Robotica (2018) volume 36, pp. 167–186. © Cambridge University Press 2016
doi:10.1017/S0263574716000679

Development and analysis of an operator steering model for teleoperated mobile robots under constant and variable latencies

Steve Vozar†*, Justin Storms‡ and D. M. Tilbury‡

†*University of Michigan, Department of Computer Science and Engineering, Ann Arbor, MI 48109, USA*
‡*University of Michigan, Mechanical Engineering Department, Ann Arbor, MI 48109, USA*
E-mails: jgstorms@umich.edu, tilbury@umich.edu, svozar@umich.edu

(Accepted August 30, 2016. First published online: October 5, 2016)

SUMMARY
Latency hinders a mobile robot teleoperator's ability to perform remote tasks. However, this effect is not well modeled. This paper develops a model for teleoperator steering behavior as a PD controller based on projected lateral displacement, which was tuned to reflect user performance determined by a 31-subject user study under constant and variable latency (having mean latencies between 0 and 750 ms). Additionally, we determined that operator performance under variable latency could be mapped to the expected performance of an equivalent constant latency. We then tested additional latency distributions in simulation and demonstrated equivalent steering performance among several different latency distributions.

KEYWORDS: Telerobotics; User models; Latency; Human–robot interaction; Wheeled robots.

1. Introduction

Latency is a significant factor affecting teleoperated robot performance. Whether the latency originates in the system communications network, processing routines, or sensing hardware, it can negatively impact a human operator's ability to perform even basic remotely-operated tasks, such as navigation and object manipulation. Moreover, while operators can sometimes adapt to system delay if it remains relatively consistent, variable latency makes it difficult for humans to predict how the robot will respond.[1,2] While many teleoperated industrial or surgical robots have the benefit of communicating over wired networks, mobile robots generally must utilize wireless protocols, which typically have more latency and latency variation. The effects of variable latency are known to be detrimental; however, they are not well modeled in telerobotic systems.

Modeling human performance in teleoperation systems has multiple potential benefits. Performance models can predict the relationship between overall system performance and factors that affect teleoperator ability, such as the presence of latency or the use of haptic feedback. Behavioral models can be used in place of a real human operator and can enable evaluation of a large number of design iterations and broader test conditions than would otherwise be feasible and cost-effective.[3] Models can be used for early testing in the design process, before using human subjects and developing full prototypes.[3] Model development can also reveal underlying patterns related to human–robot interaction that can be used to inform future design decisions.

Previous studies on human performance modeling of teleoperated robots have primarily developed performance models. For example, Yip *et al.*[4] investigated how the interaction between communication latency and haptic feedback affects performance. The model developed in the study showed that in the presence of high communication latency, haptic feedback can decrease the

* Corresponding author. E-mail: svozar@umich.edu

magnitude of the operator-commanded force the robot exerts on the environment, but the addition of haptic feedback increased the task completion time. Yang *et al.*[5] found that using analog (rather than digital) controllers can significantly improve system performance. In both cases, these studies demonstrated useful relationships; however, the performance models cannot be used to replace a human operator and simulate additional test conditions.

The automotive industry has a long history of developing and testing driver models that can be used to replace a human operator in system simulations.[6] However, these models may not be directly applicable to teleoperation, as the two tasks are quite different. Although vehicle drivers have a wide variety of sensory feedback available, teleoperators generally rely solely on visual feedback, which is often delayed and has a limited field of view.[7] Additionally, input devices for automobiles (e.g., steering wheels with haptic feedback) may not be the same as those used in teleoperation, which can be as simple as off-the-shelf video game controllers, tablet computers, or mice and keyboards. Finally, the internal mental models automobile drivers have for their vehicles are likely far more developed than those of even experienced teleoperators.

Thus, there is a need to develop models that describe the performance of human operators steering teleoperated wheeled robots in the presence of latency. Such models could be useful as a substitute for a real teleoperator when testing mobile robot designs for tasks requiring steering inputs. For example, a designer wishing to optimize a robot's suspension could use simulated inputs of realistic user steering behavior in conjunction with dynamic physical models of the chassis to evaluate design options. A user model could also be used as a first-pass test to quickly tune parameter values for a new steering-assist behavior before experimenting with human subjects.

In this paper, we develop such a model, based on a 31-subject user study designed to measure the effects of both constant and variable latency on a simulated teleoperation steering task using a commercially available gamepad input device. The input commands from the human to the robot were recorded and used to develop a driver model that simulates human behavior for simple steering tasks under different latency conditions. The two key contributions of this work are as follows: (1) We show that a teleoperator's steering commands with a gamepad can be modeled as a PD controller based on the preview of the robot's anticipated lateral displacement. To the best of our knowledge, this is the first control theoretic steering model that captures the changes in teleoperated mobile robot performance with constant and variable latencies. (2) We show that under the conditions tested in this study, there is a relationship between the characteristics of a variable latency distribution and operator performance. This relationship indicates that a variable latency distribution can be considered equivalent to a constant latency (independent of speed) when comparing steering performance. The equivalent constant latency is greater than the mean delay of the variable distribution. This relationship enables the prediction of steering model gains for a range of latency distributions without requiring human-in-the-loop testing for each latency type.

The remainder of the paper is organized as follows: First, prior work regarding the effect of latency on teleoperation as well as human operator models is discussed in Section 2. Then, Section 3 presents a user study designed to characterize human teleoperators' responses to different latency scenarios. A driver model is developed, tuned, and validated using the test data in Section 4. An equivalence between operator performance under constant latency and multiple variable latency distributions is found and discussed in Section 5. Finally, Section 6 discusses conclusions and future work regarding the direction of this research.

2. Background

2.1. Latency in teleoperation

It is well established that latency has a detrimental impact on teleoperation performance, and time delay is known to be one of the most significant factors affecting remote perception and manipulation.[7] Sources of latency in a teleoperated robot system include network delays, sensing delays, and processing delays.[8]

One of the earliest studies in this domain investigated open-loop position control of a remote manipulator, and found that users adopted a move-and-wait strategy when the delay was above 1.0 s.[9] Since Sheridan's early work,[9] many researchers have focused on methods of reducing the impact of communication delay on teleoperation performance. Strategies include using predictors,[10]

https://doi.org/10.1017/S0263574716000679 Published online by Cambridge University Press

using augmented reality,[11] adapting control gains,[12] automating subtasks,[13] and using different input modalities (e.g., hands-free operation using gestures[14] or voice[15]).

Many of these methods have been shown to be effective at improving teleoperation performance. However, it may be difficult for designers of robot teleoperation systems to decide when it is appropriate to include such methods. That is, when does the improvement in teleoperation performance justify the added cost to include such teleoperation assistance features? To answer that question, an understanding of the relationship between performance and latency is required.

Sheridan's early studies on time-delayed space telerobotics provided a theoretical basis for predicting performance as measured by task completion time under constant delay, assuming operators performed tasks as a series of discrete open-loop movements.[9] Since then, significant research has been done to describe the relationship between constant latency and mobile robot teleoperation performance under conditions ranging from 2D driving[1,2] to 3D underwater navigation tasks.[16] The directionality of the latency (whether user-to-robot or robot-to-user) has also been investigated, where it has been found that users felt robot control was more difficult when the latency was in the robot-to-user direction, but no objective difference in performance was observed.[1]

Much of the work investigating the impact of communication latency on wheeled mobile robot teleoperation performance has focused on designing stable haptic control devices,[17] incorporating techniques such as asymptotic tracking of position and force.[18] Studies have investigated how communication latency combined with human operator training[19] or additional sensor feedback or assistance[20] impacts mobile robot teleoperation performance. However, these studies only consider constant communication delay.

Real-world communication latency is often time-varying. For example, Ford demonstrated a "remote repositioning" system capable of cross-country vehicle teleoperation. The cellular networks used for communications had variable latencies and bandwidth restrictions.[21] Research that has investigated the impact of time-varying latency on performance metrics other than stability (e.g., time to complete a task, number of collisions with obstacles) has only compared time-varying latency with constant latencies.[1,2] To the best of our knowledge, no prior work has suggested how features of a time-varying latency distribution (e.g., mean and variance) could be quantitatively related to other time-varying or constant latencies in terms of teleoperation performance metrics beyond stability.

2.2. *Human operator models*

Human operator models have been used to model performance in applications ranging from aircraft control to human–computer interface design. In this brief literature overview of human operator models, we will discuss three categories of models stemming from: (1) Fitts' law, (2) control theoretic approaches, and (3) cognitive architectures.

2.2.1. Fitts' law models. Fitts' law is a human operator model that was originally developed to predict the time required to point to a target area, modeled as a function of the ratio between the distance to the target and the width of the target.[22] Fitts' law can be generalized to predict the time required to point to a target as a function of a difficulty index. Accot and Zhai expanded Fitts' law to show that the time T to steer through a path is governed by[23]

$$T = a + b \cdot ID \tag{1}$$

where a and b are constants and ID is a difficulty index. While this steering law was originally developed for 2D trajectory-based interactions, such as menu navigation with a mouse,[23] it has been demonstrated that this relationship also holds for locomotive steering tasks in virtual environments.[24] The steering law does not account for constant or variable communication latencies.[24]

Although this law has not been directly tested in the presence of latency, similar work has found that performance in 2D target-following tasks using a computer mouse is significantly degraded by latencies over 110 ms and for latency variations over 40 ms.[25]

Kaber *et al.*[26] found that the task completion time for a virtual surgical target-acquisition task in the presence of latency (between 0 and 150 ms) could be predicted by a modified Fitts' law using an additional parameter to account for latency. A similar result was found by MacKenzie for planar mouse-pointing tasks.[27] This indicates that the steering law may also require corrective terms to accommodate latency.

https://doi.org/10.1017/S0263574716000679 Published online by Cambridge University Press

2.2.2. Control theoretic models. Control theoretic human operator models allow for analysis of overall teleoperation system performance from a controls perspective, which can result in improved performance and proven stability properties for such systems. A control system formulation of the problem can leverage the larger body of work that has analyzed time-delay systems in domains such as network control systems[28] or using transforms such as the delta operator.[29]

Control theoretic based human operator models have roots in modeling airplane pilots,[30] automobile drivers,[31] and manipulator arm operators.[32] Slawiñski and Mut[10] have developed a human operator model for use in simulation of teleoperated mobile robots. However, the focus of their model is for control design and stability analysis, and does not capture how human operator behavior changes with different levels of latency.

