Assessing Quality of Control in Tactile Cyber–Physical Systems

Kurian Polachan[®], Joydeep Pal[®], Chandramani Singh[®], and T. V. Prabhakar[®], *Member, IEEE*

Abstract—We evolve a methodology and define a metric to evaluate Tactile Cyber-Physical Systems (TCPS). Towards this goal, we use the step response analysis, a well-known control-theoretic method. The adoption includes replacing the human operator (or master) with a controller with known characteristics and analyzing its response to slave side step disturbances. The resulting step response curves demonstrate that the Quality of Control (QoC) metric is sensitive to control loop instabilities and serves as a good indicator of potential factors that contribute to operatorside cybersickness. Through experiments, we demonstrate how QoC accounts for network overheads such as the link latency and jitter and non-networking overheads such as the testbed settings and robot performances in a TCPS. We show that there is a one-to-one correlation between QoC and end-to-end latency, jitter, and packet drops of a TCPS implementation. We show through experiments how QoC can be used to estimate positional errors in tactile-visual control applications. Since higher positional errors can result in poor task performance, estimating them is useful in developing a better-performing TCPS. We also evaluate a TCPS using Fitts' test and compare its results with QoC. We show that QoC is useful in distinguishing TCPS with differences in their specifications that are not detectable using Fitts' test.

Index Terms—Quality of control, tactile Internet, tactile cyberphysical systems.

I. INTRODUCTION

TELEOPERATION system consists of a human operator maneuvering a remote slave robot known as teleoperator over a network. The network transports the operator-side kinematic command signals to the teleoperator and provides the operator with audio, video, and/or haptic feedback from the remote-side. In existing teleoperation systems such as in telesurgery, due to significant end-end latency and reliability

Manuscript received 12 May 2021; revised 11 December 2021 and 19 February 2022; accepted 20 March 2022. Date of publication 1 April 2022; date of current version 31 January 2023. Visvesvaraya Ph.D. scheme and Center for Networked Intelligence (a Cisco CSR initiative) at IISc, Bengaluru for supporting this work. An earlier version of this paper was presented at the IEEE International Conference on Sensing, Communication and Networking (IEEE SECON), Boston, USA, 2019 [1] [DOI: 10.1109/SAHCN.2019.8824814]. The associate editor coordinating the review of this article and approving it for publication was L. Skorin-Kapov. (*Corresponding author: Kurian Polachan.*)

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by the IHEC committee, Indian Institute of Science.

Kurian Polachan is with the Department of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907 USA (e-mail: kpolacha@purdue.edu).

Joydeep Pal, Chandramani Singh, and T. V. Prabhakar are with the Department of Electronic Systems Engineering, Indian Institute of Science, Bengaluru 560012, India (e-mail: joydeeppal@iisc.ac.in; chandra@iisc.ac.in; tvprabs@iisc.ac.in).

Digital Object Identifier 10.1109/TNSM.2022.3164100

issues in both networking and non-networking components, the operator, i.e., the surgeon is forced to restrict his/her hand speed when he/she performs remote surgical procedures such as knot-typing, needle-passing, and/or suturing. This restriction in hand speed is required to avoid control loop instability and minimize operator side cybersickness [2], [3], [4]. Control loop instability occurs when the dynamics of the operator's hand actions exceed the response time of the teleoperation system. Instability in control loops can cause slave robots to go out of synchronization with the operator's hand movements, leading to disastrous consequences in critical applications. A possible cause of cybersickness is the presence of significantly noticeable delay in the operator receiving feedback in response to his/her actions. Cybersickness can result in general discomforts such as eye strain, headache, nausea, and fatigue and deter human operators from prolonged use of the teleoperation system [5].

More recently, researchers have envisioned combining tactile Internet, a network architecture characterized by submillisecond latency and very high packet reliability (99.999%). with advanced motion tracking hardware and high-speed robotics. This could help realizing teleoperation systems that do not impose restrictions on human hand speed [2], [3], [6]. We refer to these prospective teleoperation systems as Tactile Cyber-Physical Systems (TCPS) [7], [8], [9]. TCPS are envisioned to have applications in several domains where human skillset delivery is paramount, as in automotive, education, healthcare and VR/AR sectors. Realizing tactile Internet where end-to-end latency and reliability are within the prescribed limits is an important part. We think, however, that significant research concerning the design and evaluation is also an equally important part for building non-networking components such as hardware and its response, speed of algorithms and protocols that accompany a TCPS application.

In this paper, we formalize and investigate the evaluation part of TCPS research. In particular, we design an evaluation method catering to TCPS needs with the following objectives.

- Cybersickness and control loop quality are the two main concerns in TCPS. To ensure that they do not affect TCPS implementations, we must capture these effects during the early stages of TCPS prototyping. Further, to minimize the TCPS prototyping efforts, the evaluation method should also have the capability to assess the performance of individual TCPS components such as network connection, robot, or embedded boards.
- The experiments constituting TCPS evaluation should be reproducible and do not dependent on the judgments of human operators.

1932-4537 © 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information. • The experimental results should be captured by a single metric that has a physical significance. This is necessary to judge the effect of different uncorrelated variables on the planned TCPS application.

An evaluation method for TCPS needs catering to the above objectives is still non-existent, and therefore, there is an immediate need to develop one [10], [11].

A. Related Work

In the literature, we find many works that deal with the evaluation of TCPS (or TCPS like) systems. Depending on the method used for evaluation, we can classify these works into two broad categories.

1) Quality of Experience (QoE) Evaluation [12], [13], [14], [15], [16], [17], [18]: QoE evaluation studies a system by employing human subjects and evaluating how they experience the system while performing a task. It comes in different flavors — subjective, physiological and performance evaluations. In subjective evaluations, several human operators are invited to use the system for a specific duration of time. These operators provide feedback regarding their experience of using the system. Next, engineers convert this feedbacks to numerical scores, which are combined to determine a metric that is used as a vardstick to measure the quality of the system. A higher metric value indicates a better performing system. In physiological evaluation, the score is not decided based on the operator's answers. Instead, scores are decided based on the measures of the psychological parameters of the operators, such as the stress, muscle strain, or heartbeats, while they perform the task. In performance evaluation (e.g., Fitts' test), the system's performance is evaluated not by directly evaluating operators but by using scores that are commensurate of the performance of the application (e.g., the number of tasks completed in a given time or the accuracy of the task completion). All of the above discussed QoE based evaluation methods employ human operators performing a specific task, owing to which their results are highly subjective, consume a significant amount of time, application-specific, and many times lack repeatability.

2) Quality of Service (QoS) Evaluation [19], [20], [21]: In QoS evaluation, engineers use QoS metrics such as latency, jitter and packet drops to evaluate systems. In the context of TCPS, however, these metrics have many limitations. First, these metrics do not yield unique values for similar performing systems. There could be multiple combinations of the QoS metrics that would yield similar performing systems(e.g., specific high latency, low jitter system may perform similar to a different low latency, high jitter system). It is thus not easy to determine what combinations of these metrics values we need to consider to judge or compare the performance of different systems. Second, these metrics have limitations in determining a system's suitability for a specific application because these metrics do not account for applicationspecific inputs. For instance, even if QoS metric values are known, it is not particularly easy to determine if a TCPS implementation would reduce cybersickness in a specific application.



Fig. 1. Block representation of a generic control system.

Since human operators play an indispensable role in TCPS, QoE evaluation methods can provide critical insights into designing TCPS for better operability and user experience. However, in many cases, evaluations involving human operators (e.g., Fitts' test) may not identify TCPS performance issues that lie outside the sensory perception limits of human operators. For example, when operating a TCPS, human operators can miss detection of minor control-loop instabilities. Although such minor instabilities may be acceptable to some applications, this may not be the case with critical applications like telesurgery, where even minor levels of instabilities can cause injuries to patients. An evaluation method that eliminates the human operator is thus necessary. Though QoS assessment methods do not require human operators, they have many issues limiting their use in conducting an objective evaluation of TCPS, as discussed above. In our work, we design an evaluation framework that eliminates human operators' by replacing them with a controller. Further, to remove the dependency of evaluation on application-specific actions, we use the controller to perform a standard TCPS task, which is to correct step disturbances.

B. QoC in Literature: Definitions and Differences

We find that QoC is a term often used in real-time control literature. It refers to objective metrics used for measuring the performance of control systems. Several definitions of QoCs are found in literature [22], [23], [24], [25], [26], [27], [28]. Broadly, we classify these definitions into two groups depending on the performance measures authors use to determine QoC.

Definition-1: To determine QoC, authors measure the integral of error, *e*, after simulating a step disturbance in the control loop shown in Figure 1. Authors may use different approaches to measure integral of error like integral of absolute error (IAE), integral of square error or integral of weighted absolute error. For example, authors in [22], [23] measure IAE to determine QoC as follows.

$$IAE = \int_0^\infty |e(t)| dt \tag{1a}$$

$$QoC = \frac{1}{IAE}.$$
 (1b)

Definition-2: To determine QoC, authors measure the quadratic cost, J, instead of integral of error [24], [25], [26], [27], [28]. Equation(2) represent the generic form of J. u is the input to the plant, s is the plant state variable, and R and Q are weighing inputs. QoC evaluation using J has the advantage that it accounts for both the error and the energy consumed

by the controller.

$$J = \frac{1}{2} \int_0^\infty Ru(t)^2 + Qs(t)^2 dt.$$
 (2)

We list below the essential difference between our definition of QoC and QoC definitions in the literature.

- Although QoCs in the literature and QoC in our work both measure control performance, their objectives are different. In literature, authors use QoC to assess the quality of the controller for a given plant and a given load. However, we use QoC with the intent of assessing the quality of the plant, i.e., the TCPS.
- In the literature, authors, determine QoC by time-domain integration of signals (e.g., IAE, J). QoC's in the literature thus cannot comprehend the real dynamics of a control system [22]. For instance, it is possible that two different control systems, one that generates a spiky transient response with a low steady-state error and the other that generates a non-spiky transient response with a high steady-state error, can both yield the same IAE. In generic control systems, this might be acceptable but not for TCPS applications, where the presence of spikes can hamper the remote operation. For this reason, we use the rise time of the tuned step response curves as the performance measure to determine QoC. The method of tuning of step response curves ensures the rise time to account for both the transient and steady-state components of the error.
- The performance measures the authors use to determine QoCs in the literature devoids the resultant metric from having a physical significance, e.g., for a TCPS, they cannot comment on the recommended maximum operator hand speed. We determine QoC by measuring the rise time of the tuned good step response curves which enable us to derive the recommended maximum operator hand speed from QoC.

C. Our Contributions

Our contribution in this work is a method to perform an objective evaluation of TCPS and a metric (QoC) to quantify TCPS performance. We reason that a higher QoC indicates a better performing TCPS that is more tolerant to the effects of cybersickness and control-loop instabilities. We validate our claim through several experiments, including evaluation of a TCPS using Fitts' test. Below, we list our key contributions.

