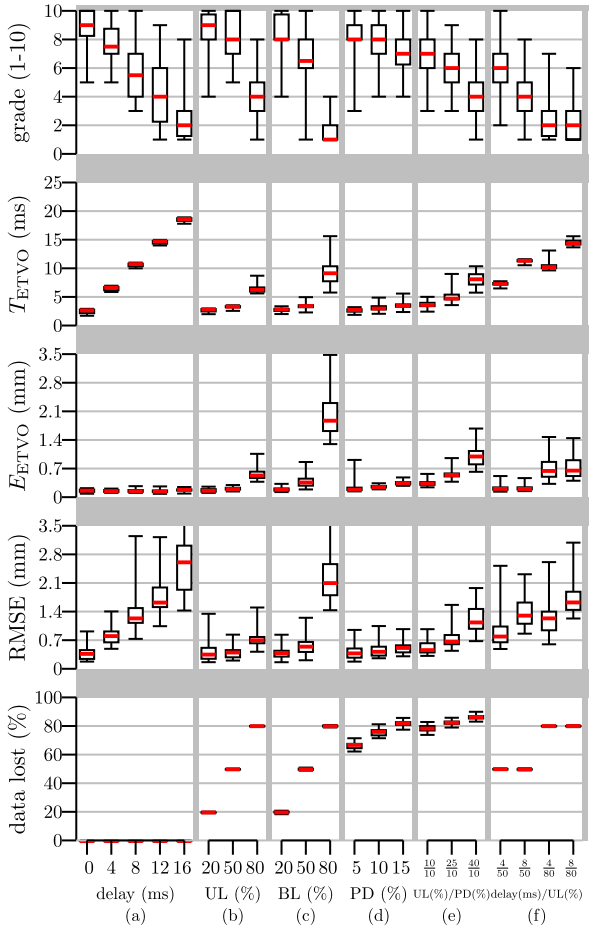

Fig. 13. Demonstration of ETVO's strong correlation with the user grades along with comparison against QoS and QoE metrics. The experiments are performed under diverse settings of (a) constant network delay, (b) uniform random packet loss, (c) bursty packet loss, (d) perceptual deadband (PD) scheme, (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.

1) ETVO versus QoS Methods: In this section, we take up each network setting (described in Section VI-A) separately and shed light on the important observations. Each column in Fig. 13 corresponds to a different network setting. To substantiate the performance of ETVO, we also present discussions relating to different network settings.

1. Network Delay. Fig. 13(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.

2. Uniform Loss (UL). In Fig. 13(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

$$\Delta t_{\text{update}} = \frac{1}{2f_s} + \frac{p}{f_s r(p+r)}, \tag{5}$$

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 Fig. 13(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 Fig. 13(a) with 80 % UL in Fig. 13(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 Fig. 13(c), we present the results for the BL scenario. The average update duration introduced previously and expressed as (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 (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 Fig. 13(c), the theoretical worst-case delay is significantly higher than what is projected by T_{ETVO}. The reason for this is twofold. First, 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}. Second, 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 Fig. 13(b), 20 % BL (Fig. 13(c)) and the 0 ms delay (Fig. 13(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 [31], [32], [33]. This highlights that the speculated ultra-reliability aspect of TI (99.9999 %) needs thorough investigations going forward.

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 Fig. 13(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 Fig. 13(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 (Fig. 13(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 (Fig. 13(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 %). Fig. 13(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.

2) ETVO versus QoE Methods: As a representative of this broad category of metrics, we use RMSE, since, as described in Section II-B, 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 Fig. 13(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 Fig. 13(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 Fig. 13(f), there are combinations of delay and packet loss. For RMSE two observations can be made. First, there is once again a high variance, that does not increase for higher packet loss. Second, 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., Fig. 13(b)–(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.

VII. Conclusion and Future Work

As the field of Tactile Internet (TI) is advancing fast, there is a strong need for quantifying its performance objectively. In this paper, we addressed the limitations of existing TI performance metrics. 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 TI applications. We developed an analytical framework – *Effective Time- and Value-Offset (ETVO)* – which addresses these issues and can be used to quantify TI performance. Through objective analysis, using realistic TI 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 which contradicts the anticipated ultra-reliability requirement.

While the current work looks at an offline session, we intend to design a real-time version of ETVO. Further, we would like to explore how different application-specific parameters can be adapted based on ETVO for maintaining the quality of TI sessions under time-varying network conditions. Furthermore, leveraging the existing video QoS/QoE metrics in tandem with ETVO for joint performance evaluation of haptic-video streams is another interesting research direction we would like to pursue in future.