Most relevant to the present study is the modeling of human driver behavior in the automotive domain.[6,31,33] From transfer function models to nonlinear and adaptive controllers,[6,33] there are myriad methodologies for modeling vehicle lateral control (steering), longitudinal control (acceleration and braking), and combined control. These models can be used to simulate human drivers when testing new vehicle designs and technologies,[6,33] and despite the complexity of human behavior, low order models are often sufficient for many control tasks.[34]

In our prior work, we introduced[8] and performed a stability analysis[35] on a low-order model inspired by MacAdam[31] that can be used to simulate human steering behavior in a teleoperated navigation task in the presence of latency. In this paper, we discuss the development of this model in further detail. We note that a model based solely on constant latency is limited in applicability, so we developed a method by which such a model could be applied to multiple distributions of variable latency.

2.2.3. Cognitive architecture models. Cognitive architectures have been used to model higher-level robot operator behavior. For example, Ritter *et al.*[36] developed a cognitive model of a user for a robotic navigation and pick-and-place task. The goal of their model is to aid in HRI design and predict human performance during teleoperation tasks. This model was developed in the ACT-R cognitive architecture.[37] While cognitive architecture models can capture many human behavior characteristics and features, they are more computationally expensive to simulate than many of the control theoretic models discussed. Cognitive architecture models often must be run in real-time, so they are not suitable for optimizations requiring thousands of simulated iterations or in situations where processing power is limited.

3. User Study

A user study was performed to gather data on task performance and operator driving behavior in robot steering tasks using a simulation of a teleoperated mobile robot driving on a test track of constant width. The user's goal was to steer the robot such that it followed the track's indicated centerline as closely as possible. The results of the study were used to tune the gains of the model described in Section 4.

3.1. Task setup

3.1.1. Simulation environment. A custom simulation environment for this set of tests (with a look inspired by the classic arcade game *Pole Position*) was written in Java using the April Robotics Toolkit[38] and Lightweight Communications and Marshalling (LCM)[39] libraries. A block diagram of the teleoperation system, including the human operator, is shown in Fig. 1. The simulation uses a kinematic driving model of a representative skid-steer robot chassis (see Fig. 2). The realism of a physics-based simulation is not as crucial for this teleoperation task as it might be in an automotive driving task since teleoperators do not experience any of the dynamic physical feedback cues (such as lateral acceleration) that affect the robot while in motion.

3.1.2. User interface. The user provides steering commands to the robot using an analog mini joystick on a standard computer gamepad (Logitech Cordless Rumblepad 2). A gamepad was chosen for use in the study rather than a steering wheel or other input devices such as multi-degree-of-freedom haptic devices because of its widespread usage in mobile robot teleoperation systems.[40,41] A small amount of noise was artificially added to the gamepad input, generated from a uniform distribution on the range of $[-10\%, 10\%]$ of the maximum possible input command, to simulate the noise present

https://doi.org/10.1017/S0263574716000679 Published online by Cambridge University Press

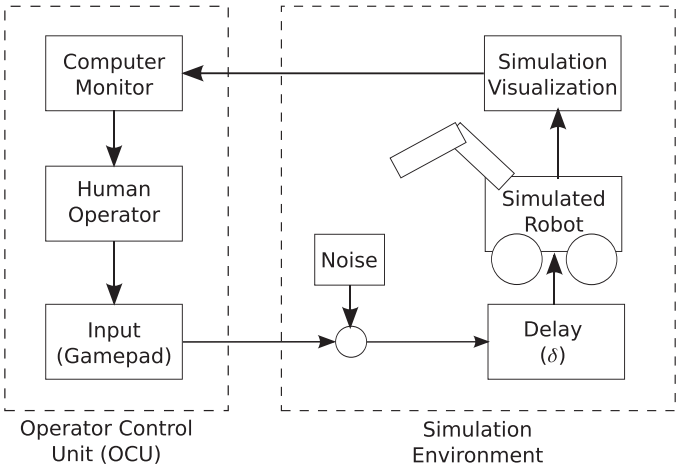


Fig. 1. Block diagram of the teleoperated robot simulation. Noise and delay are added to the input command from the user. There is no additional delay in the loop other than delays associated with sampling, display refresh rate, and computer processing, which are all negligible in comparison to the induced delay.

(a) Simulated Robot (b) Third-Person Viewpoint (c) First-Person Viewpoint

Fig. 2. Renderings of (a) the simulated robot, (b) the exocentric (third-person) viewpoint, and (c) the egocentric (first-person) viewpoint presented to the users in the study.

in a physical robot system. Because the focus of this study is to investigate steering behavior rather than combined longitudinal and lateral control, operators do not have control over the robot speed. The simulation visualization is displayed to the user via the Operator Control Unit (OCU) on a 25″ (63.5 cm) monitor with a resolution of 1920 × 1200 pixels in a full-screen window. Two different viewpoints are used in this study. A third-person view (Fig. 2(b)) shows the scene from a virtual camera following behind the robot, which can also be described as a "rigidly tethered" viewpoint.[42] A first-person view (Fig. 2(c)) shows the scene from the point of view of the robot's camera, with a small portion of the robot's gripper visible in the bottom portion of the window. Both viewpoints are aimed at a point having the same distance in front of the robot, giving both views identical lookahead distances.

3.1.3. Insertion of delay. Gamepad instruction packets are read by the OCU and enter a queue waiting to be read by the simulation, representing a delay in the human-to-robot direction (see Fig. 1). For each incoming instruction, a simulated delay (δ) inserted between the gamepad instruction and the simulation is determined by

$$\delta(\delta_{\min}, \sigma) = \delta_{\min} + |\delta_X| \tag{2}$$

where $\delta_{\min}$ is a minimum baseline delay, and $\delta_X \sim \mathcal{N}(0, \sigma^2)$ is a continuous random variable having a normal distribution with variance σ^2. This results in a stochastic latency distribution approximating the qualitative shape of packet intervals for wireless networks specified by the IEEE 802.11 standard.[43] A sample distribution of instruction delay values is shown in Fig. 3. These wireless networks characteristically appear to have a random distribution in addition to a baseline delay, the parameters of which depend on network type, distance between devices, the type of hardware used, and other environmental factors, such as interference from walls or other wireless networks. For simulation of

https://doi.org/10.1017/S0263574716000679 Published online by Cambridge University Press

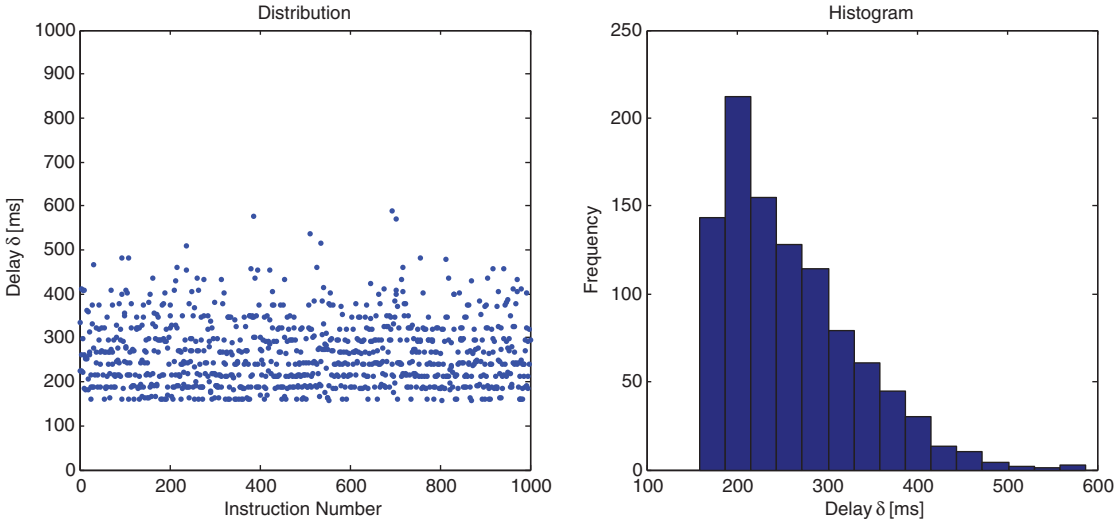


Fig. 3. Distribution of gamepad instruction packet delays with $\delta_{\min} = 150$ ms, $\sigma = 125$ ms, and mean delay $E[\delta] = 250$ ms. The quantization is due to the gamepad sampling rate of 40 Hz, which also causes the delay minimum (and therefore the mean delay) to be slightly greater (<10 ms) than the nominal value (negligible compared to the induced delay).

constant latency, we set $\sigma = 0$ such that $\delta = \delta_{\min}$. Once the delay is determined for the current time step, the newest gamepad instruction in the queue that is at least δ milliseconds old is used as the command input to the driving simulation, and older instructions are discarded. If no instruction is older than the desired delay, the previous instruction is used until the oldest instruction is older than δ milliseconds.

This induced delay is added to the system and does not include or compensate for delays due to the simulation rate (60 Hz), gamepad sampling rate (40 Hz), or display refresh rate (15 Hz), which are negligible compared to the magnitude of the latencies induced in the trials.