• *Evaluation Methodology:* We present an evaluation method for TCPS based on the step response method, a classic control-theoretic method for analyzing the quality of closed-loop systems. We leverage this method for TCPS by replacing the human operator with a controller with known characteristics and analyzing its response to slave side step changes. The conference version of this work [1] presented an evaluation method for accounting cybersickness in TCPS. In comparison, the evaluation methodology presented in this work accounts for both cybersickness and control loop instability issues in TCPS (see Section II and III.)

• Quality of Control: In the quest for an index to grade TCPS, we propose the metric *Quality of Control*(QoC). QoC is intended for non-real-time assessment of TCPS applications and is designed to capture the effect of different networking (e.g., link latency, jitter and packet drops) and non-networking (e.g., the latency associated with sensing, actuation, and algorithms) parameters on the TCPS application. Further, the relation we deduce between QoC and the maximum allowed hand speed of a human operator helps estimate positional errors in tactile-visual control applications. In comparison to the conference version of this work [1], the present work conducts several experimental evaluations of QoC. Specifically, we conduct QoC experiments for different end-to-end latency, jitter, and packet drops of a TCPS implementation. The present work also demonstrates the use of QoC in estimating the extent of positional errors in tactile-visual control applications (see Section VII-D, Section VII-C, and Section VII-E). Further, the present work also conducts evaluation of a TCPS using Fitts' test to validate QoC (see Section VIII).

D. Outline

We organize this paper as follows: In Section II, we propose an evaluation model for TCPS. From the different control loops in TCPS, we identify the critical control loops to avoid cybersickness and control loop instability. We propose that the quality of these critical control loops is an indicator of TCPS quality. Section III describes evaluation methodologies to assess the quality of the critical control loops. In Section IV, we propose the QoC metric and explain how to determine QoC from step response experiments. Section V describes the relation between QoC and the maximum allowed operator hand speed (to avoid cybersickness). In section VI, we introduce QoC performance curves and their use. In Section VII, we describe the evaluation of QoC. In Section VIII, we describe the evaluation of a TCPS that involves tactile-visual control using Fitts' test and compare the results with QoC. In Section IX, we discuss the potential limitations of QoC as an indicator of cybersickness. We conclude the paper in Section X.

II. EVALUATION MODEL

A typical TCPS can have multiple control loops, as shown in Figure 2. The control loops differ in their feedback modality. For instance, in the control loop *kinematic-video*, the feedback is video. Similarly, for the control loops *kinematic-audio* and *kinematic-haptic*, the feedbacks are audio and haptic respectively. In all these cases, the feedback is in response to the same kinematic commands from the human operator. Here, the presence of a human operator warrants that these control loops adhere to the stringent QoS specifications, in particular concerning their Round Trip Times (RTT). This is to minimize the operator-side cybersickness and control-loop instabilities.

This section first discusses the RTT requirements for different TCPS control loops. Next, the control loops that have the most stringent RTT requirement is determined. We call



Fig. 2. Functional blocks and the three control loops in a typical TCPS. The control loops *kinematic-audio loop (kal)*, *kinematic-video loop (kvl)* and *kinematic-haptic loop (khl)* are marked as 1, 2 and 3, respectively in the figure.

these *the critical control loops* and propose that the quality of these loop be used to benchmark the TCPS. We use Section II-A and Section II-B to separately study the two main issues in TCPS, cybersickness and control-loop instability. Cybersickness results when the round-trip time of the *kinematic-video* loop is high in TCPS designed for tactilevisual control applications. Control-loop instability results when the *kinematic-haptic* loop has a high round-trip time and the haptic feedback is delay-sensitive.

For this study, we make the following assumptions:

- In Section II-A, we consider TCPS where the haptic feedback is of type cutaneous¹ and does not require stringent RTT requirements (e.g., vibratory feedback). The assumption helps focus our attention on TCPS applications where the *kinematic-video* loop is critical.
- In Section II-B, we consider TCPS where the haptic feedback is of type kinesthetic and delay-sensitive (e.g., force-feedback). The assumption helps focus our attention on TCPS applications where the *kinematic-haptic* loop is critical.

A. Cybersickness

Cybersickness to humans occurs as a result of conflict between different sensory systems. The conflict arises when different sensory systems perceive the occurrence of the same event at noticeably distinct times. In TCPS, cybersickness may result from the asynchronous arrival of different feedback modalities at the operator side in response to the same kinematic commands.

We derive the maximum permissible feedback latencies (or maximum allowed RTTs) for different feedback modalities to avoid cybersickness. These latencies determine the RTT specification of the corresponding TCPS control loops. Note that in a TCPS, delays incurred by both networking and

 TABLE I

 MAXIMUM PERMISSIBLE SYNCHRONIZATION ERRORS OF DIFFERENT

 MEDIA STREAMS RELATIVE TO VIDEO [30], [31]

Media	Max Permissible Sync Error				
Streams	Relative to Video				
audio	+45 ms, -41 ms				
haptic (vibratory feedback)	$+125\mathrm{ms},-87\mathrm{ms}$				
TABLE II RTT Specification for TCPS Control Loops					
Feedback Modality	Control Loop Specification				

Feedback Modality	Control Loop Specification
video	$RTT_{kvl} \leq 1 \mathrm{ms}$
audio	$RTT_{kal} \le 46 \mathrm{ms}$
haptic (vibratory feedback)	$RTT_{khl} \le 126 \mathrm{ms}$

non-networking components contribute to RTT of its control loops.

We begin by observing in Table I the maximum permissible synchronization errors allowed for audio and haptic (vibratory feedback) streams relative to video.² Note that the maximum permissible synchronization errors can be either positive or negative depending on whether the audio and haptic streams arrive after or ahead of the video.

Table II shows the maximum permissible feedback latencies (or, maximum allowed RTTs of the corresponding control loops) derived from Table I. For these derivations, we assume that the display screen at the operator end shows the actual size of the remote field.

One of the foreseen application of TCPS is to realize tactilevisual control across a network [32]. In such applications, when there is a delay in video feedback, the operator will see a lack of synchronization between the movement of his/her hand and the remote robotic arm displayed on the screen. A typical human operator can move his/her hands at a maximum speed of 1 mm/ms [2]. Moreover, he/she can visually distinguish differences greater than or equal to 1 mm [2], [32]. Therefore, to avoid 1 mm or larger differences in positions of the operator's hand and the robotic arm displayed on the screen, the maximum permissible delay for the video feedback is required to be 1 ms [2], [32]. This restricts the RTT of *kinematic-video* loop, RTT_{kvl} , to be less than or equal to 1 ms (See Figure 3 (a) and (b)).³

When the operator moves his/her hand, he/she may expect to hear the sound from the remote environment (e.g., sound of the moving motor joints of the remote side robot or sound of the robot hitting a target) within a specific time limit. If the delay in audio feedback is more than this limit, the operator will find a lack of synchronization between the movement of his/her hand and the audio response. This will also result in

¹Haptic feedback is of two types, kinesthetic and cutaneous (or tactile). Kinesthetic feedback provides information about the stiffness of materials, while tactile feedback provides information concerning the texture and friction of material surfaces. We sense kinesthetic feedback through our muscles, joints and tendons while we sense the cutaneous feedback through the mechanoreceptors of our skin [29].

 $^{^2 \}rm We$ consider a TCPS use case where audio and video are streamed independently of each other, unlike the case with streaming methods like MPEG.

³Observe that if we restrict the human operator's maximum hand speed to v m/s, RTT_{kvl} can be allowed to be up to 1/v ms. Furthermore, if the display screen at the operator's end shows a zoomed out picture of the remote field (say a zoom factor $\alpha < 1$), then even with a maximum hand speed of 1 m/s and RTT of 1ms, the synchronization error seen by the operator will be less than $\alpha \text{ mm}$. In this case, RTT_{kvl} could be allowed to be larger.



Fig. 3. The error between the operator's hand and robotic arm displayed on the screen for different hand speeds. (a) The error is zero when RTT of the control loop, kvl, is zero. (b) The error is αmm when the RTT is αms . For a 1 m/s hand speed, the error in display translates to 1 mm.

cybersickness. We compute the maximum permissible latency for the audio feedback by adding the maximum permissible latency for the video feedback and the maximum permissible +ve synchronization error of audio relative to the video stream. We thus get the maximum permissible RTT_{kal} to be 46 ms.

Following similar reasoning, we find the maximum permissible latency of the haptic feedback to 126 ms. This restricts RTT_{khl} to 126 ms.

Note-1: Audio and haptic feedback in response to an operator action can reach the operator-side either ahead or after the video feedback. However, if RTT_{kvl} is maintained within 1 ms, then audio and haptic feedback will never reach ahead of video feedback by more than 1 ms. Since the maximum permissible negative synchronization errors between audio-video and haptic-video streams are much higher than 1 ms, audio and haptic streams arriving ahead of the video are not a cause of concern

Note-2: Restricting RTT_{kvl} to less than or equal to 1 ms in tactile-visual control applications requires that the displays for such applications should refresh at intervals of 1 ms or less. This requirement is multifold times stringent compared to the specifications of the vast majority of displays used in televisions and computers that refreshes at intervals of ≈ 17 ms (60 Hz). However, there do exist newer displays and lab-prototypes designed for fast-paced games and tactilevisual control applications that can refresh at ≈ 4 ms (240 Hz) and ≈ 1 ms (1 KHz) intervals, respectively [33], [34].

B. Control Loop Instability

As already described, the RTT requirement on the *kinematic-video* loop is much more stringent than the requirement on the *kinematic-haptic* loop to avoid cybersickness resulting from positional errors in the video display. Thus, towards avoiding cybersickness, the *kinematic-haptic* loop is not critical. However, in TCPS, where haptic feedback exists in the form of kinesthetic feedback (e.g., force-feedback), RTT of the *kinematic-haptic* loop, RTT_{khl} , becomes critical as follows.

In teleoperation systems with kinesthetic feedback, the operator experiences the haptic property (e.g., stiffness) of objects in the remote environment by commanding the teleoperator to tap the surface of these objects at different taping velocities and sensing the feedback forces through the operator side



Fig. 4. Presence of fast-acting local control loops in a TCPS with kinesthetic feedback [37].

manipulator [35]. For instance, to differentiate a hard material from a soft material, the operator taps the surface at a higher velocity. At a higher tapping velocity, the force feedback felt by the operator from a hard material will be higher in comparison to a soft material. In TCPS with kinesthetic feedback, to support for the maximum tapping velocity of a typical human operator and thereby enable the operator to differentiate materials of wide stiffness range, the RTT of the kinematic*haptic* loop, RTT_{khl} is required to be less than 1 ms [10]. RTT higher than 1 ms implies that the operator has to restrict his tapping velocity and thus limit his/her ability to differentiate materials of higher stiffness. However, if tapping velocity is not restricted, depending on the stiffness of the material, the kinematic-haptic loop may go unstable. This results in highfrequency oscillations in the operator side manipulator and remote side robot.