Acknowledgments

This work has been undertaken in the Internet of Touch project sponsored by Cognizant Technology Solutions and Rijksdienst voor Ondernemend Nederland under PPS O&I.

References

[1] J. P. Verburg, H. Kroep, V. Gokhale, R. V. Prasad, and V. Rao, "Setting the yardstick: A quantitative metric for effectively measuring tactile internet," in *Proc. IEEE Conf. Comput. Commun.*, 2020, pp. 1937–1946.
[2] G. P. Fettweis, "The tactile internet: Applications and challenges," *IEEE Veh. Technol. Mag.*, vol. 9, no. 1, pp. 64–70, Mar. 2014.
[3] M. Maier, M. Chowdhury, B. P. Rimal, and D. P. Van, "The tactile internet: Vision, recent progress, and open challenges," *IEEE Commun. Mag.*, vol. 54, no. 5, pp. 138–145, May 2016.
[4] A. Aijaz and M. Sooriyabandara, "The tactile internet for industries: A review," *Proc. IEEE*, vol. 107, no. 2, pp. 414–435, Feb. 2019.

[5] V. Gokhale, M. Eid, K. Kroep, R. V. Prasad, and V. S. Rao, "Toward enabling high-five over WiFi: A tactile internet paradigm," *IEEE Commun. Mag.*, vol. 59, no. 12, pp. 90–96, Dec. 2021.
[6] R. Chaudhari, E. Steinbach, and S. Hirche, "Towards an objective quality evaluation framework for haptic data reduction," in *Proc. IEEE World Haptics Conf.*, 2011, pp. 539–544.
[7] N. Sakr, N. Georganas, and J. Zhao, "A perceptual quality metric for haptic signals," in *Proc. IEEE Int. Workshop Haptic Audio Vis. Environ. Games*, 2007, pp. 27–32.
[8] M. Müller, "Dynamic time warping," in *Information Retrieval for Music and Motion*. Berlin, Germany: Springer, 2007, pp. 69–84.
[9] M. Eid, J. Cha, and A. El Saddik, "Admux: An adaptive multiplexer for haptic–audio–visual data communication," *IEEE Trans. Instrum. Meas.*, vol. 60, no. 1, pp. 21–31, Jan. 2011.
[10] B. Cizmeci, X. Xu, R. Chaudhari, C. Bachhuber, N. Alt, and E. Steinbach, "A multiplexing scheme for multimodal teleoperation," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 13, no. 2, pp. 21:1–21:28, Apr. 2017.
[11] P. Hinterseer, E. Steinbach, and S. Chaudhuri, "Perception-based compression of haptic data streams using Kalman filters," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2006, pp. V–V.
[12] V. Gokhale, J. Nair, and S. Chaudhuri, "Congestion control for network-aware telehaptic communication," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 13, no. 2, pp. 17:1–17:26, Mar. 2017.
[13] X. Xu, S. Zhang, Q. Liu, and E. Steinbach, "QoE-driven delay-adaptive control scheme switching for time-delayed bilateral teleoperation with haptic data reduction," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2021, pp. 8838–8844.
[14] E. Muschter et al., "Perceptual quality assessment of compressed vibrotactile signals through comparative judgment," *IEEE Trans. Haptics*, vol. 14, no. 2, pp. 291–296, Second Quarter 2021.
[15] S. Kakade and S. Chaudhuri, "Perceptually compressive communication of interactive telehaptic signal," in *Proc. Int. Conf. Hum. Haptic Sens. Touch Enabled Comput. Appl.*, Springer, 2020, pp. 480–488.
[16] P. Hinterseer, S. Hirche, S. Chaudhuri, E. Steinbach, and M. Buss, "Perception-based data reduction and transmission of haptic data in telepresence and teleaction systems," *IEEE Trans. Signal Process.*, vol. 56, no. 2, pp. 588–597, Feb. 2008.
[17] R. Hassen and E. Steinbach, "HSSIM: An objective haptic quality assessment measure for force-feedback signals," in *Proc. IEEE Int. Conf. Qual. Multimedia Experience*, 2018, pp. 1–6.
[18] R. Hassen and E. Steinbach, "Subjective evaluation of the spectral temporal similarity (ST-SIM) measure for vibrotactile quality assessment," *IEEE Trans. Haptics*, vol. 13, no. 1, pp. 25–31, First Quarter 2020.
[19] L. Chen, M. T. Özsu, and V. Oria, "Robust and fast similarity search for moving object trajectories," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2005, pp. 491–502.
[20] L. Chen and R. Ng, "On the marriage of Lp-norms and edit distance," in *Proc. Int. Conf. Very Large Data Bases*, 2004, pp. 792–803.
[21] M. Vlachos, D. Gunopoulos, and G. Kollios, "Discovering similar multidimensional trajectories," in *Proc. 18th Int. Conf. Data Eng.*, 2002, pp. 673–684.
[22] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 26, no. 1, pp. 43–49, Feb. 1978.
[23] D. J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series," in *Proc. 3rd Int. Conf. Knowl. Discov. Data Mining*, 1994, pp. 359–370.
[24] S. Salvador and P. Chan, "Toward accurate dynamic time warping in linear time and space," *Intell. Data Anal.*, vol. 11, no. 5, pp. 561–580, 2007.
[25] D. F. Silva and G. E. Batista, "Speeding up all-pairwise dynamic time warping matrix calculation," in *Proc. SIAM Int. Conf. Data Mining*, SIAM, 2016, pp. 837–845.
[26] A. Bhardwaj et al., "A candidate hardware and software reference setup for kinesthetic codec standardization," in *Proc. IEEE Int. Symp. Haptic Audio Vis. Environ. Games*, 2017, pp. 1–6.
[27] E. Steinbach et al., "Haptic codecs for the tactile internet," *Proc. IEEE*, vol. 107, no. 2, pp. 447–470, Feb. 2019.
[28] T. N. Minhas, O. G. Lagunas, P. Arlos, and M. Fiedler, "Mobile video sensitivity to packet loss and packet delay variation in terms of QoE," in *Proc. 19th Int. Packet Video Workshop*, 2012, pp. 83–88.
[29] J. Nightingale, Q. Wang, C. Grecos, and S. Goma, "The impact of network impairment on quality of experience (QoE) in H.265/HEVC video streaming," *IEEE Trans. Consum. Electron.*, vol. 60, no. 2, pp. 242–250, May 2014.
[30] J. M. Silva, M. Orozco, J. Cha, A. E. Saddik, and E. M. Petriu, "Human perception of haptic-to-video and haptic-to-audio skew in multimedia applications," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 9, no. 2, pp. 1–16, 2013.
[31] S. Lee, S. Moon, and J. Kim, "A network-adaptive transport scheme for haptic-based collaborative virtual environments," in *Proc. ACM SIGCOMM Workshop Netw. Syst. Support Games*, 2006.
[32] J.-Y. Lee and S. Payandeh, "Forward error correction for reliable teleoperation systems based on haptic data digitization," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2013, pp. 5871–5877.
[33] M. Rank, Z. Shi, H. J. Müller, and S. Hirche, "Predictive communication quality control in haptic teleoperation with time delay and packet loss," *IEEE Trans. Human-Mach. Syst.*, vol. 46, no. 4, pp. 581–592, Aug. 2016.