3.1.4. Test track. Sixteen non-self-intersecting test tracks were randomly generated that each contains exactly one of the following elements: Right Turn, Left Turn, U-Turn Right, U-Turn Left, S-Turn Right-Left, and S-Turn Left-Right. All turns have a constant radius of 2 m, and the width of the track is 2 m, with 0.125 m borders on either side. This is similar to the tests described by ASTM Standard E2829-11 for evaluating robot mobility in maneuvering at sustained speeds,[44] which uses a figure-eight track with a 2 m turning radius. The width of the robot (wheel-to-wheel) is 0.74 m. The turn gain of the gamepad input is scaled by robot speed such that the minimum turning radius of the robot is always 1.6 m, preventing users from relying on the actuator limits of the gamepad to execute ideal turning motions. Each track element has a section of straight-line path at least 5 m long immediately following it to allow the user to try to recover from any deviation sustained during the turn. Additionally, there is a 10 m straight-line warm-up section at the start of each track in which the user can familiarize him/herself with the test condition. A sample track is shown in Fig. 4.

3.1.5. Scoring. The path-following score for each trial is determined as a function of the robot's distance from the centerline over the course of the path, after the robot has passed the start line. The normalized score at each time step i of the simulation is given by

$$S_i = \max\left(0, \frac{w/2 - |y_i|}{w/2}\right) \tag{3}$$

where y_i is the lateral displacement at step i, and w is the width of the test track (in this case, 2 m). Then, the total score is determined as the average of the scores at each time step:

$$S = \frac{1}{n}\sum_{i=1}^{n} S_i. \tag{4}$$

https://doi.org/10.1017/S0263574716000679 Published online by Cambridge University Press

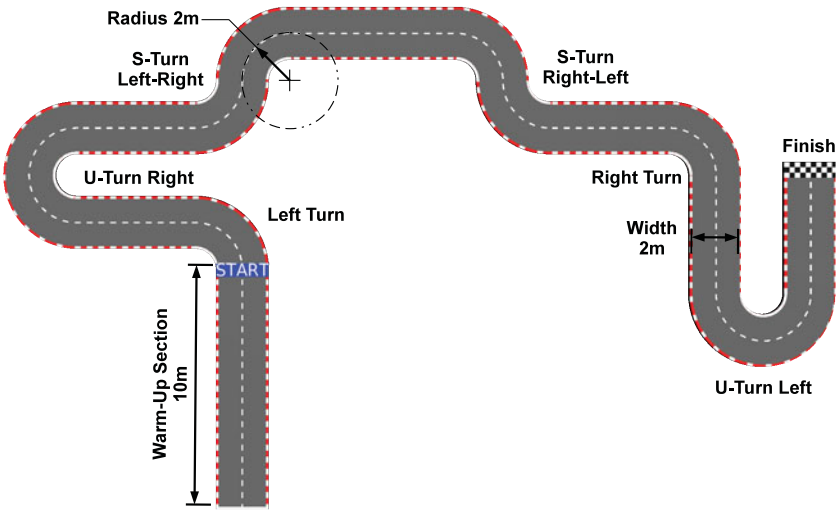


Fig. 4. Representative track with dimensions for simulated driving tasks. Sixteen total tracks were randomly generated for use in the study, all with the same set of features and dimensions. A section was included at the beginning of each track to enable users to familiarize themselves with the test conditions. Scoring for each trial commenced after the robot passed the "Start" line.

A score of 1 indicates that the path was followed perfectly, and a score of 0 indicates that the robot was never on the test track. Saturation was included to prevent high score variations as a result of singular mistakes. This metric was chosen as a simple representation of task performance as measured by path-following ability, and its similarity to traditional metrics is used for position tracking in teleoperation tasks,[19,45,46] as it is equivalent to the saturated mean error. It is also straightforward to explain to users that might not be familiar with metrics such as root mean square error (RMSE), and is consistent with the 'high-score' incentivization used in video games. A post-hoc comparison showed a strong correlation between RMSE and Path-Following Score for these trials ($r(493) = -0.98$), $p < 0.001$).

3.2. Procedure

User tests were conducted with 32 volunteers recruited via flier and email advertisements distributed to a population of engineering students. One participant withdrew from the study, leaving 31 users in the dataset. A total of 22 men and 9 women completed the tasks, ranging in ages from 18 to 37 years, with a mean age of 23.5 years (S.D. = 4.2 years). Participants self-reported an average prior experience with video games of 4.8 on a scale from 1 (least) to 7 (most) (S.D. = 1.3), and an average prior experience with robotics as 3.4/7 (S.D. = 1.2). Users were given $10 for participating, with the knowledge that an additional $10 bonus would be awarded to the top performer of six different tasks as determined at the end of the trials, with a $30 bonus cap. The tests were designed to take less than 1 h, and most participants needed approximately 45 min. These tests were approved by the University of Michigan Health Sciences and Behavioral Sciences Institutional Review Board (UM IRB #HUM00044265).

3.2.1. Study design. This study used a test design with three independent variables: viewpoint – first- and third-person; latency type – see Table I; and robot speed – 1.0 m/s and 1.5 m/s. These speeds are within the range used by ASTM E2829-11,[44] and are within the middle 50% of maximum speeds for common UGVs, as reported by the Association for Unmanned Vehicle Systems International.[47] To make the study more efficient, both speed levels for a given viewpoint-latency scenario were tested consecutively, and a survey was administered once per scenario. Due to time constraints, six of the scenarios were performed by all users (these were used as the basis for the bonus payments), and the remaining two scenarios for each user were evenly drawn from the secondary set of scenarios (see Table II). Users were not informed which scenarios were common to all participants. The order of the viewpoint-latency scenarios, speeds, and track numbers was randomized and counterbalanced to control for learning and fatigue effects.

https://doi.org/10.1017/S0263574716000679 Published online by Cambridge University Press

Table I. List of latency types used in the user study. All values are listed in milliseconds. $E[\delta]$ is the expected value of the random variable δ, representing the delay inserted between the user and the robot for each latency type.

Latency type	$\delta_{\min}$	σ	$E[\delta]$
A	0	0	0
B	250	0	250
C	500	0	500
D	750	0	750
E	150	125	250
F	300	250	500

Table II. Number of users participating in each scenario. All users participated in six of the scenarios (starred), while the remaining six scenarios were distributed evenly among the participants.

	Constant latency				Variable latency	
Visualization POV	A	B	C	D	E	F
Third person	31*	11	31*	10	31*	11
First person	31*	10	31*	10	31*	10

3.2.2. Test procedure. Test subjects were informed of the nature of the experiment, including the scoring mechanism of the trials.

For each trial, users would push a button on the gamepad to initiate driving, and would steer the robot along the test track with a joystick on the gamepad, at a constant pre-determined speed, attempting to keep the center of the robot in line with the center of the test track. If the center of the robot passed completely off the track, the speed of the robot was automatically reduced to half of the original speed, and returned to normal when the user was able to get the robot back onto the track. Without this reduction, some users would have been unable to recover from large errors, and their test results would not have been usable. Users were not informed of their trial scores during the tests to avoid biasing their survey responses.

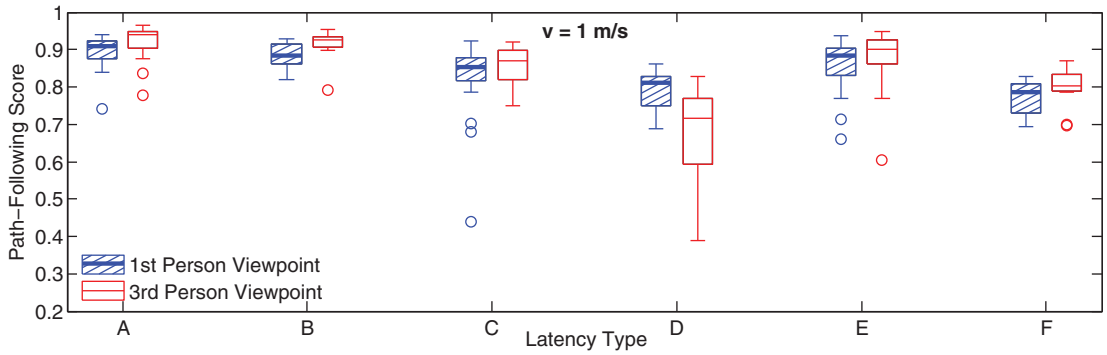
Data logs recorded the user's input to the robot as well as the robot state during all tests, and post-trial surveys were administered on the OCU via web browser. One data log for latency scenario C with a first-person view at a speed of 1.5 m/s did not record properly, so this trial is omitted from the dataset, but the survey results are included.

3.3. Results and discussion

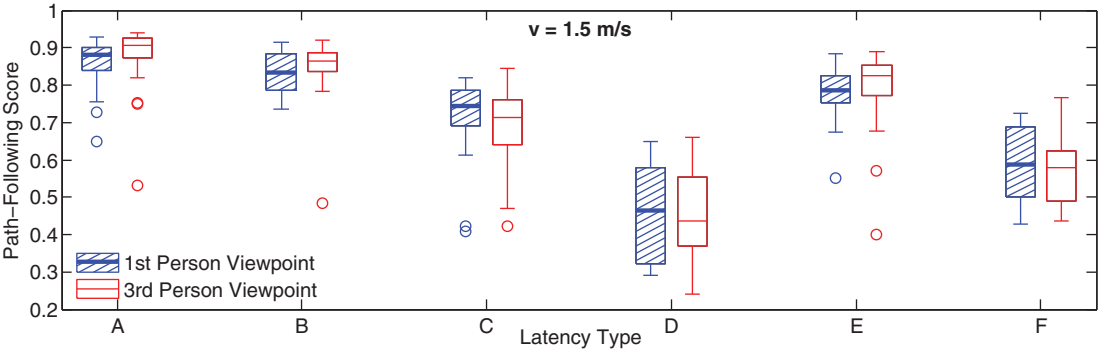
3.3.1. Task performance. To begin the analysis of human subjects' objective performance, we observed their path-following scores for the different viewpoint, latency and speed conditions. Figure 5 shows boxplots of trial scores under each viewpoint, latency, and speed combination. The plot shows decreasing scores and increasing score variance for increasing amounts of latency. Both of these trends agree qualitatively with prior experiments on task completion time in a 3D navigation task performed by Lane *et al.*[16] There is a clear decrease in performance for higher robot speeds under all latency conditions, and the performance drop-off is particularly stark above 500 ms of delay. Scenarios E and F show worse performance than B and C, respectively, indicating the scores for trials under variable latency are lower than the scores for trials under constant latency with the same mean.