Unlike in *kinematic-audio* and *kinematic-video* loops, control loop instability can occur in the *kinematic-haptic* loop due to the existence of fast-acting local control loops at the operator and teleoperator sides. These local control loops close the global *kinematic-haptic* control loop detouring the human operator (see Figure 4). In regular operation, the local control loops are meant to synchronize both the position and force variables between the operator and the teleoperator sides. However, at higher RTTs, higher tapping velocities and in the presence of stiff materials, they assist in the generation of positive feedback destabilizing the global control loop [35], [36].

C. Control Loop for Evaluation

We propose to benchmark a TCPS by evaluating the quality of its critical control loop, the one with the most stringent RTT requirement. We have seen that both *kinematic-video* and *kinematic-haptic* loops are critical loops for avoiding cybersickness and control loop instability, respectively. We propose evaluating one of these loops, one that is critical for a given TCPS application. For instance, cybersickness is more prominent, i.e., the RTT_{kvl} requirement is more stringent, in applications that involve tactile-visual control [32]. Here, the term tactile refers to touch than tactile feedback [32]. On the other hand, control loop instability is more prominent, i.e., the RTT_{khl} requirement is more stringent, in applications involve tactile-visual control [32].

III. EVALUATION METHODOLOGY

We propose an evaluation methodology where we use the step response method to evaluate the critical control loops in a TCPS. In the following paragraph, we describe the step response method in the context of a generic control system.



Fig. 5. Evaluation framework in a haptic setting.



Fig. 6. Sample haptic step response curve. Haptic signal at time instants t_0 , t_1 , and t_2 are marked using red dots.

In the subsequent paragraph, we describe how we adopt the step response method for evaluating TCPS.

The step response method is a common approach to analyze the transient behavior and stability of control systems [38], [39]. In the step response experiment, a step disturbance is applied at the output-side (or load-side) of the control system, and then the waveform at the output terminal is recorded, all while the controller detects the disturbance through the feedback path and corrects the output (see Figure 5). The output waveform so captured is known as the step response profile of the control system (see Figure 6). The parameters of the step response profile, such as the rise time, overshoot, undershoot, and steady-state error, are dependent on the responsiveness of the controller and plant. For particular responsiveness settings of the controller, if the overshoots and undershoots are within the desired range, the control system is considered stable. Among all possible responsiveness settings of the controller, we define the step response profile that yields the lowest rise time as the optimal step response curve. The optimal step response curve is unique for a given controller and plant. Further, the rise time of the optimal step response curve is suggestive of the control system performance. A lower rise time implies that the control system is responsive and has an excellent transient performance.

For evaluating the critical control loops in a TCPS, we design control systems with TCPS acting as the plant and a known Proportional Integral (PI) controller substituting the human operator. We then tune the responsiveness of the PI controller to yield optimal step response curves. We note the corresponding rise time and use it to derive, QoC, a metric that is suggestive of the quality of the TCPS. In this section, we describe the design of these control systems to perform step response experiments. We describe how to determine the optimal step response curve and QoC in Section IV.

Algorithm 1 PI Controller Implementation
OperatorSide ():
initialize coordinates $[x = 0, y = 0]$
while $(x < X unit)$:
send robot coordinates x and y
wait for Δ seconds
if (haptic signal available):
receive and store in variable P
else:
store old haptic signal in P
find $error = P_{ref} - P$
compute $y = y + k_p \times error$
increment x by 1 unit

A. Evaluation Framework in a Haptic Setting

Evaluating the kinematic-haptic loop helps in quantifying control loop instability in a TCPS. For evaluating the kinematic-haptic loop, we propose the evaluation framework in Figure 5. To develop this, we replace the TCPS operator with a PI controller. At the teleoperator side of the testbed, we place a robotic arm with a haptic sensor mounted to its end effector. The haptic sensor detects pressure when the end effector of the robotic arm comes in contact with the material surface. In the step response experiment, we use the PI controller to move the robotic arm along the X-axis and to apply constant pressure of P_{ref} on the material surface along its Y-axis. When the robotic arm crosses the transition point separating the hard and the soft materials, the pressure abruptly drops from P_{ref} to P_{ref}/k_2 , simulating a step-change in the haptic domain. Here k_2 is a constant and is dependent on the characteristics of the materials in use.

The haptic sensor detects this change in pressure and communicates the new pressure to the operator side PI controller. The PI controller uses this data to compute a new y-coordinate of the robotic arm and communicates it back to the teleoperator side to take action. The intent here is to adjust y to increase the applied pressure and correct the haptic step change. The procedure repeats itself to generate a characteristic haptic step response curve.

1) Operator Side Implementation: We describe the implementation of the PI controller using Algorithm 1. Here in every loop, the x coordinate of the robotic arm is incremented by 1 unit. The increment can be decided based on the length of the material and the robot arm's reach. The parameter Δ is the loop wait time used for tuning the responsiveness of the controller and the system's stability. A controller using smaller Δ can potentially respond faster to the haptic signal, and therefore lead to smaller rise times in the step response curve. However, setting a very small Δ can potentially result in overshoots and oscillations in the step response curve. In Algorithm 1, k_p is the PI controller constant.

Note: The PI controller reuses the last received haptic signal to process the new robot coordinates if it does not receive any signal at the current time instance. For this reason, setting Δ smaller than Δ_{opt} can cause oscillations. We design the PI controller the above way intentionally. The design allows us to

Algorithm 2 Simulating Haptic Step-Change	
TeleOperatorSide ():	
while (true):	
wait to receive coordinates x and y	
if $(x < X/2)$:	
set haptic signal to $k_1 y$	
else:	
set haptic signal to $k_1 y/k_2$	
send haptic signal to operator side	
log haptic signal to file	

distinguish two TCPS of different performances by evaluating Δ_{opt} .

2) Tele-Operator Side Implementation: In the evaluation framework in Figure 5, we use a robot and a force sensor to simulate and measure the haptic step change. This implementation is useful if, in the TCPS evaluation, we desire to account for the robot's characteristics and the force sensor. Often we are not interested in accounting for the robot's or sensor's physical limitations (e.g., when we want to evaluate the TCPS communication network). In such cases, we simulate haptic step-change without a robot, as shown in Algorithm 2. In this algorithm, we have used k_1 as the conversion constant to translate y to pressure. The change in pressure from k_1y to k_1y/k_2 is simulated at x = X/2 unit — X being the stop value of x. We choose X = 100 in our experiments.

X decides the number of sample points step response curves contain from the onset of step-change. Stopping x at X = 100gives us X - X/2 = 50 sample points. Considering that it only takes three sample points to determine the step response parameters such as rise time, overshoots and settling errors of an ideal TCPS (see Figure 9 and its explanation), 50 sample points give enough buffer to work with a wide range of TCPS implementations. We also found 50 sample points to be adequate through experiments.

3) Step Response Curve: Figure 6 shows a sample haptic step response curve expected from the experiment. The curve resembles the classic step response curve in control system with its quality determined by certain rise time, overshoot, undershoot, settling time, and steady-state error. The quality of this curve directly indicates the quality of the *kinematic*-*haptic* loop, which in turn indicates the quality of the TCPS under test.

In our work, we define *rise time* as $t_r = t_2 - t_0 = t_d + t_2 - t_1$ where,

- 1) t_0 is the time at which the pressure drops to $P_{ref}/k_2 + 0.1(P_{ref} P_{ref}/k_2)$ (due to step change).
- 2) t_1 is the time at which the pressure rises back to $P_{ref}/k_2 + 0.1(P_{ref} P_{ref}/k_2)$.
- 3) t_2 is the time at which the step response reaches $P_{ref}/k_2 + 0.9(P_{ref} P_{ref}/k_2)$.

We define *overshoot* as the peak percentage fluctuation in the step response relative to $(P_{ref} - P_{ref}/k_2)$. Both the rise time and overshoot are affected by the characteristics of the TCPS components and the network.



Fig. 7. Evaluation framework in a non-haptic setting.

B. Evaluation Framework in a Non-Haptic Setting

Evaluating kinematic-video loop of a TCPS helps in determining V_{max} , the maximum operator hand speed the TCPS can support to avoid cybersickness contributed by positional errors in the video display. Since haptic sensors and haptic feedback are not part of the kinematic-video loop, the evaluation framework in Figure 5, which simulates step-change in the haptic domain is not useful. In Figure 7, we propose a modified evaluation framework. To develop this, we replace the TCPS operator using a PI controller with controller constant, k_p , and loop wait time parameter Δ . At the teleoperator side, we place a robotic arm with a video camera to detect the y-coordinate of the robot end-effector.

In the step response experiment, we use the PI controller to always maintain a constant y-coordinate, Y_{ref} , for the robotic arm. The PI controller in every control loop (i.e., after every Δ interval of time), increments, and send an epoch variable, n, to the teleoperator side. The PI controller initializes n to 1 at the start of the experiment. At the teleoperator side, n is used to set the value of k_2 . The teleoperator initializes k_2 to 1 at the start of the experiment and is set to a higher value when n is ≥ 50 . This simulates a step-change in the robot y-coordinate, y', at n = 50. The video camera at the teleoperator side detects the change in y-coordinate of the robot and communicates this new y-coordinate to the operator side PI controller. The PI controller uses this data to compute y, a new y-coordinate for the robotic arm and communicates it back to the teleoperator side to take action. The intent here is to adjust y to correct the step-change in y'. The parameters of the resultant step response curve, i.e., the plot of y', is similar in shape to Figure 6 and is used to evaluate the quality of the kinematic-video loop and thus the TCPS under test.

As for the evaluation framework in Figure 5, here also, we can replace the robot and the video camera with a code snippet if accounting the overhead of these components is not desired in evaluation. In our work, we discount the characteristics introduced by the display driver block. This is because display driver overheads are generally deterministic and known apriori, and their impact can be theoretically determined.

IV. QUALITY OF CONTROL

In this section, we first describe how to arrive at the optimum value of Δ , denoted as Δ_{opt} . We then describe how to determine the parameters P_{ref} , k_1 , k_2 and k_p of the evaluation



Fig. 8. Step response curves of a sample TCPS for a few values of Δ . Here $P_{ref} = 100$ and $P_{ref}/k_2 = 80$.

framework. We then describe QoC, the evaluation metric we propose for TCPS. For descriptions, we consider step response curves from the evaluation framework proposed for evaluating the *kinematic-haptic* loop. For this reason, subsequent figures showing step response curves have haptic signals in their y-axis. Note that the methods we describe here are also valid for the evaluation framework proposed for a non-haptic setting, i.e., for evaluating the *kinematic-video* loop. In the following, we write rise time as $t_r(\Delta)$ to show its dependence on Δ .