Kees Kroep received the MSc degree in electrical engineering from the Delft University of Technology, in 2018. Currently, he is working toward the PhD degree with the Embedded and Networked Systems Group, EEMCS, TU Delft. His PhD work focuses on edge computing for tactile internet. His research interests include tactile internet and haptics technology.

Vineet Gokhale received the PhD degree from the Indian Institute of Technology Bombay, India, in 2017. During 2017–2019, he was an assistant professor at the University of South Bohemia, Czech Republic. He is currently a postdoctoral researcher with the Embedded and Networked Systems Group, Delft University of Technology, The Netherlands. His research interests include tactile internet, haptics technology, and MAC protocols for wireless communication. He is a contributing member of IEEE standardization committee for Tactile Internet P1918.1.

Joseph Verburg received the MSc degree in embedded systems from the Delft University of Technology, in 2019. He is currently working with KPN IoT as embedded full stack engineer working on IoT solutions using LoRa Wan, 5G, and LTE-m. His research interests include IoT, security, and tactile internet.

R. Venkatesha Prasad is an associate professor with ENSys Group, TUDelft. His research interest include the area of Tactile Internet, IoT, and 60 GHz mmWave networks. He has supervised 18 PhD students and more than 50 MSc students. He has close to 300 publications in the peer-reviewed international journals and conferences and standards, and book chapters. He has served on the editorial board of *IEEE Transactions on Mobile Computing*, *IEEE Communications Surveys and Tutorials*, *IEEE Transactions on Green Communications and Networking* and many other IEEE Transactions. He was the vice-chair of IEEE Tactile Internet standardization workgroup and now a mentor. For more information, please refer to http://homepage.tudelft.nl/w5p50