In contrast to our expectations, Fig. 5 shows that the viewpoint did not have a pronounced effect on performance in our data. An analysis of variance, considering the user as a random variable, confirmed that the effect of viewpoint on path-following score was not statistically significant ($F_{1,494} = 2.31$, $p > 0.1$). This comes despite videogamers' general preference for a third-person point of view,[49] and does not replicate the prior findings of Pazuchanics[50] and Hollands and Lamb.[42] We believe that this discrepancy could have arisen due to the relative simplicity of this task, and the fact that robot arm's gripper was visible in the first-person view, providing a reference point to the

https://doi.org/10.1017/S0263574716000679 Published online by Cambridge University Press



(a) User results for 1 m/s speed.



(b) User results for 1.5 m/s speed.

Fig. 5. Boxplot showing the path-following score, as defined by Eq. (4), indicating user performance under varying latency and viewpoint conditions. For all boxplots in this paper, the center line of each box represents the data median, while the edges of the box correspond to the 25th and 75th percentiles (the interquartile range, IQR), and the whiskers extend to the most extreme data points within 1.5 IQR of the 25th and 75th percentiles. Data points outside this range are considered outliers.[48] (a) User results for 1 m/s speed. (b) User results for 1.5 m/s speed.

user that is similar to the types of references available in a third-person view. As a result, we combine data from the two viewpoints for the remainder of the analysis in this paper, focusing on delay type only.

3.3.2. Survey responses. Figure 6 shows the participant responses to the questions on the survey measuring the teleoperators' sense of the latency on a seven-point Likert scale. Responses for the different speeds were combined for each latency type due to the structure of the experiments. The responses show good internal consistency in measuring how much delay the user felt in the system (Chronbach's $\alpha = 0.87$). Responses, summarized in Fig. 6, indicate a higher mean delay value caused users to feel that there was a higher delay in the system. Additionally, responses indicate that users sensed there was a higher delay in scenarios involving variable latency than in scenarios having constant latency with the same mean. For example, users agreed with the statement "There was a lot of delay between my actions and the expected outcome in the robot's environment" more (the response distribution is shifted more to the right) for latency scenario E than latency scenario B.

4. Driver Model

This section discusses the development and validation of a model for simulating the steering commands issued by the teleoperator under different constant latency conditions, based on user data described in Section 3.

4.1. Driver behavior

To act as an acceptable substitute for a human driver, the steering model must accurately replicate the key characteristics of the human operators. Figure 7 shows two example datasets from runs under

https://doi.org/10.1017/S0263574716000679 Published online by Cambridge University Press

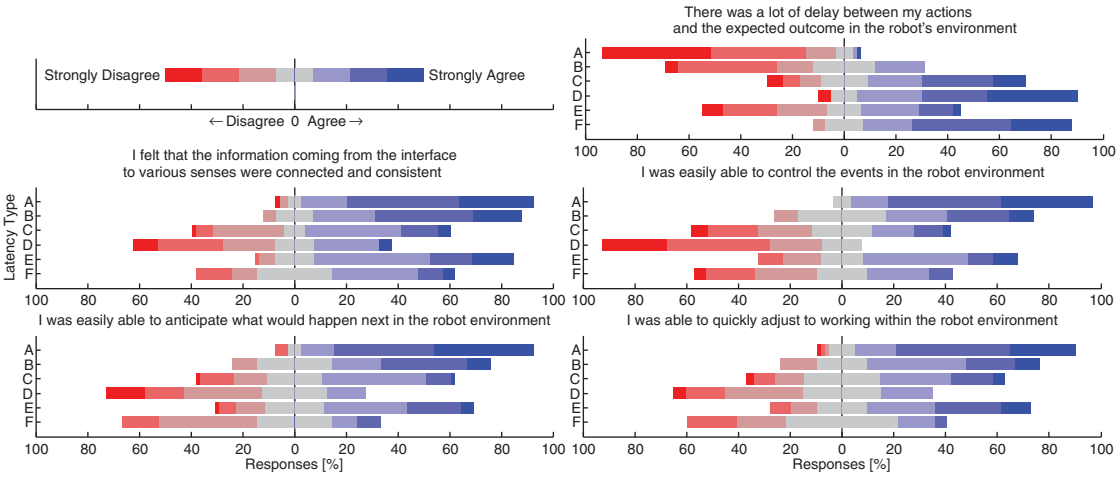


Fig. 6. User responses to questions designed to assess how much delay the user felt in the system for each latency type. The survey with seven-point Likert items was administered at the conclusion of the two speed trials for each latency scenario. The width of each bar represents the percentage of responses with the corresponding Likert value.

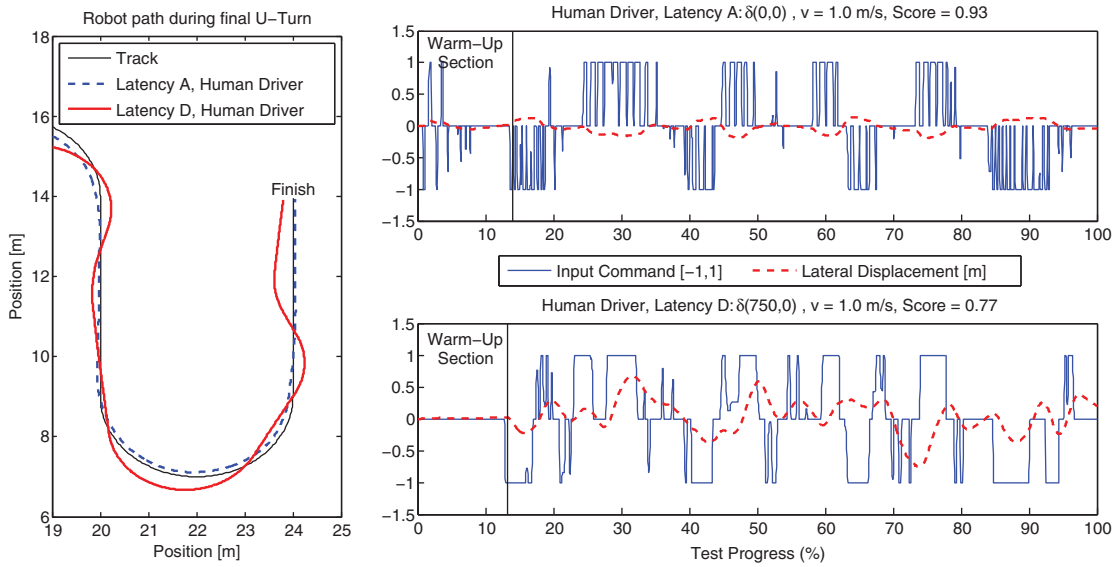


Fig. 7. Plots showing example datasets of low-latency and high-latency test cases. The datasets are from two different users and are representative of the median user performance in the test. Scores were not accumulated during the warm-up section. The path shown on the left is only a section of the whole track. Note that even though the operators could use the joystick to command any value between -1 and 1 to the robot, users generally only toggled to 0 or ± 1.

low latency (Latency A), and under high latency (Latency D), which were chosen because their path-following scores were close to the median user path-following score for their respective latency types.

The first characteristic that the model should accurately reproduce is the profile of the lateral displacement of the robot along the path. This can be represented by the path-following score, but the simulated displacement should also show similar patterns to the measured data. Two convenient measures to characterize the shape of the path are the maximum lateral displacement (maximum error) and the length of the path (meandering paths will be longer).

We also wish to emulate the users' input commands. As shown in both traces in Fig. 7, the input command is almost always saturated. This is because most users tended to move the control stick on the gamepad to its far left or right position rather than use an intermediate input. Thus, we characterize

https://doi.org/10.1017/S0263574716000679 Published online by Cambridge University Press

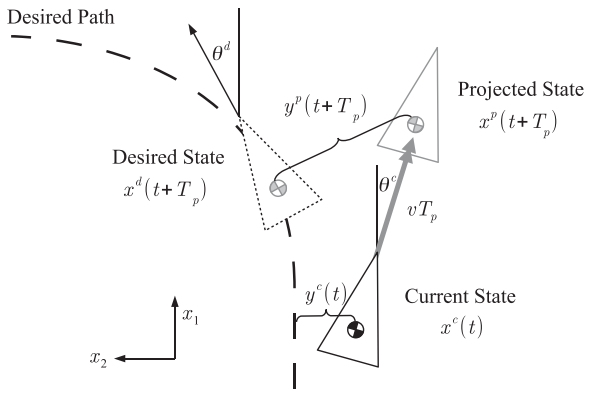


Fig. 8. Diagram illustrating the determination of the projected lateral displacement. The projected state is the location of the robot at a future time $t + T_p$, assuming a constant angle and velocity. The desired state is the position and orientation of the desired path that is closest to the projected state. The projected lateral displacement $y^p(t + T_p)$ is then defined to be the perpendicular distance between the desired state and the projected state.

the input command by both the average absolute input ([0–1]) and the rate at which the user toggled the control stick in Hz.