A. Determining Δ_{opt}

The step response curve extracted from the evaluation setup depends on the characteristics of the TCPS under test and also on Δ . For any given TCPS, different values of Δ lead to different step response curves. If Δ is very small, the resulting step response curve can have oscillations and overshoots. As we increase Δ , the controller's response gets slower, and hence oscillations and overshoots reduce, and the rise time increases. We define a good step response curve to be a response curve in which the overshoots and steady-state error are within prescribed limits - we set these limits to 20% and 10% respectively in our work.⁴ We want to achieve the fastest good response curve, i.e., the good response curve with the least rising time. Accordingly, we should set Δ to the least value that results in a good response curve; we refer to this value as Δ_{opt} . Alternatively stated, $\Delta = \Delta_{opt}$ results in a response curve that has the least rise time among all the response curves with overshoots and steady-state errors within 20% and 10% respectively. In the following, we refer to this good response curve as a Δ_{opt} -curve. Δ_{opt} and Δ_{opt} -curve are unique for a TCPS. They are dependent on the networking and non-networking parameters of the TCPS.

We determine Δ_{opt} through experiment. For the TCPS under test, Figure 8 shows the step response curves for a few values of Δ . Observe that, with 20% and 10% limits, the overshoots and steady-state error for good response curves should be less than 4 units and 2 units respectively. In particular, peaks of good response curves should be less than 104 units. $\Delta = 0.1$ ms and $\Delta = 0.6$ ms yield response curves with peaks exceeding 104 units. On the other hand, $\Delta = 0.7$ ms yields the good step response curve with the



Fig. 9. Haptic step response curve for different $k_p k_1$ settings for a TCPS with zero packet drops and $RTT_{\text{max}} < \Delta$. For $k_p k_1 = 1.25$, we see overshoot at the start of the step response experiment.

least rise time. We thus determine $\Delta_{opt} = 0.7$ ms. Note that a smaller Δ_{opt} implies a potentially smaller t_r and hence a better quality TCPS.

B. Design of Evaluation Parameters

For a TCPS with zero packet loss and $RTT < \Delta$, we can deduce a difference equation from the basic PI controller equation as follows. Let y_l be the current y-coordinate of the robot and y_{l+1} be its next y-coordinate determined by the PI controller. Then, for stability and fast convergence of this difference equation, the roots of its characteristic equation in the Z domain, $z - (1 - k_p k_1/k_2) = 0$, should be within the unit circle and close to the origin.

$$y_{l+1} = y_l + k_p \times error$$

= $y_l \left(1 - \frac{k_p k_1}{k_2} \right) + k_p P_{ref}$ (3)

We see from the characteristic equation that P_{ref} does not influence the stability or the convergence speed of the difference equation. Thus we can choose any value for P_{ref} . For ease of design we fix $P_{ref} = 100$ units. The parameter k_2 allows realizing the haptic step. We want to set k_2 to restrict the step signal to 20% of the P_{ref} . This is necessary to maintain the operation of the slave device around its operating point. We thus set $k_2 = 1.25$. Furthermore, we could set $k_p k_1 = k_2 = 1.25$ so that the root of the characteristic equation would be at the origin of the Z-plane. This would ensure that the configuration of the evaluation setup does not mask the characteristics of the TCPS under test, and we also achieve the fastest possible step response. However, this results in an overshoot at the start of the step response experiment (See Figure 9). This is because, initially, when x < 50 units the effective value of k_2 is 1 which results in the root of the characteristic equation to be in the left half of the Z-plane. In order to ensure that the root of the characteristic equation is positive all through, we set $k_p k_1 = 1$. In particular, we choose $k_1 = k_p = 1.$

We do not recommend setting $k_p k_1 < 1$. A lower $k_p k_1$ can increase the response time and can impact the ability of the evaluation methodology to detect packet drops or latency issues commonly observed with the TCPS communication networks. We demonstrate this in Figure 10. We simulate packet drops in the communication network of a TCPS under test. We see that packet drops cause an overshoot when $k_p k_1 = 1$ but do not when $k_p k_1 = 0.6$. Hence we

⁴In our work, we set limits only on overshoot and steady-state error to identify good step response curves. This is to simplify the classification of step response curves. In practice, we advise setting limits on all step response curve parameters including undershoot and settling time.



Fig. 10. Haptic step response curve (zoomed-in) for different $k_p k_1$ settings; with and without packet drops.

can detect packet drops from overshoot in the former case but cannot detect these in the latter case.

C. Design of Evaluation Metric

To evaluate TCPS, we propose a metric that is an indicator of TCPS control loop quality. We call this metric *Quality of Control (QoC)*.⁵ QoC of a TCPS is a relative measure of its quality compared to the quality of an ideal TCPS. We define an ideal TCPS as a TCPS with 1*ms* RTT, zero packet loss and zero jitter. Notice that Δ_{opt} for the ideal TCPS will be equal to its RTT, i.e., 1*ms*. Let $t_{r,ideal} \triangleq t_{r,ideal}(1ms)$ be the rise time of the corresponding Δ_{opt} curve. We then define QoC for the TCPS under test as,

$$QoC = \log_{10} \left(\frac{t_{r,ideal}}{t_{r}(\Delta_{opt})} \right) \tag{4}$$

where, $t_r(\Delta_{opt})$ is the rise time of the Δ_{opt} -curve of the TCPS under test. We use \log_{10} for the convenience of representing t_r ratio. We expect rise time ratio in (4) to assume a wide range as the latencies associated with TCPS components (e.g., network latency) can vary widely. This is why we use a logarithmic scale to specify QoC.

We now describe how we compute $t_{r,ideal}$. Observe that $t_{r,ideal}$ depends on k_p and k_1 through their product. We set $k_p k_1 = 1$ as suggested in Section IV-C. From Figure 9, we see that, for an ideal TCPS with $k_p k_1 = 1$, the PI controller takes three loop times ($\approx 3\Delta_{opt} = 3ms$) from x = 50 unit to x = 53 unit to correct the step change. Also $t_{r,ideal}$ is 1.5 loop times ≈ 1.5 ms.

Observe that QoC of the TCPS under test will be, (i) positive, if it has $t_r(\Delta_{opt}) < 1.5$ ms, which occurs when it performs better than an ideal TCPS. (ii) negative, if it has $t_r(\Delta_{opt}) > 1.5$ ms, which occurs when it performs poorly in comparison to an ideal TCPS. (iii) zero, if it has $t_r(\Delta_{opt}) =$ 1.5 ms, which occurs when it performs equivalent to an ideal TCPS. The metric intuitively indicates how fast (or slow) the operator can control the teleoperator using the haptic feedback without introducing significant control glitches at the teleoperator side.

⁵We use the term "metric" to mean a standard of measurement, as is often done in computer science and systems engineering. Examples include Signalto-Interference-plus-Noise Ratio (SINR) and Mean Opinion Score (MOS). This usage is different from the usage of "metric" in mathematics where it is a notion of distance in a metric space. More precisely, there it refers to a mathematical function that associates a real nonnegative number with each pair of elements in a metric space such that three axioms (a) identity of indiscernibles (b) symmetry and (c) triangle inequality are satisfied. *Note:* The RTT requirements that we put forward for the kinematic-video and kinematic-haptic loop are just the reference values used for computing QoC. In the event that these limits are not applicable for a specific use case, they can be adjusted to the desired application use case and rework the QoC calculation. Our evaluation framework and the method of deriving QoC from step response curves will remain agnostic of what reference RTT one chooses. In our work, we decided to select the reference RTT value as 1 ms for computing QoC as it is the most stringent RTT among all RTT values demanded by the envisioned TCPS applications such as force-feedback systems and tactile-visual control systems.

Note: Changing the reference RTT used for computing QoC only changes how the results are represented. It does not change the step-response experiment's objective nature, and outcome since $t_r(\Delta_{opt})$ does not depend on reference RTT. QoC thus remains objective irrespective of the reference RTT.

D. Methodology to Determine QoC

The above definition of QoC is on the presumption that each TCPS (for a given k_p , k_1) has an optimal loop wait time, Δ_{opt} . Also, Δ_{opt} , which is a function of TCPS RTT, packet drop rate, etc., can be estimated through a sequence of step response experiments. However, for any TCPS, its RTT, packet drops, etc., may vary with time. Hence, QoC, if measured as defined above, will vary with time. One can define QoC for a TCPS for the worst-case RTT and packet drops, but such a measure will present a pessimistic picture of the system and will be of little consequence in practice. In most of the applications, we want good responses for a prescribed fraction, say g_{spec} , of time. We call g_{spec} the desired goodness percentage. For instance, g_{spec} could be 0.99 for critical applications and smaller for others. Here, we give an alternate characterization of the quality of control as a function of the desired goodness percentage; we call it $QoC(g_{spec})$, We also describe a way to measure it in practice.

Recall that Δ_{opt} is the minimum value of Δ that yields a good response. We define $\overline{\Delta_{opt}}(g_{spec})$ to be the minimum Δ that yields a good response curve g_{spec} fraction of time. Note that $\overline{\Delta_{opt}}(g_{spec})$ will be large for a large g_{spec} . We can obtain $\overline{\Delta_{opt}}(g_{spec})$ as follows. We choose a small Δ and run the step response experiment a large number of times, say m times. We determine the fraction of times, say $g \approx (g > 0)$, good response curves are obtained. The parameter m is selected to limit the 95% confidence interval range of g to be within $\pm 5\%$. We now increase or decrease Δ depending on whether g is smaller or larger than g_{spec} and repeat the above experiments until $g = g_{spec}$. The final value of Δ is the desired $\overline{\Delta_{opt}}(g_{spec})$. We define $t_r(\overline{\Delta_{opt}}(g_{spec}))$ to be the average value of t_r over all the good response curves corresponding to $\overline{\Delta_{opt}}(g_{spec})$. Finally, we define,

$$\overline{QoC}(g_{spec}) = \log_{10} \left(\frac{t_{r,ideal}}{t_r(\overline{\Delta_{opt}}(g_{spec}))} \right).$$
(5)

Remark: $\overline{QoC}(g_{spec})$ averages out the time-varying characteristics of TCPS that influence QoC over a long period of time. Averaging ensures that the metric is reproducible.



Fig. 11. Step response plots of the operator's hand position, y, and haptic signal for an ideal TCPS with zero packet drop and $RTT_{max} = 1 ms$, and for a data sampling interval of 1 ms.

However, this also means that QoC measurements are not real-time. Real-time analysis of cyber-physical systems are required in scenarios where the system uses the result to tune its functionalities in real-time. Such as the network link to use and the algorithm settings to choose. However, in many CPS, such versatility is not inbuilt, and thus real-time evaluation is not required. For these systems, non-real time assessment such as QoC will be statistically more accurate than their real-time counterparts as they are done over a long period and can take into account the fluctuations in performances over time.

Remark: The passivity-based control method can guarantee system stability in the presence of delay. However, we do not use passivity-based methods for the following reason. Δ_{opt} , the minimum value of Δ that stabilizes a TCPS is used to derive QoC, a measure of the TCPS quality. We do not find a parameter (like Δ) in the passivity-based control methods that can directly or indirectly quantify the system's performance. Moreover, our step response experiments and QoC can also characterize TCPS employing passivity-based control. QoC, in these cases, can be used as a tool to test the performance of passivity-based control algorithms.

V. QOC VS. OPERATOR KINEMATICS

For a human operator, the maximum allowed hand speed, V_{max} , to avoid cybersickness in TCPS contributed by positional errors in the display is limited by two factors: the natural limit on human hand speed (1 m/s) and the quality of the TCPS kinematic-video loop. The latter is not the limiting factor in an ideal TCPS. More specifically, an ideal TCPS with RTT = 1 ms and $\overline{QoC}(1) = 0$ supports $V_{max} = 1 \text{ m/s.}^6$ A TCPS with lower $\overline{QoC}(g_{spec})$ (for the given g_{spec}) will support smaller hand speeds only, i.e., it will have $V_{max} < 1 \text{ m/s}$. In this section, we first derive the relation between $\overline{QoC}(g_{spec})$ and V_{max} . We then validate this relation through simulation.

A. Maximum Hand Speed

For an ideal TCPS with RTT = 1ms and no packet drop and jitter, the step response graph of robot y-coordinate, i.e., plot of y', and the graph of operator's hand position, i.e., plot of y, controlling the y-coordinate of the robotic arm, sampled at every 1ms, will have the same rise time t_r ; see Figure 11. For the plot, we choose $Y_{ref} = 100$ mm and use δy to represent



Fig. 12. Graph relating V_{max} and QoC.



Fig. 13. For a TCPS with $\overline{QoC}(1) = -0.3$, error between the operator's hand position, y, and robot position, y', exceeds 1 mm when V crosses $(V_{\text{max}} = 0.5 \text{ m/s})$.

the change in operator's hand position. This condition (the rise time being the same) also prevails in a non-ideal TCPS with RTT, packet drop and jitter of any value, provided we use $\Delta \geq \overline{\Delta_{opt}}(g_{spec})$ and we replace t_r with $t_r(\overline{\Delta_{opt}}(g_{spec}))$ and δy with $y(\overline{\Delta_{opt}}(g_{spec}))$.

We define the maximum allowed operator's hand speed V_{max} as dy/dt. For a TCPS,

$$V_{\max} = \frac{dy}{dt} = \frac{\delta y(\overline{\Delta_{opt}}(g_{spec}))}{t_r(\overline{\Delta_{opt}}(g_{spec}))}.$$
 (6)

Substituting $t_r(\Delta_{opt})$ from (4) in (6), we get

$$V_{\max} = \frac{\delta y(\overline{\Delta_{opt}}(g_{spec}))}{t_{r,ideal}} \times 10^{\overline{QoC}(g_{spec})}.$$
 (7)

Applying (7) to the ideal TCPS, i.e., substituting $V_{\max} = 1$ m/s and $\overline{QoC}(g_{spec}) = 0$, we obtain $\delta y(\overline{\Delta_{opt}}(g_{spec}))/t_{r,ideal} = 1$ m/s, a constant. Using this and also noting that V_{max} is limited to 1 m/s, the natural limit on human hand speed, we obtain the following simple relation between V_{max} and $\overline{QoC}(g_{spec})$. We illustrate this relation in Figure 12.

$$V_{\max} = \min\left\{1, 1 \times 10^{\overline{QoC}(g_{spec})}\right\} m/s.$$
(8)

B. Simulation Results

V

In any TCPS applications, the human operator is expected to restrict his hand speed to V_{max} a limit posed by the underlying TCPS. If the hand speed exceeds, the operator may experience cybersickness due to visible error (>1 mm) between his/her hand position and the robot position seen in the video feedback (see Figure 3). We demonstrate this in Figure 13. Here, y represents the operator's hand position and y' represents the robot position displayed on the operator side screen. We

⁶For an ideal TCPS with no randomness, for all g, $\overline{QoC}(g) = QoC = 0$.



Fig. 14. Sample QoC performance curves for three different TCPS.

consider a TCPS with $\overline{QoC}(1) = -0.3$ which corresponds to $V_{\text{max}} = 0.5$ m/s. As long as the operator maintains his hand speed, V, within 0.5 m/s, the error between y and y' is bounded within 1 mm. This error crosses 1 mm when V exceeds 0.5 m/s.

VI. QOC PERFORMANCE CURVE

When comparing performance of different TCPS using QoC, we use a common g_{spec} (i.e., $\overline{QoC}(g_{spec})$). Here g_{spec} is specified by the TCPS application. However, when vendors publish QoC for TCPS targeting multiple applications, listing QoC for one g_{spec} is not enough. As a solution, we propose publishing the QoC performance curve for TCPS. The curve illustrates how QoC varies with g_{spec} . In Figure 14, we plot sample QoC performance curves for three different TCPS. From the curves, we conclude the following; (i) for applications that demand $g_{spec} = 0.7$, TCPS-1 has better performance compared to TCPS-2 and TCPS-3 (ii) for applications that demand $g_{spec} = 0.9$, TCPS-2 has the better performance compared to TCPS-1. For this application, we cannot consider TCPS-3 as its QoC for $g_{spec} > 0.8$ is not specified.

Note that in Figure 14, QoC for $g_{spec} > 0.8$ is not specified for TCPS-3. This, however, is not a drawback of QoC or goodness percentage. The following are the two reasons why QoC for a TCPS may not be specified for a particular g_{spec} and beyond.

- First, lack of experimental results. If QoC is sharply decreasing beyond a g_{spec} , it is not worth testing the TCPS beyond that g_{spec} if one feels that the resultant QoC above g_{spec} is going to be too low for the TCPS to be considered for the intended application or compared with other TCPS. For TCPS-3 in Figure 14, the QoC is rapidly decreasing, and clearly, one can judge it performs poorly compared to TCPS-1 and TCPS-2 even without the availability of data beyond $g_{spec} = 0.8$.
- Second, QoC cannot be measured for a particular goodness percentage and beyond. The inability to specify QoC for a TCPS system for a given g_{spec} indicates that the system is unreliable at that g_{spec} due to higher packet loss percentage, and we should not consider it for applications that demand this g_{spec} and beyond. To give a parallel example, consider three network links of 1ms latency but different loss percentages 10%, 20% and 30%. In this setting, if one wants to measure the average latency experienced by 80% of the transmitted packets, for the first link, the measured value will be 1 ms, second-link will also yield 1 ms. However, we cannot specify the average



Fig. 15. Component level design of the testbed with the forward and backward flows marked [21].

latency value for the third link because it experiences a packet loss of 30%. Thus latency can be measured only for 70% (and not 80%) of the transmitted packets.

VII. PERFORMANCE EVALUATION

In this section, we first demonstrate how to use QoC to evaluate the overhead of a TCPS testbed. We then show how to use QoC to evaluate the quality of a TCPS network under different traffic conditions. Further, we demonstrate how QoC tracks end-to-end latency, jitter and packet drops of a TCPS implementation. Finally, we validate the use of QoC to estimate positional errors in a tactile-visual control application using real traces from a telesurgical dataset.

A. TCPS Testbed

Figure 15 shows the modular testbed we developed for TCPS and used for testing our proposed evaluation method.

- *operator* is the human operator or an embedded controller; *tele-operator* is the remote side slave device being controlled.
- ms embsys is the master side embedded system which houses sensors, actuators, and algorithms to capture the kinematic motions of the operator, to display audio and video, and to apply haptic feedback to the operator. ss embsys is the slave side embedded system that houses sensors, actuators, and algorithms for driving the slave side robot and for capturing the remote audio, video, and haptic signal.
- ms com is the master side communication component that connects ms embsys to the network; ss com is the slave side counterpart which connects ss embsys to the network.
- *srv* is the computer for offloading various TCPS algorithms in sensing, actuation, coding, and compression.
- *emu* is the optional emulator used to interconnect the testbed components. The emulator replaces a real physical network. We use ns-3 and Mininet to build this emulator [40], [41].

B. Evaluating the Testbed Overhead

We determine the overhead introduced by the testbed in Figure 15 in terms of QoC. We find the overhead of the testbed



Fig. 16. Setup for evaluating the testbed overhead. Arrows mark the forward kinematic and backward haptic signal flows.



(a) Plot of 1000 step curves for $\Delta=1.9\,{\rm ms.}$ Here 90.6% of step curves are good.



(b) Plot of 1000 step curves for $\Delta=1.8\,{\rm ms.}$ Here 80% of step curves are good.

Fig. 17. Plot of 1000 step curves for two different Δ . We find, g > 0.9 only when $\Delta \ge 1.9$ ms. Thus we determine $\overline{\Delta_{opt}}(0.9) = 1.9$ ms for a $g_{spec} = 0.9$.

framework without the network emulator to be $\overline{QoC}(0.9) = -0.35$. With network emulator, the overhead increases to $\overline{QoC}(0.9) = -0.92$.

1) Overhead of Testbed Framework: To evaluate the overhead introduced by the testbed framework, we realize the components ms com, srv and ss com in a single desktop PC (Processor:Intel-Core-i5, Core Count: 4, Processor Frequency: 3.4 Ghz, Memory: 3.7 GB) with <10% and <20% CPU and RAM utilizations, respectively (see Figure 16). To ensure the results are representative of the testbed overheads alone, (i) we retain minimal code in the testbed components to enable only the needed inter-component communication and (ii) we substitute the teleoperator side using a code snippet that also simulates the step input (to avoid accounting for teleoperator side component overheads). Following the steps (i) and (ii) results in the evaluation frameworks defined for a haptic setting and for a non-haptic setting to appear indistinguishable. i.e., irrespective of which evaluation framework we use, the end OoC will be the same.

We run the experiment to extract the step response curves for different values of Δ . For each Δ , we extract m = 1000step curves.

Figure 17 shows the results. We find $\overline{\Delta_{opt}}(0.9) = 1.9$ ms, $t_r(\overline{\Delta_{opt}}(0.9)) = 3.364$ ms (±0.21%) and $\overline{QoC}(0.9) =$



Fig. 18. Network topology simulated in Mininet.

-0.35. The metric indicates that the testbed implemented on the selected desktop PC and OS configuration is inadequate to realize ideal TCPS with 1 ms RTT. We either have to optimize the socket I/O calls or optimize background processes or replace the hardware. Note that, QoC < 0 does not mean that the TCPS implementation is not useful. It means the operator has to restrict his/her maximum hand speed to V_{max} during teleoperation. From (8), $V_{\text{max}} = 0.44$ m/s.

2) Overhead of Integrating Network Emulator: To understand the overall overhead of the testbed with the network emulator in place, we modify the setup in Figure 16. We evaluate QoC by placing the components *ms com* and *srv* in PC#1, *emu* running the ns-3 code in PC#2 and *ss com* in PC#3. The PCs are connected using point-to-point gigabit Ethernet links to interconnect the components, as shown in Figure 15. In ns-3, we emulate an ideal point-to-point link of zero latency, and zero packet drops to ensure the QoC measurement captures the testbed overhead alone and not the effects of any network components in ns-3.