4.2. Model development

To develop a model for teleoperated robot steering, we can draw from some of the techniques previously developed for automotive steering models. Specifically, we use a preview of the desired path combined with an internal model of the vehicle kinematics and an operator time delay.[31,51] MacAdam[6] notes that essential requirements for a driver model include: a time delay due to human processing, a preview of the upcoming control requirements (i.e., a mechanism for feed forward control), the ability to adapt to different vehicle and operating conditions, and an internal model to predict vehicle responses. Automotive steering models can use one or more feedback cues as inputs to the driver model, including any or all of the following: lateral displacement, lateral acceleration, roll angle, heading angle, and yaw rate. The cues can be processed by the model in one or more forms, which may include visual cues, motion effects, sound, and tactile information.[6] However, in this teleoperation scenario, visual signals are the only cues available.

A simple steering model can be developed based on the driver's anticipated deviation from the desired path at some future time $(t + T_p)$. In this case, we choose the projected lateral displacement $y^p(t + T_p)$ of the robot as the feedback cue. Figure 8 illustrates the process of determining the projected lateral displacement, which is based on the projected state of the robot $x^p(t + T_p)$, assuming it continues its trajectory from the current state $x^c(t)$ at a constant velocity:

$$x^p(t + T_p) = \begin{bmatrix} x_1^p \\ x_2^p \\ \theta^p \end{bmatrix} = \begin{bmatrix} x_1^c \\ x_2^c \\ \theta^c \end{bmatrix} + \begin{bmatrix} v\cos\theta^c \\ v\sin\theta^c \\ 0 \end{bmatrix} T_p. \tag{5}$$

The desired future state of the robot $x^d(t + T_p)$ is the point along the desired path closest to the projected state, as measured by Euclidian distance:

$$x^d(t + T_p) = \underset{x \in path}{\arg\min} \sqrt{(x_1 - x_1^p)^2 + (x_2 - x_2^p)^2}. \tag{6}$$

The projected lateral displacement is the component of the difference between the projected and desired states perpendicular to the direction of the desired path. It is obtained by rotating the vector from x^d to x^p by the desired heading angle θ^d and taking the component perpendicular to the path:

$$y^p(t + T_p) = (x_2^d - x_2^p)\cos\theta^d - (x_1^d - x_1^p)\sin\theta^d. \tag{7}$$

https://doi.org/10.1017/S0263574716000679 Published online by Cambridge University Press

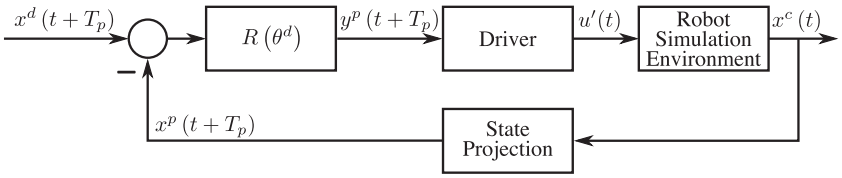


Fig. 9. Block diagram showing the steering control loop. The lateral displacement $y^p(t+T_p)$ is determined from the difference between the projected and desired robot states at time $t+T_p$. The $R(\theta^d)$ block represents the rotation operation described in Eq. (7). Delay and noise are added in our system inside the Robot Simulation Environment block.

For a continuous path, this is equivalent to taking the length of the vector, but for paths consisting of a discrete set of points, this method results in smaller computational errors due to gaps in the path.

We now model the steering action of the user as a PD controller[52] based on the anticipated lateral displacement feedback cue, with an additional delay δ_H representing the driver's physical actuation time:

$$u(t+\delta_H) = K_p y^p(t+T_p) + K_d \dot{y}^p(t+T_p) \tag{8}$$

The control signal generated by Eq. (8) is continuous and unbounded. However, the gamepad input device is only capable of generating control inputs on the interval $[-1, 1]$, and users tend to issue commands at one extreme of the interval or the other. We can capture both the actuator saturation and the users' tendency to max out the limits of the gamepad by conditioning the control input with a simple threshold ($\mu > 0$):

$$u'(t) = \begin{cases} -1 & \text{if} \quad u(t) \le -\mu \\ 0 & \text{if } \mu > u(t) > -\mu \\ 1 & \text{if } \mu \le u(t) \end{cases} \tag{9}$$

and using $u'(t)$ as the simulated gamepad steering command issued to the robot. Removing or modifying this conditioning could be used to model steering with other input devices. Figure 9 shows a block diagram of the overall steering control loop.

4.3. Model parameter tuning

In this section, we find PD gains to tune the controller to behave similarly to the median operator in our user tests for each constant latency scenario. We focus here on the case in which the robot speed is 1 m/s.

To simplify the process of tuning this model, we can make some assumptions about the parameters. The physical actuation time (δ_H) of the model driver is assumed to be 200 ms, determined by inspecting the log data of user inputs. Also, we assume a lookahead time (T_p) of 1250 ms based on prior work in the automotive domain,[53] which states human driver preview times are roughly between 500 and 2000 ms. Additionally, we set the threshold for conditioning the control input to $\mu = 0.5$, as this value resulted in qualitatively similar toggling behavior for the model as the human operators. Therefore, the only two parameters left to tune are the control gains K_p and K_d. The gains were tuned by hand to reflect the path-following score and average control input of the median user for each constant latency case. It would be possible to tune the controller to represent a more- or less-skilled user by changing the gains to match the measured performance the user at a given skill level. A summary of the model constants and tuned parameters is shown in Table III. Additionally, a two-dimensional search was performed in simulation over values of K_p and K_d. The controller gains in these simulations that most closely matched the median human operator behavior (for both path-following score and control effort) were found to be similar to the gains determined by hand-tuning the controller.

For the zero latency case, the K_d value of zero is consistent with vehicle steering models having only proportional feedback to errors in projected lateral displacement.[54] For the scenarios with latency, the ratio of K_p/K_d decreases as the latency increases, demonstrating that the steering model more heavily weighs the projected error for low latency, and relies more on the predicted displacement

https://doi.org/10.1017/S0263574716000679 Published online by Cambridge University Press

Table III. Tuned control gains and parameter values for constant latency cases.

Type	δ [ms]	K_p	K_d	T_p [ms]	δ_H [ms]	μ
A	0	1.7	0.0	1250	200	0.5
B	250	1.6	0.3	1250	200	0.5
C	500	1.3	0.7	1250	200	0.5
D	750	1.0	0.9	1250	200	0.5

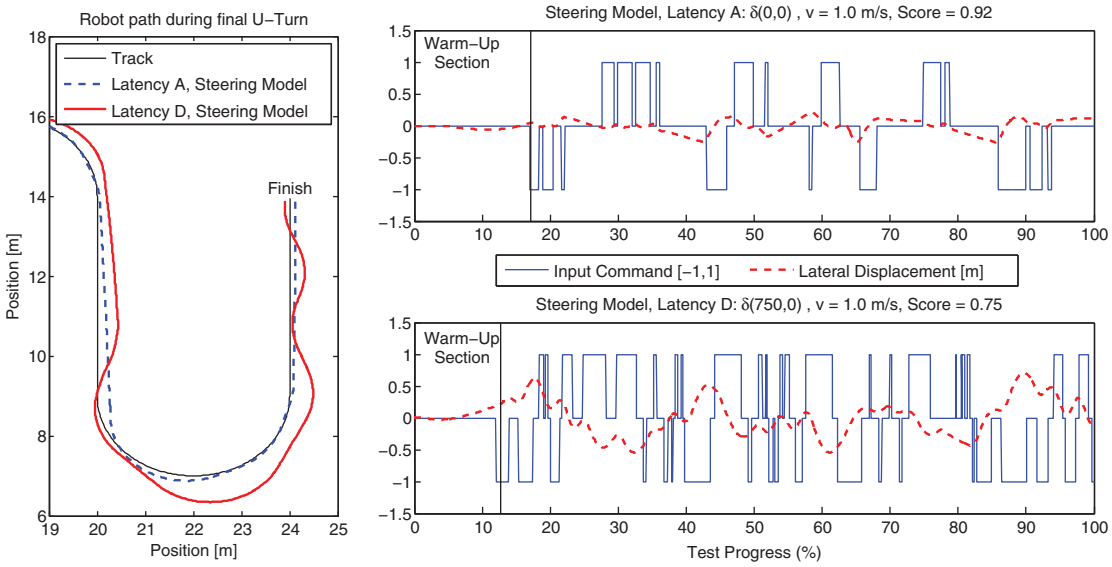


Fig. 10. Example paths and input profiles of the robot as commanded by the steering model. These paths and inputs show similar qualitative characteristics to those produced by human drivers. Note that the path shown on the left is only a section of the whole track.

trend when the delay is high. Intuitively, this reflects the strategy employed by a human teleoperator in the control loop, who must rely more on prediction based on anticipation of the track's features when the latency is high rather than direct visual feedback. Additionally, the decreasing proportional gain agrees with the qualitative analysis described by Jagacinski[55] and reflects the users' increased tolerance for steady state errors in the difficult-to-control high latency cases.

We performed a stability analysis on this steering model for constant latencies.[35] The analysis was carried out considering the robot traveling along a straight-line path and indicated that the maximum allowable delay, while maintaining stability, depends on robot speed, turn gain, lookahead distance, and control gains. More specifically, the criteria found agree that the ratio of K_p/K_d must decrease as latency increases.[35]

4.4. Model validation

To test the performance of the steering model, the teleoperation scenarios were run with the gamepad command simulated in real-time by a MATLAB script running the steering model at 40 Hz and communicating to the robot simulation via LCM over the gamepad channel. Each of latency scenarios A-D were run 20 times on different test tracks (randomly chosen from the 16 test tracks used in the user tests) with a robot speed of 1 m/s. Figure 10 shows two example datasets generated by the steering model, which appear similar to the datasets produced by the human drivers shown in Fig. 7. Path-following scores are similar for both latency scenarios A (human: 0.93; model:0.92) and D (human: 0.77; model: 0.75). Both the saturated input behavior and the overall lateral displacement profiles are qualitatively represented by the steering model.