From the experiment, We find QoC(0.9) = -0.92 indicating that incorporating the emulator block and running testbed components in different PCs increases the testbed overhead. From (8), this restricts V_{max} to 0.12 m/s.

C. Comparing TCPS Networks Using QoC

In this experiment, we demonstrate how to use QoC to compare networks with different traffic conditions. For evaluation, the network in Figure 18, which represents the northwest subset of the USNET-24 topology, is simulated using Mininet [41]. It consists of six switches managed by a central SDN controller. TCPS endpoints TE(Master) and TE(slave) connects to switches S1 and S6. Each switch in the network connects to a host, and every host runs the Linux tool iPerf to communicates with every other host bidirectionally to simulate external traffic. Spanning tree protocol is used to avoid looping issues. TE(Master) runs the PI controller, and TE(Slave) runs the algorithm to simulate step change. The latency and bandwidth of all links connecting hosts and switches are set to 0 ms and 10 Mbps, respectively. Latency and bandwidth of all links interconnecting the switches are set to 0.1 ms and 10 Mbps, respectively. The Mininet simulation is run on a server to reduce simulation overheads. In the experiment, we



Fig. 19. QoC curves for different traffic conditions.

measure QoC across the TCPS endpoints for different traffic rates simulated between host pairs.

Figure 19 shows the corresponding QoC performance curve for different simulated traffic rates. H-H Traffic in the legend corresponds to the unidirectional traffic rate simulated between host pairs. From the figure, we find QoC curves corresponding to H-H Traffic of 250 Kbps and 500 Kbps to be close. Because for both these cases, the net bitrate of the simulated traffic is not enough to throttle the bandwidth of the links connecting tactile endpoints. At rates above 500 Kbps, the net simulated traffic contributed by all the hosts in the network on the TCPS path gets close to the link bandwidth of 10 Mbps, which causes TCPS packets to experience higher latencies and resulting in lower QoC's. Further, we also note that the QoC performance curve has a higher slope at higher values of g_{spec} . Thus, aiming for higher values of g_{spec} is costly for maintaining a higher value of QoC.

Note that we restrict the bandwidth of the links interconnecting the switches in the Mininet topology to 10 Mbps. This is to minimize the effect of the server's performance that runs the Mininet simulation on the results. Note that unlike network simulators such as ns-3, Mininet emulates links and hosts in Linux kernel and thereby, its results are affected by the system performance, particularly when simulating links of very high bandwidth.

D. QoC vs. Latency, Jitter and Packet Drops

In this experiment, we demonstrate how QoC can track changes in end-to-end latency, jitter and packet drops of a TCPS implementation. In particular, we want to demonstrate that QoC measures deteriorate when the performance of a TCPS deteriorates due to higher end-to-end latency, jitter and packet drops. The experimental setup consists of the TCPS endpoints TE(Master) and TE(Slave) simulated in Mininet and interconnected over a 10 Mbps link whose latency, jitter and packet drops can be varied. TE(Master) and TE(Slave) communicates each other with packets of size 100 B. We conduct three experiments. In the first experiment, we measure QoC for different link latency with zero jitter and packet drops. In the second experiment, we measure QoC for different jitter values for a link latency of 5 ms and zero packet drops. In the third experiment, we measure the supported maximum g_{spec} , $\max(g_{spec})$, of the QoC for different packet drop percentages, 5 ms link latency and zero jitter. Figure 20 show the results of the first experiment in the clustered column chart format. The chart shows how QoC and V_{max} vary with link latencies.



Fig. 20. Experimental results showing how QoC(0.9) and V_{max} varies with link latencies.

TABLE III
LEFT-TABLE: EXPERIMENTAL RESULTS SHOWING HOW QOC(0.9)
VARIES WITH JITTER. RIGHT-TABLE: EXPERIMENTAL RESULTS
Showing how Supported Maximum g_{spec} , $\max(g_{spec})$, of QoC
VARIES WITH DIFFERENT LINK PACKET DROP PERCENTAGES

QoC vs. Jitter(mdev)		Packet Drop vs. $max(g_{spec})$			
Jitter (ms)	QoC(0.9)	Packet Drop (%)	$ \max(g_{spec}) $		
0	-1.00	0	1.00		
0.41	-1.07	5	0.89		
0.84	-1.14	10	0.77		
2.07	-1.29	15	0.67		

At lower link-latencies, both QoC and V_{max} are higher. QoC and V_{max} drops with increase in link latency Table III show the results of the second and third experiment.

We find that when latency or jitter of the link increases, the measured QoC drops and when packet drops increase, the supported maximum g_{spec} , max (g_{spec}) , which is an indicator of TCPS reliability drops. We find that when latency or jitter of the link increases, the measured QoC drops and when packet drops increase, the supported maximum g_{spec} , max (g_{spec}) , which is an indicator of TCPS reliability drops.

E. QoC and Positional Error

In this experiment, we first measure the combined QoC of a communication network and a connected robot in a non-haptic setting. We then validate the use of QoC to estimate positional errors in displays in tactile-visual control applications.

1) QoC Measurements: For evaluation, the network in Figure 18 is simulated using Mininet [41]. TE(Master) runs the PI controller, and TE(Slave) simulates step change. The step change is simulated using the robot, IRB 4600, as described in Section III-B. We use the Virtual Robot Experimentation Platform (VREP) to simulate the robot [42]. We run VREP in TE(Slave) in headless mode configuration, i.e., without GUI, to minimize the simulation overhead. A custom written python script interfaces the network socket of TE(Slave) with VREP. We configured the delay and bandwidth of the links interconnecting the switches to 5 ms and 10 Mbps, respectively. H-H Traffic is disabled.

We measure QoC with and without the robot in place. Without the robot, we find $\overline{QoC}(1) = -1.42$, and with the robot, we find $\overline{QoC}(1) = -1.7$. From (8), this restricts V_{max} to 0.02 m/s.



Fig. 21. Histogram of operator's hand velocity for $F_s = 30$ Hz. For 82% of the time velocity is less than $V_{max} = 0.02$ m/s.

TABLE IV EXPECTED AND MEASURED VALUES OF ${\it E}$ for Different Values of ${\it F}_s$

$\overline{F_s}$ (Hz)	Expected E (%)	Measured E (%)
40	77	81
30	82	86
20	88	92

2) QoC Validation: If QoC of the TCPS and dynamics of the operator's hand movements are known a priori, we can predict for what percentage of time positional error's in displays will be within the operator's visual tolerance limit of 1 mm using the QoC- V_{max} relation of (8) as follows. For example, consider a TCPS with $\overline{QoC}(1) = -0.1$ (i.e., $V_{max} = 0.1$ m/s) and historical operator's hand velocity dataset has velocities less than 0.1 m/s for 90% of the time. Here, following the definition of V_{max} , one can conclude that the error between the operator's hand position and robot position in the display will be less than 1 mm for 90% of the time.

For validating the above claim, we use the TCPS in Section VII-E1, whose QoC and V_{max} is known (QoC(1) =-1.7 and $V_{max} = 0.02$ m/s). Further, we define E as the percentage time for which the error between the position of the operator's hand and the robot is within 1 mm during a TCPS operation.. In the experimental setup, the operator side replays data corresponding to the hand movements of a surgeon performing the suturing operation using the da Vinci surgical system captured originally at a sampling frequency of 30 Hz [43]. At the teleoperator side, the robot (in VREP) replicates the surgeon's hand movement. Figure 21 shows the histogram of the operator's hand movement for a sampling frequency of $F_s = 30$ Hz. We find that for 82% of the time, the velocity is less than V_{max} of 0.02 m/s. We thus expect E to be 82%. In Table IV, we list the expected and the measured values of E for different F_s (we up-sample and down-sample the dataset in [43] to generate $F_s = 40$ Hz and $F_s = 20$ Hz datasets, respectively). We find the measured results to be very close to expectation and also to follow the expected E vs. F_s trend, i.e., lower E at higher F_s .

In the experiments, to measure E, at every $1/F_s$ seconds, the robot position is fed back to the operator side. At the operator side, at every instance of receiving the feedback, the error is calculated. For this, we subtract the robot position from the estimated position of the operator's hand at the receiving instance. From the error values, we find E by determining the percentage of time the error values stay within 1 mm.

VIII. QOC VALIDATION USING FITTS' TEST

We find that several works in the literature use Fitts' test [44] for evaluating TCPS like applications involving human operators performing aimed movements and target acquisitions [45], [46], [47], [48], [49], [50], [51], [52], [53], [54]. In particular, Fitts' test has been used to evaluate quality of tactile-visual control applications [45], [46] and performance of teleoperation systems [47], [48], [49], [50], [51], [52], [53], [54]. Since one of the foreseen applications of TCPS is tactile-visual control over a network [2], [32] and since TCPS is primarily a teleoperation system, we decided to use Fitts' test to evaluate TCPS and validate QoC.

This section discusses the evaluation of a TCPS application involving tactile-visual control using Fitts' test. We run the tests for different values of network latencies, jitter, and packet drops and compare the results with QoC.

A. Basics of Fitts' Test

Fitts' test consists of rapid aimed movements where a human operator selects targets of a certain size over a certain distance. An example is the alternate selection of two lines of specific width, *W*, and distance of separation, *A*. Fitts models the difficulty of this task using the task difficulty index, *ID*, as follows.

$$ID = \log_2\left(\frac{A}{W} + 1\right)$$

The equation for ID is inspired by the theorem proposed by Shannon for determining the capacity of a communication channel. The unit of ID is bits because the ratio within the parenthesis is unitless and the log is taken to base 2. Further, Fitts proposed throughput, TP, which is measured from a sequence of trials consisting of different W, A and subjects. TP is calculated as a simple quotient with ID of the task in the numerator and the mean movement time MT in the denominator as follows.

$$TP = \frac{ID}{MT}.$$

TP has a unit of bits per second (bps). Fitts via experiments show that TP is independent of A and W though they are embedded in the equation of ID. When ID changes due to changes in A or W), MT changes in an opposing manner and TP remains more or less the same.

TP however is dependent on the platform, e.g., the display in use, the joystick used for selection etc. Thus performing a Fitts' test on two platforms can result in different *TP*s. The platform that results in the highest *TP* is considered to have better usability. In other words, the user experience of a platform, in our case TCPS, is proportional to *TP*.

B. Experimental Setup

Figure 22 describes the experimental setup. It consists of two hosts one at the operator-side and one at the teleoperator-side separated by a network link. The host at the



Fig. 22. Experimental setup for conducting Fitts' test of a TCPS application that involves tactile-visual control.