Figure 11 displays both the human driver path-following scores and the path-following scores from the steering model. The spread of the steering model data at each of the constant latencies is smaller than that of the human driver. In its current form, the model does not capture the variability among all users who participated in the study. However, the model could also be tuned to match the

https://doi.org/10.1017/S0263574716000679 Published online by Cambridge University Press

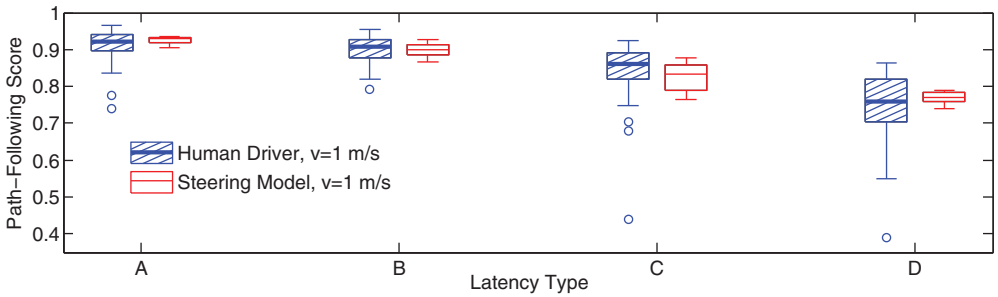


Fig. 11. Human driver scores of path-following alongside those from simulations of the robot at a speed of 1 m/s with input commands from the steering model. The model was tuned using the constant latency scenarios from the user trials. The gains used are given in Table III.

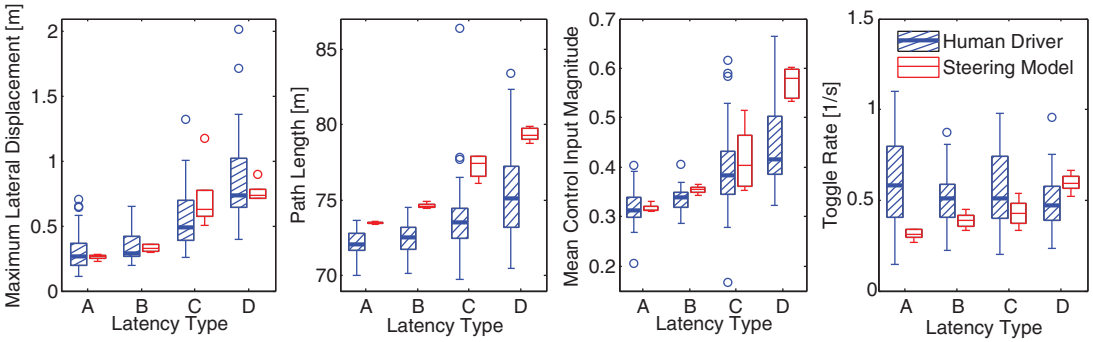


Fig. 12. Boxplot comparison between the human drivers and steering model for path characteristics of maximum lateral displacement (overshoot), path length, mean control input magnitude (control effort), and gamepad toggle rate.

behavior of users who scored in a different percentile, e.g., the model could be tuned to produce user scores in the 75th percentile to represent a more expert human operator.

The data in Fig. 11 show that simulations with the steering model line up well with the median human user for each of the constant latencies tested. For nearly all of the constant latencies tested, the interquartile range (IQR) of the model data falls within the IQR of the human drivers. The only exception is Latency C. However, the range of the model data still falls within the range of the human driver data. Figure 12 shows that the model, as tuned, accurately reproduces the overshoot (maximum lateral displacement) produced by the human driver in constant latency scenarios. Additionally, although they both show the same overall trends, the path lengths for the human operator are consistently shorter than for the steering model. This is likely because the human drivers in the trial were anticipating the turns and took the inside corner of the track more often than the model. If desired, the model could be tuned to be more anticipatory by increasing the derivative gain K_d. The steering model also shows reasonable agreement with the human drivers for average control input and toggle rate. Overall, the driver model appears to be a reasonable representation of a human driver under the conditions tested, and could be used to simulate teleoperator steering responses for evaluation of potential robot designs and technologies. Under different conditions of track geometries or input devices, we expect that the overall model would be applicable, but different parameter values may be required.

5. Variable Latency Equivalence

The model described in the previous section was developed and tuned for constant latency scenarios. However, the latency profile of communications networks is often stochastic. In this section, we use both experimental user data and steering model simulations to propose an equivalence relation for driver performance between variable and constant latency scenarios. Understanding the relationship between latency distribution and performance could greatly simplify the process of developing user models for telerobotic tasks with variable latency characteristics.

https://doi.org/10.1017/S0263574716000679 Published online by Cambridge University Press

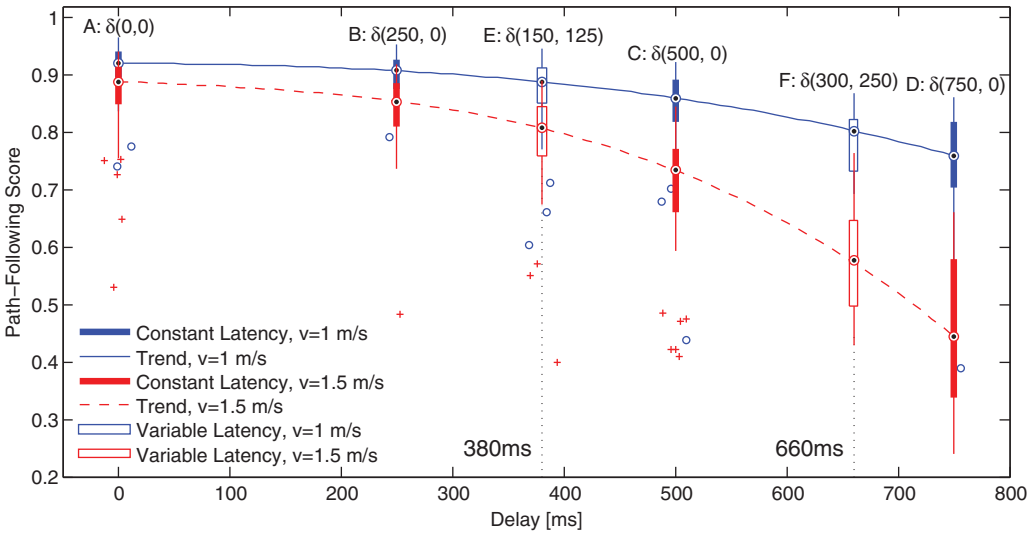


Fig. 13. Model of user performance as measured by the path-following score for various latencies and robot speeds. The data in this plot is the same as in Fig. 5, but the scores are now plotted by numerical latency value on the horizontal axis. The trendline runs through the median score for each constant latency case (A–D), and the variable latency cases (E, F) are then shown at their corresponding equivalent constant latencies.

5.1. Performance modeling and variable latency equivalence

Based on user scores for the constant latency scenarios tested, scores at intermediate constant latencies not explicitly tested in this study can be predicted by interpolation. Figure 13 shows the score distribution of the constant latency scenarios plotted with a trendline fit to the median user for each of the two speeds. By determining the point at which the median path-following scores for the variable latency scenarios intersect with the trend for the constant latency scenarios (see Fig. 13), we can find a magnitude of constant latency that would likely result in similar path-following performance as a given variable latency distribution. Figure 13 indicates that for variable latency scenario E, the median scores at the two different speeds correlate with the same constant latency, 380 ms. Similarly, latency scenario F corresponds to a constant latency of 660 ms for both speeds.

Figure 6 in Section 3.3.2 shows that users also reported experiencing the variable latency at a similar equivalence, in that the users' sense of the delay in the system followed the same trend as the objective score, wherein variable latency type E fell in between constant latency types B and C, and type F was between C and D. Figure 14 shows the sum of responses to these five Likert items (with the responses to "There was a lot of delay between my actions and the expected outcome in the robot's environment" reversed to make the sentiment of the statements consistent). A linear trend can be found between the constant latency scenarios and the sum of the ratings. By plotting the survey results of variable latencies E and F at their constant equivalent latencies of 380 and 660 ms, respectively, it is shown that the constant equivalences determined by the objective scores are consistent with the trend in the survey responses. This indicates that users' perceptions of the variable latency agree with their objective performance on the tasks, despite their not being informed of their scores during the trials.

Based on the quantitative path-following performance and qualitative survey responses, we conjecture that for this steering task, a variable source of latency can be mapped to an equivalent constant latency, independent of speed. This mapping could simplify the process of understanding user responses to delay for teleoperated tasks, as once an equivalence is found, it may be possible to use this equivalence instead of the latency distribution for modeling and predicting user behavior. Put another way, if a subset of latency distributions $\mathcal{D} = \{\delta_1, \delta_2, \ldots, \delta_n\}$ was found to result in the same user performance as some constant latency $\delta_c \in \mathcal{D}$, then characterizing system response to δ_c also characterizes the system response to each $\delta \in \mathcal{D}$, and the number of characterizations that must be performed is reduced from n to 1. Section 5.2, below, shows how we can find such subsets of latency distributions with equivalent steering performance using a small number of experimental tests. Note that this equivalence is with regards to the system performance under the different latency scenarios,

https://doi.org/10.1017/S0263574716000679 Published online by Cambridge University Press

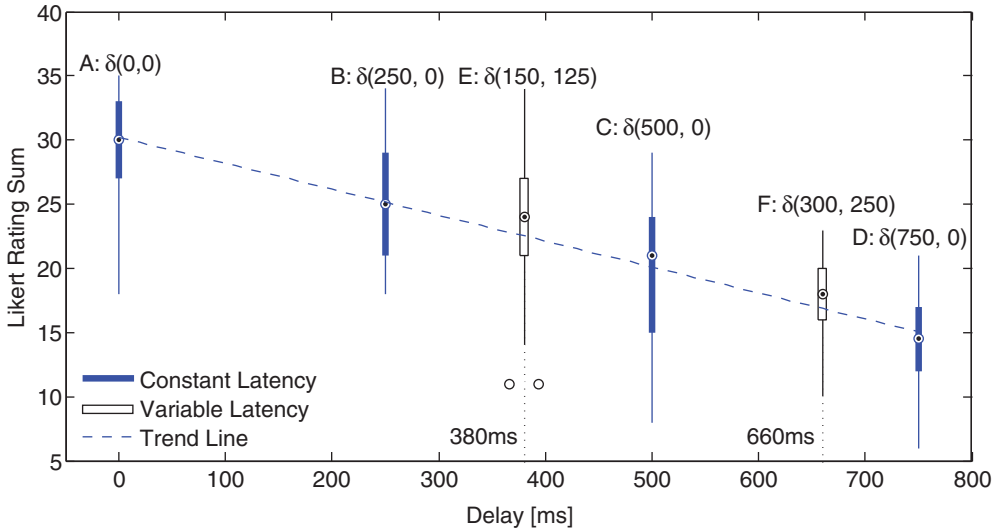


Fig. 14. Boxplot of user responses to survey questions related to operator sense of delay. The fit line is generated from the constant-latency cases (A–D), and the variable latency scenarios (E, F) are plotted at their constant latency equivalents, as determined by path-following score in Fig. 13.

and does not imply that pure delay and latency variability have equivalent underlying mechanisms resulting in decreased performance.

5.2. Expected performance for different latency distributions

The relationship determined above between human performance under variable latencies E and F and the expected performance under constant latencies of 380 and 660 ms inspired exploration to find similar patterns among several different variable latency distributions. Recall that the variable latency distribution used in this study $\delta(\delta_{\min}, \sigma)$ has two parameters: the minimum baseline delay $\delta_{\min}$ and the variance parameter σ^2. We predict that in terms of path-following score, there exists a relationship between the variable latency distribution and an equivalent constant latency δ_{equiv} that can be described with a linear function of $\delta_{\min}$ and σ^2:

$$f(\delta_{\min}, \sigma) = \alpha \cdot \delta_{\min} + \beta \cdot \sigma^2 = \delta_{\text{equiv}} \tag{10}$$

where α and β are scaling factors for $\delta_{\min}$ and σ^2, respectively.

The terms α and β can be estimated by solving a linear system of equations. Based on the results of our human subjects' tests, latency types E and F yield

$$\begin{aligned} f(150\text{ ms}, 125\text{ ms}) &= 380\text{ ms} \\ f(300\text{ ms}, 250\text{ ms}) &= 660\text{ ms}. \end{aligned} \tag{11}$$

We can select another two distributions with a known constant latency equivalence trivially:

$$\begin{aligned} f(380\text{ ms}, 0\text{ ms}) &= 380\text{ ms} \\ f(660\text{ ms}, 0\text{ ms}) &= 660\text{ ms}. \end{aligned} \tag{12}$$

These four linear equations can be represented in matrix form $Ax = b$, where

$$A = \begin{bmatrix} 0.15 & 0.125^2 \\ 0.3 & 0.25^2 \\ 0.38 & 0 \\ 0.66 & 0 \end{bmatrix}, \quad x = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}, \quad b = \begin{bmatrix} 0.38 \\ 0.66 \\ 0.38 \\ 0.66 \end{bmatrix}. \tag{13}$$

https://doi.org/10.1017/S0263574716000679 Published online by Cambridge University Press

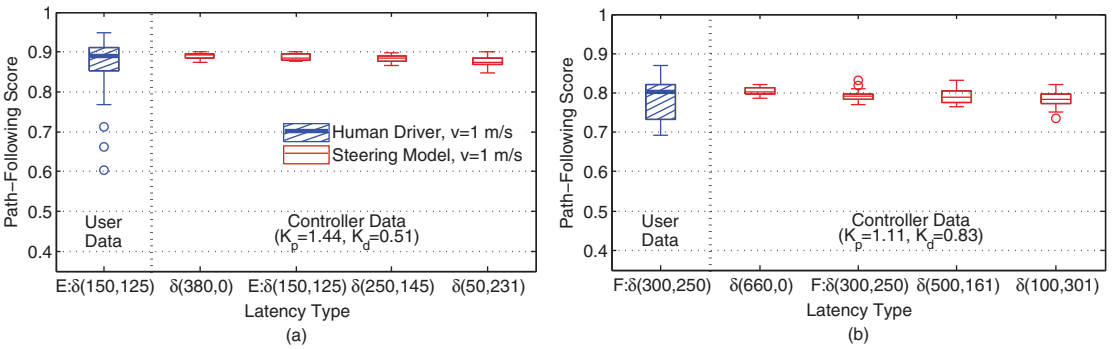


Fig. 15. Boxplot comparison between the steering model and human users for the proposed equivalent latencies. The gains used in the steering model for the distributions equivalent to latency types E and F were selected based on their equivalent constant latencies of 380 and 660 ms, respectively. The results for the model steering boxplots are based on 20 runs at each latency type. The user data for latency types E and F are based on sample sizes of 62 and 21 runs, respectively.

Since there is no exact solution to this set of equations, the least squares solution for α and β (defined as $\hat{x}^T = [\hat{\alpha} \quad \hat{\beta}]^T$) will be used: $\hat{x} = (A^T A)^{-1} A^T b$. The least squares solution of Eq. (13) yields $\hat{\alpha} = 1.02$ and $\hat{\beta} = 6.20$. Thus, the equivalent constant latency for any latency distribution of the type tested can be estimated.

To test whether these different latency distributions result in similar performance, 160 additional simulations were run and compared to user data for four new latency distributions. Based on Eq. (13), $\delta\,(50, 231)$ and $\delta\,(250, 145)$ are predicted to be equivalent to $\delta_{\text{equiv}} = 380$ ms, and $\delta\,(100, 301)$ and $\delta\,(500, 161)$ are predicted to be equivalent to $\delta_{\text{equiv}} = 660$ ms. Figure 15 displays user data from the variable latencies tested along with the results of the steering model (run 20 times each at the latencies displayed on the horizontal axis). The data from the steering model in Fig. 15 was simulated using controller gains linearly interpolated from Table III based on constant equivalent latencies of 380 and 660 ms, respectively.

Figure 15 compares the steering model at constant latencies of $\delta(380, 0)$ and $\delta(660, 0)$ to user data with variable latencies E and F, respectively. It shows that the steering model captures the variable to constant latency equivalence. Figure 15 also shows that the steering model is a good predictor of user performance when run with variable latencies. This is evident when comparing the human steering data with the model steering data each for latency E:$\delta(150, 125)$ and latency F:$\delta(300, 250)$.

Figure 15(a) supports our prediction that variable latencies of $\delta\,(50, 231)$ and $\delta\,(250, 145)$ would result in similar performance as latencies of $\delta\,(380, 0)$ and $\delta\,(150, 125)$. In fact, the median scores of the four latencies tested with the steering model are within 2% of the median human score (latency E) and nearly all of the path-following scores fall within the interquartile range (IQR) of the human users. Figure 15(b) also supports our prediction that variable latencies of $\delta\,(100, 301)$ and $\delta\,(500, 161)$ would result in similar performance as latencies of $\delta\,(660, 0)$ and $\delta\,(300, 250)$. The median scores of the four latencies tested with the steering model are within 3% of the median human score (latency F) and nearly all of the path-following scores fall within the IQR of the human users. Overall, Fig. 15 supports our claim that several variable latency distributions can result in similar path-following performance with the steering model developed in this paper. Therefore, steering performance results from simulations under a constant delay of (for example) 380 ms would be applicable to the subset of latency distributions satisfying Eq. (10) for $\delta_{\text{equiv}} = 380$ ms, reducing the need to test different latency distributions explicitly. Further work is required to investigate whether this variable latency-equivalent constant latency relationship holds for humans steering the robot.

6. Conclusions and Future Work

This paper presents the results of a user study exploring the effects of constant and variable latency on teleoperated steering tasks using a simulated mobile robot receiving input commands from a teleoperator via computer gamepad. Using fundamental concepts from automotive steering models, and examining the users' input commands to the simulated robot under different latency conditions,

https://doi.org/10.1017/S0263574716000679 Published online by Cambridge University Press

a model of human teleoperator behavior for a steering task was developed, tuned, and validated. The model is a PD controller with feedback based on the projected lateral displacement of the robot. The tuning of the model gains for different latency scenarios reflects the real-world control strategies that users employ when adapting to system latency.

The objective results of the study indicate that variable latency distributions can have equivalent performance to a particular constant latency, independent of speed. User responses to survey questions related to sense of delay further supported the proposed variable to constant latency equivalence. Results from simulations run with the steering model support the idea that variable latency distributions can be related to an equivalent constant latency through a linear combination of the delay minimum and variance for the latency distributions tested. These results serve as a foundation for future user tests that can explore whether the variable to constant latency equivalence relationship holds for users with respect to their performance.