TABLE V Different QoS Settings Configured for the Network Link in Figure 22

No.	Session	Condition	Latency (ms)	Jitter (ms)	Packet Drop (%)
1		C0	0	0	0
2	S0	C1	10	0	0
3		C2	15	0	0
4		C3	20	0	0
5		C1	20	6	0
6	\$1	C2	20	12	0
7		C3	20	18	0
8		C1	20	0	5
9	\$2	C2	20	0	10
10		C3	20	0	15

teleoperator-side runs Fitts' law software in [55]. Operator controls the selection of the lines in the Fitts' law software using a computer mouse connected to the operator-side host and using video feedback of the teleoperator-side screen. The QoS of the network link is adjusted to vary latency, jitter and packet drops to simulate different network conditions.

We conduct Fitts' test with ten participants; five males and five females of different age groups (24-45). Each participant is given five minutes to familiarize with the experimental setup before conducting actual evaluation. For each participant, we conduct the Fitts' test for ten different link QoS settings. Table V lists these settings. We divide the settings into three sessions. Each session consists of multiple test conditions, each defining a specific QoS setting. In Session S0, we study the impact of latency variations on system performance by varying latency of the link. In session S1, we study impact of jitter on the system performance. In session S2, we study the effect of packet drops on system performance. Participants has to perform fifteen trials of target selections to complete a test condition. Each test condition is repeated for two different A settings of 400 pixels (10 cm) and 100 pixels (2.5 cm). For the experiments W is fixed to 20 pixels (0.5 cm).

Figure 23 shows a participant performing the Fitts' test using our experimental setup. The participant uses a headrest placed 60 cm away from the video display. During the experiment, participants select targets as quickly and accurately as possible at a comfortable pace. Fitts' law software



Fig. 23. Operator-side of the experimental setup in Figure 22.

TABLE VI AVERAGE VALUES OF ID_e , A_e , W_e , MT and TP Measured From Fitts' Test

		A_e (pixels)	$ W_e$ (pixels)	IDe (bits)	MT (ms)	TP (bps)
 S0	C0	250.65	14.36	3.96	1096.44	3.68
	C1	250.17	13.71	4.01	1185.51	3.47
	C2	251.22	14.72	3.93	1196.31	3.34
	C3	249.83	15.71	3.81	1399.04	2.75
	C1	250.82	14.88	3.89	1169.36	3.34
^{SI}	C2	249.68	14.12	3.98	1255.59	3.20
	C3	250.81	15.77	3.83	1279.80	3.04
S2	C1	251.14	15.96	3.80	1115.26	3.42
	C2	250.22	16.27	3.77	1353.22	2.80
	C3	250.97	15.71	3.83	1720.22	2.31

detects missing targets during the test. Participants are asked to repeat the trials in a test condition if they miss the targets more than 5% of the time.

We also conduct QoC evaluation for each test condition in Table V. For this, the operator-side host runs the PI controller code and the teleoperator-side host simulates step change.

C. Results and Discussions

Table VI summarizes the Fitts' test results. Here, TP is computed using the adjusted values of ID (ID_e), A (A_e) and W (W_e) for improved accuracy [44]. Figure 24 plots the variation of measured TP and QoC across latency, jitter and packet drops.

Discussions: Both TP and QoC tend to decrease when QoS parameters such as latency, jitter or packet drops are higher. However, for specific combinations and ranges of QoS settings, TP is not sensitive to variations in QoS. For instance, TP remains almost constant when jitter varies from 0 ms to 6 ms or when packet drop varies from 0% to 5%. The reason for this behavior can be attributed to the following. The variations in the QoS parameters at certain regions may have been masked by experimental setting, e.g., use of specific A and W values in Fitts' test or the hardware in use (e.g., the use of 60 Hz video display). Variations in these regions may also be unnoticeable to the operator due to human sensory limitations. QoC on the other hand is capable of faithfully tracking changes in



Fig. 24. TP and QoC across latency, jitter and packet-drops.

QoS parameters for all ranges and combinations experimented with. Note that even if the operator's experience is not affected by minor variations in QoS, they can still be a cause of concern in TCPS. For some settings, minor variations in QoS can result in minor control-loop instabilities. Although such minor levels of instabilities may be acceptable with some applications, that may not be the case with critical applications like telesurgery where even minor levels of instabilities can cause injuries to patients.

Further, If we look at the QoC and *TP* variation with latency, QoC exhibits a logarithmic behavior. QoC shoots up when the latency is small. In other words, the differences are amplified when the latency is very small. This characteristic is useful in differentiating the quality of different TCPS (and its components) having minor differences in their QoS settings. Note that TCPS operate at very low latencies.

IX. POTENTIAL LIMITATION OF QOC AS AN INDICATOR OF CYBERSICKNESS

Specific to TCPS implementation, several works in the literature have experimentally shown how higher latency, jitter, and packet drops in the presence of high operator dynamics can result in poor task performance and cybersickness [45], [46], [56], [57]. Our proposed metric, QoC, can help track end-to-end latency, jitter and packet drop in a TCPS implementation and positional errors resulting from fast operator dynamics (e.g., fast hand speeds). We show in Section VII-D that there is a one-to-one correlation between QoC and end-to-end latency, jitter, and packet drops of a TCPS implementation. QoC deteriorates when latency, jitter and packet drops are high and improves when they are low. We also show in Section VII-E using real-traces from a telesurgical dataset how QoC tracks positional errors in tactile-visual control applications. For these reasons, we argue that QoC is a good indicator of cybersickness in TCPS implementations.

However, In TCPS, operator-side cybersickness depends on several factors, not merely on the TCPS implementation alone. Cybersickness also depends on the type of TCPS task and its duration, the complexity of the video scenes, the operator's experience, the type of human-computer interface, and even the lighting condition in the room. Cybersickness thus has many contributing factors, out of which QoC tracks only factors specific to TCPS implementation, which we consider to be the main limitation of QoC.

X. CONCLUSION

In this paper, we present an evaluation method and metric for Tactile Cyber-Physical Systems. Our evaluation method is based on step response analysis, a classic control-theoretic method to evaluate closed-loop systems. Our metric, QoC, is derived from the parameters of the resultant step response curves and is demonstrated to detect increases in TCPS's endto-end latency, jitter and packet drops, which are potential factors for poor task performance and cybersickness. We compare QoC of a TCPS with results from Fitts' test, a popular evaluation method for TCPS like systems. Although QoC and output of Fitts' test exhibit similar trends for different network settings, we notice that, human visual receptions and subsequent reactions, which determine the outcome of Fitts' test, fail to notice the impact of small changes in network parameter values in certain ranges. These changes though minor may be significant enough for critical TCPS applications, e.g., telesurgery. We see that QoC can capture these variations, affirming its usefulness for TCPS evaluation.

ACKNOWLEDGMENT

The authors thank Visvesvaraya Ph.D. scheme and Center for Networked Intelligence (a Cisco CSR initiative) at IISc, Bengaluru for supporting this work.

REFERENCES

- K. Polachan, T. V. Prabhakar, C. Singh, and D. Panchapakesan, "Quality of control assessment for tactile cyber-physical systems," in *Proc. 16th Annu. IEEE Int. Conf. Sens. Commun. Netw. (SECON)*, Jun. 2019, pp. 1–9.
- [2] G. P. Fettweis, "The tactile Internet: Applications and challenges," *IEEE Veh. Technol. Mag.*, vol. 9, no. 1, pp. 64–70, Mar. 2014.
- [3] "The Tactile Internet." ITU. 2014. [Online]. Available: https://www.itu.int/en/ITU-T/techwatch/Pages/tactile-internet.aspx
- [4] N. Arjun, S. M. Ashwin, K. Polachan, T. V. Prabhakar, and C. Singh, "An end to end tactile cyber physical system design," in *Proc. 4th Int. Workshop Emerg. Ideas Trends Eng. Cyber-Phys. Syst. (EITEC)*, Apr. 2018, pp. 9–16.
- [5] J. J. LaViola, "A discussion of cybersickness in virtual environments," ACM SIGCHI Bull., vol. 32, no. 1, pp. 47–56, Jan. 2000. [Online]. Available: http://doi.acm.org/10.1145/333329.333344
- [6] A. Aijaz, Z. Dawy, N. Pappas, M. Simsek, S. Oteafy, and O. Holland, "Toward a tactile Internet reference architecture: Vision and progress of the IEEE P1918.1 standard," 2018, arXiv:1807.11915v1.

- [7] K. Polachan, J. Pal, C. Singh, T. V. Prabhakar, and F. A. Kuipers, "TCPSbed: A modular testbed for tactile Internet-based cyber-physical systems," *IEEE/ACM Trans. Netw.*, early access, Nov. 15, 2021, doi: 10.1109/TNET.2021.3124767.
- [8] K. Polachan, C. Singh, and T. V. Prabhakar, "Decentralized dynamic scheduling of TCPS flows and a simulator for time-sensitive networking," ACM Trans. Internet Technol., to be published. [Online]. Available: https://doi.org/10.1145/3498729
- [9] K. Polachan, "Tactile cyber-physical systems: A testbed, a performance metric and network resource allocation protocols," Ph.D. dissertation, Dept. Electron. Syst. Eng., Indian Inst. Sci., Bangalore, India, 2021.
- [10] K. Antonakoglou, X. Xu, E. Steinbach, T. Mahmoodi, and M. Dohler, "Towards haptic communications over the 5G tactile Internet," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 3034–3059, 4th Quart., 2018.
- [11] D. V. D. Berg *et al.*, "Challenges in haptic communications over the tactile Internet," *IEEE Access*, vol. 5, pp. 23502–23518, 2017.
- [12] A. Hamam and A. E. Saddik, "Evaluating the quality of experience of haptic-based applications through mathematical modeling," in *Proc. IEEE Int. Workshop Haptic Audio Visual Environ. Games (HAVE)*, Oct. 2012, pp. 56–61.
- [13] Y. Kusunose, Y. Ishibashi, N. Fukushima, and S. Sugawara, "QoE assessment in networked air hockey game with haptic media," in *Proc. 9th Annu. Workshop Netw. Syst. Support Games*, Nov. 2010, pp. 1–2.
- [14] A. Tatematsu, Y. Ishibashi, N. Fukushima, and S. Sugawara, "QoE assessment in haptic media, sound and video transmission: Influences of network latency," in *Proc. IEEE Int. Workshop Tech. Committee Commun. Qual. Rel. (CQR)*, Jun. 2010, pp. 1–6.
- [15] M. A. Jaafreh, A. Hamam, and A. E. Saddik, "A framework to analyze fatigue for haptic-based tactile Internet applications," in *Proc. IEEE Int. Symp. Haptic Audio Visual Environ. Games (HAVE)*, Oct. 2017, pp. 1–6.
- [16] N. Sakr, N. D. Georganas, and J. Zhao, "A perceptual quality metric for haptic signals," in *Proc. IEEE Int. Workshop Haptic Audio Visual Environ. Games*, Oct. 2007, pp. 27–32.
- [17] C. G. Corrêa, D. M. Tokunaga, E. Ranzini, F. L. S. Nunes, and R. Tori, "Haptic interaction objective evaluation in needle insertion task simulation," in *Proc. 31st Annu. ACM Symp. Appl. Comput.*, 2016, pp. 149–154.
- [18] R. Chaudhari, E. Steinbach, and S. Hirche, "Towards an objective quality evaluation framework for haptic data reduction," in *Proc. IEEE World Haptics Conf.*, Jun. 2011, pp. 539–544.
- [19] 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.
- [20] V. Gokhale, J. Nair, and S. Chaudhuri, "Congestion control for networkaware telehaptic communication," ACM Trans. Multimedia Comput. Commun. Appl., vol. 13, no. 2, p. 17, Mar. 2017. [Online]. Available: https://doi.org/10.1145/3052821
- [21] K. Polachan, T. V. Prabhakar, C. Singh, and F. A. Kuipers, "Towards an open testbed for tactile cyber physical systems," in *Proc. 11th Int. Conf. Commun. Syst. Netw. (COMSNETS)*, 2019, pp. 375–382.
- [22] G. Buttazzo, M. Velasco, and P. Mart, "Quality-of-control management in overloaded real-time systems," *IEEE Trans. Comput.*, vol. 56, no. 2, pp. 253–266, Feb. 2007.
- [23] G. Buttazzo, M. Velasco, P. Marti, and G. Fohler, "Managing qualityof-control performance under overload conditions," in *Proc. 16th Euromicro Conf. Real-Time Syst.*, Jul. 2004, pp. 53–60.
- [24] Y. C. Tian and L. Gui, "QoC elastic scheduling for real-time control systems," *Real-Time Syst.*, vol. 47, no. 6, pp. 534–561, 2011.
- [25] E. Bini and A. Cervin, "Delay-aware period assignment in control systems," in *Proc. Real-Time Syst. Symp.*, 2008, pp. 291–300.
- [26] A. Aminifar, Analysis, Design, and Optimization of Embedded Control Systems, vol. 1746. Linköping Univ. Electron. Press, 2016.
- [27] A. Aminifar, P. Eles, Z. Peng, A. Cervin, and K. E. Arzen, "Controlquality-driven design of embedded control systems with stability guarantees," *IEEE Design Test*, vol. 35, no. 4, pp. 38–46, Aug. 2018.
- [28] A. Cervin, J. Eker, B. Bernhardsson, and K. E. Årzén, "Feedback-feedforward scheduling of control tasks," *Real-Time Syst.*, vol. 23, nos. 1–2, pp. 25–53, 2002.
- [29] A. Aijaz, M. Dohler, A. H. Aghvami, V. Friderikos, and M. Frodigh, "Realizing the tactile Internet: Haptic communications over next generation 5G cellular networks," *IEEE Wireless Commun.*, vol. 24, no. 2, pp. 82–89, Apr. 2017.
- [30] D. J. Levitin, K. MacLean, M. Mathews, L. Chu, and E. Jensen, "The perception of cross-modal simultaneity (or 'the greenwich observatory problem' revisited)," in *Proc. AIP Conf.*, vol. 517, 2000, pp. 323–329.

- [31] 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, p. 9, May 2013.
- [32] G. P. Fettweis, "A 5G wireless communications vision," *Microw. J.*, vol. 55, no. 12, pp. 24–39, 2012.
- [33] NVIDIA. "Nvidia-Gsync." 2020. [Online]. Available: https://www.nvidia.com/en-in/geforce/products/g-sync-monitors/
- [34] A. Ng, M. Annett, P. Dietz, A. Gupta, and W. F. Bischof, "In the blink of an eye: Investigating latency perception during stylus interaction," in *Proc. SIGCHI Conf. Human Factors Comput. Syst.*, 2014, pp. 1103– 1112. [Online]. Available: https://doi.org/10.1145/2556288.2557037
- [35] E. Steinbach et al., "Haptic codecs for the tactile Internet," Proc. IEEE, vol. 107, no. 2, pp. 447–470, Feb. 2019.
- [36] E. Steinbach *et al.*, "Haptic communications," *Proc. IEEE*, vol. 100, no. 4, pp. 937–956, Apr. 2012.
- [37] K. Antonakoglou, X. Xu, E. Steinbach, T. Mahmoodi, and M. Dohler, "Toward haptic communications over the 5G tactile Internet," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 3034–3059, 4th Quart., 2018, doi: 10.1109/COMST.2018.2851452.
- [38] "Control System Basics." Ledin Engineering. 2019. [Online]. Available: https://ledin.com/control-systems-basics/
- [39] J. Ledin, Embedded Control Systems in C/C++. San Francisco, CA, USA: CRC Press, Inc., 2003.
- [40] "NS-3." 2018. [Online]. Available: https://www.nsnam.org
- [41] "Mininet." 2019. [Online]. Available: http://mininet.org/
- [42] "V-Rep Virtual Robot Experimental Platform." Coppeliarobotics. 2018. [Online]. Available: http://www.coppeliarobotics.com/
- [43] Y. Gao et al., "JHU ISI gesture and skill assessment working set (JIGSAWS) a surgical activity dataset for human motion modeling," in Proc. Model. Monitor. Comput. Assist. Interventions (M2CAI) MICCAI Workshop, 2014, pp. 1–10.
- [44] I. S. MacKenzie, "The Wiley handbook of human computer interaction," in *Chapter Fitts' Law*. Chichester, U.K.: Wiley, 2018, ch. 17, pp. 347–370.
- [45] H. V. Le, V. Schwind, P. Göttlich, and N. Henze, "PredicTouch: A system to reduce touchscreen latency using neural networks and inertial measurement units," in *Proc. ACM Int. Conf. Interactive Surfaces Spaces*, 2017, pp. 230–239.
- [46] R. Jota, A. Ng, P. Dietz, and D. Wigdor, "How fast is fast enough? A study of the effects of latency in direct-touch pointing tasks," in *Proc. SIGCHI Conf. Human Factors Comput. Syst.*, 2013, pp. 2291–2300. [Online]. Available: https://doi.org/10.1145/2470654.2481317
- [47] R. Dubey, S. Everett, N. Pernalete, and K. Manocha, "Teleoperation assistance through variable velocity mapping," *IEEE Trans. Robot. Autom.*, vol. 17, no. 5, pp. 761–766, Oct. 2001.
- [48] P. Malysz and S. Sirouspour, "Task performance evaluation of asymmetric semiautonomous teleoperation of mobile twin-arm robotic manipulators," *IEEE Trans. Haptics*, vol. 6, no. 4, pp. 484–495, Oct.–Dec. 2013.
- [49] K. Hayashi and T. Tamura, "Teleoperation performance using excavator with tactile feedback," in *Proc. Int. Conf. Mechatronics Autom.*, Aug. 2009, pp. 2759–2764.
- [50] R. L. Williams, J. M. Henry, and D. W. Repperger, "Evaluation of rate-based, force-reflecting teleoperation in free motion and contact," *Presence Teleoper. Virtual Environ.*, vol. 9, no. 1, pp. 25–36, Feb. 2000. [Online]. Available: https://doi.org/10.1162/105474600566592
- [51] T. Bonaci, J. Yan, J. Herron, T. Kohno, and H. J. Chizeck, "Experimental analysis of denial-of-service attacks on teleoperated robotic systems," in *Proc. ACM/IEEE 6th Int. Conf. Cyber-Phys. Syst.*, Apr. 2015, pp. 11–20. [Online]. Available: https://doi.org/10.1145/2735960.2735980
- [52] S. Chen, T. Rahman, and W. Harwin, "Performance statistics of a head-operated force-reflecting rehabilitation robot system," *IEEE Trans. Rehabil. Eng.*, vol. 6, no. 4, pp. 406–414, Dec. 1998.
- [53] K. Corker, J. H. Lyman, and S. Sheredos, "A preliminary evaluation of remote medical manipulators," *Bull. Prosthet. Res.* vol. 16, no. 2, pp. 107–134, 1979.
- [54] C.-S. Chung, H. Wang, and R. A. Cooper, "Functional assessment and performance evaluation for assistive robotic manipulators: Literature review," J. Spinal Cord Med., vol. 36, no. 4, pp. 273–289, Jul. 2013.
- [55] S. MacKenzie. "Fitts' Law Software." 2021. [Online]. Available: http://www.yorku.ca/mack/FittsLawSoftware/
- [56] Z. Shi, H. Zou, M. Rank, L. Chen, S. Hirche, and H. J. Muller, "Effects of packet loss and latency on the temporal discrimination of visual-haptic events," *IEEE Trans. Haptics*, vol. 3, no. 1, pp. 28–36, Jan.–Mar. 2010.
- [57] J. Stauffert, F. Niebling, and M. E. Latoschik, "Effects of latency jitter on simulator sickness in a search task," in *Proc. IEEE Conf. Virtual Real. 3D User Interfaces (VR)*, 2018, pp. 121–127.



Kurian Polachan received the B.Tech. degree in electronics and communication engineering from the Cochin University of Science and Technology in 2008, and the M.Tech. degree in electronics design and technology and the Ph.D. degree working on tactile Internet systems and time-sensitive networking from the Indian Institute of Science in 2014 and 2021, respectively. He is currently a Postdoctoral Researcher with Purdue University, working on ultra-low power wearables and human-body communications for biomedical applications. He has over

five years of industry research experience with Cypress Semiconductors (currently Infineon Technologies), developing capacitive touch sensing userinterface solutions for electronic products. His research interests include wearable designs, human body communications, time-sensitive networking, tactile Internet systems, and capacitive touch sensing technologies.



Chandramani Singh received the M.E. and Ph.D. degrees in electrical communication engineering from the Indian Institute of Science, Bengaluru, India, in 2005 and 2012, respectively. He is currently an Associate Professor with the Department of ESE, Indian Institute of Science. From 2005 to 2006, he was with ESQUBE Communication Solutions Pvt. Ltd., Bengaluru. From 2012 to 2013, he was a Research Engineer with TREC, a joint research team between INRIA Rocquencourt and ENS de Paris, Paris, France, and from 2013 to 2014, a Postdoctoral

Research Associate with CSL, University of Illinois at Urbana–Champaign, Champaign, IL, USA. His interests include communication networks, data centers, and smart grids.



Joydeep Pal received the B.Tech. degree in electronics and communication engineering from Delhi Technological University (formerly DCE) in 2016. He is currently pursuing the Ph.D. degree from the Department of Electronic Systems Engineering, Indian Institute of Science. He has worked on IoT systems for home automation. His research focuses on tactile Internet systems.



T. V. Prabhakar (Member, IEEE) received the M.Sc. (Engg.) degree from the Indian Institute of Science (IISc), Bengaluru, and the Ph.D. degree from TU Delft, The Netherlands. He is currently a Principal Research Scientist with the Department of Electronic Systems Engineering, IISc, where he is currently involved in networked embedded systems research. The Zero Energy Networks Laboratory with IISc, specializes in building ultralow power embedded hardware and software network stacks. The areas of research include RFID, Device ID

PUFs, energy harvesting and power management algorithms, tactile CPS infrastructure, digital twin, IoT data management, speech decoding on embedded systems, and condition monitoring.