It should be noted that the application of the results presented in this paper are constrained to the type of teleoperation scenario tested in the study. Vehicles having different steering characteristics (such as Ackerman vehicles), user interfaces with different input modalities (such as steering wheels), and scenarios having different latency ranges (such as space teleoperation experiencing multi-second delays) may not show the same behavior. Additionally, these results are limited to steering commands only, rather than full teleoperated driving control, or other types of teleoperation, such as remote arm manipulation. However, designers can use the models and techniques demonstrated in this paper to assist in robot design and optimization by rapidly simulating multiple design options without the need for full user testing. In the future, such models could also be used in conjunction with processed data from robot sensors to predict user intent to aid in human–robot collaboration.

This paper raises new questions about the relationships between system latency and operator performance. Thus, one area of future work is further exploration of the possible mapping between variable latency and equivalent constant latencies. It remains to be studied under which conditions of latency distribution, task type and difficulty, and output measures such an equivalence may exist. Additionally, more work can be performed on teleoperation driver models, including the validation of the steering model developed in this work by using more varied track configurations and latency scenarios, as well as more realistic robot simulations or hardware. Although one clear extension of this work is the development of longitudinal and/or combined (lateral and longitudinal) driver models for teleoperated mobile robots, it may also be possible to model teleoperators similarly in other contexts, such as pointing or object manipulation tasks. Additionally, input devices other than the gamepad, such as steering wheels, joysticks, and devices with bilateral (force) feedback could also be investigated.

Acknowledgments

This research was supported in part by the Automotive Research Center (ARC) at the University of Michigan, with funding from government contract DoD-DoA W56H2V-04-2-0001 through the US Army Tank Automotive Research, Development, and Engineering Center. The assistance of Mathew Ko in helping develop the simulation environment is acknowledged and greatly appreciated. Additionally, the authors wish to thank Dhananjay Anand for providing insightful advice and participating in valuable discussions concerning this research. Finally, we wish to thank our volunteer teleoperators (and beta testers), without whom this research would not have been possible.

References

1. J. P. Luck, P. L. McDermott, L. Allender and D. C. Russell, "An Investigation of Real World Control of Robotic Assets Under Communication Latency," *Proceedings of the 1st ACM SIGCHI/SIGART Conference on Human-Robot Interaction*, *HRI'06*, New York, NY, USA, ACM (2006) pp. 202–209.
2. J. Davis, C. Smyth and K. McDowell, "The effects of time lag on driving performance and a possible mitigation," *IEEE Trans. Robot.* **26**(3), 590–593 (Jun. 2010).
3. C. E. Harriott and J. A. Adams, "Modeling human performance for human-robot systems," *Rev. Human Factors Ergon.* **9**, 94–130 (Nov. 2013).
4. M. C. Yip, M. Tavakoli and R. D. Howe, "Performance analysis of a haptic telemanipulation task under time delay," *Adv. Robot.* **25**(5), 651–673 (2011).

5. T. Yang, Y. Fu and M. Tavakoli, "Digital versus analog control of bilateral teleoperation systems: A task performance comparison," *Control Eng. Pract.* **38**, 46–56 (2015).
6. C. C. MacAdam, "Understanding and modeling the human driver," *Veh. Syst. Dyn.* **40**(1–3), 101–134 (2003).
7. J. Chen, E. Haas and M. Barnes, "Human performance issues and user interface design for teleoperated robots," *IEEE Trans. Syst. Man Cybern. Part C: Appl. Rev.* **37**(6), 1231–1245 (2007).
8. S. Vozar, A Framework for Improving the Speed and Performance of Teleoperated Mobile Manipulators *Ph.D. Thesis* (Ann Arbor: University of Michigan, Aug. 2013).
9. T. B. Sheridan and W. R. Ferrell, "Remote manipulative control with transmission delay," *IEEE Trans. Human Factors Electron.* **HFE-4**(1), 25– 29 (Sep. 1963).
10. E. Slawiñski and V. Mut, "Control scheme including prediction and augmented reality for teleoperation of mobile robots," *Robotica* **28**(01), 11–22 (2010).
11. Y. Xiong, S. Li and M. Xie, "Predictive display and interaction of telerobots based on augmented reality," *Robotica* **24**(04), 447–453 (2006).
12. M. A. Sheik-Nainar, D. B. Kaber and M.-Y. Chow, "Control gain adaptation in virtual reality mediated human–telerobot interaction," *Human Factors Ergon. Manuf. Service Ind.* **15**(3), 259–274 (2005).
13. M. A. Goodrich, D. R. Olsen, J. Crandall and T. J. Palmer, "Experiments in Adjustable Autonomy," *Proceedings of IJCAI Workshop on Autonomy, Delegation and Control: Interacting with Intelligent Agents* (2001) pp. 1624–1629.
14. M. Marge, A. Powers, J. Brookshire, T. Jay, O. C. Jenkins and C. Geyer, "Comparing heads-up, hands-free operation of ground robots to teleoperation," *Proceedings of Robotics: Science and Systems VII* (2012) p. 193.
15. B. Wang, Z. Li and N. Ding, "Speech Control of a Teleoperated Mobile Humanoid Robot," *Proceedings of 2011 IEEE International Conference on Automation and Logistics ICAL* (2011) pp. 339–344.
16. J. Corde Lane, C. Carignan, B. Sullivan, D. Akin, T. Hunt and R. Cohen, "Effects of Time Delay on Telerobotic Control of Neutral Buoyancy Vehicles," *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA '02*, vol. 3 (2002) pp. 2874–2879.
17. D. Lee, O. Martinez-Palafox and M. W. Spong, "Bilateral Teleoperation of a Wheeled Mobile Robot Over Delayed Communication Network," *Proceedings of the IEEE Internation Robotics and Automation ICRA*, (2006) pp. 3298–3303.
18. F. Hashemzadeh and M. Tavakoli, "Position and force tracking in nonlinear teleoperation systems under varying delays," *Robotica* **33**(04), 1003–1016 (2015).
19. F. Janabi-Sharifi and I. Hassanzadeh, "Experimental analysis of mobile-robot teleoperation via shared impedance control," *IEEE Trans. Syst. Man Cybern. Part B: Cybern.* **41**(2), 591–606 (2011).
20. D. Sanders, "Analysis of the effects of time delays on the teleoperation of a mobile robot in various modes of operation," *Ind. Robot: Int. J.* **36**(6), 570–584 (2009).
21. H. McCracken, I Drove Ford's Golf Cart in Atlanta (Note: I Was in Silicon Valley at the Time), *Fast Company* (Jan. 2015).
22. P. M. Fitts, "The information capacity of the human motor system in controlling the amplitude of movement," *J. Exp. Psychol.* **47**(6), 381 (1954).
23. J. Accot and S. Zhai, "Beyond Fitts' Law: Models for Trajectory-Based HCI Tasks," *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing SystemsCHI'97*, New York, NY, USA, ACM (1997) pp. 295–302.
24. S. Zhai, J. Accot and R. Woltjer, "Human action laws in electronic virtual worlds: An empirical study of path steering performance in VR," *Presence: Teleoperators Virtual Environ.* **13**(2), 113–127 (Apr. 2004).
25. A. Pavlovych and W. Stuerzlinger, "Target Following Performance in the Presence of Latency, Jitter and Signal Dropouts," *Proceedings of Graphics InterfaceGI'11*, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, Canadian Human-Computer Communications Society (2011), pp. 33–40.
26. D. Kaber, Y. Li, M. Clamann and Y.-S. Lee, "Investigating human performance in a virtual reality haptic simulator as influenced by fidelity and system latency," *IEEE Trans. Syst. Man Cybern. Part A: Syst. Humans* **42**(6), 1562–1566 (2012).
27. I. S. MacKenzie and C. Ware, "Lag as a Determinant of Human Performance in Interactive Systems," *Proceedings of the INTERACT'93 and CHI'93 Conference on Human Factors in Computing System*, New York, NY, USA, ACM (1993) pp. 488–493.
28. Y. Tipsuwan and M.-Y. Chow, "Control methodologies in networked control systems," *Control Eng. Pract.* **11**(10), 1099–1111 (2003).
29. X. Xie, S. Yin, H. Gao and O. Kaynak, "Asymptotic stability and stabilisation of uncertain delta operator systems with time-varying delays," *Control Theory Appl. IET* **7**(8), 1071–1078 (2013).
30. D. T. McRuer, "Human pilot dynamics in compensatory systems," Technical report, DTIC Document (1965).
31. C. C. MacAdam, "An optimal preview control for linear systems," *J. Dyn. Syst. Meas. Control* **102**(3), 188–190 (1980).
32. J. M. Van De Vegte, P. Milgram and R. H. Kwong, "Teleoperator control models: Effects of time delay and imperfect system knowledge," *IEEE Trans. Syst. Man Cybern.* **20**(6), 1258–1272 (1990).
33. I. Delice and S. Ertugrul, "Intelligent Modeling of Human Driver: A Survey," *Proceedings of the 2007 IEEE Intelligent Vehicles Symposium* (2007) pp. 648–651.

34. L. Brito Palma, F. Vieira Coito and P. Sousa Gil, "Low Order Models for Human Controller–Mouse Interface," *Proceedings of the 2012 IEEE 16th International Conference on Intelligent Engineering Systems INES* (2012) pp. 515–520.
35. S. Vozar and D. M. Tilbury, "Driver Modeling for Teleoperation with Time Delay," *Proceedings of the 19th IFAC World Congress* (2014) pp. 3551–3556.
36. F. E. Ritter, U. Kukreja and R. S. Amant, "Including a model of visual processing with a cognitive architecture to model a simple teleoperation task," *J. Cogn. Eng. Decis. Making* **1**(2), 121–147 (Jun. 2007).
37. ACT-r, http://act-r.psy.cmu.edu/ (Aug. 2014).
38. APRIL Robotics Laboratory, APRIL Laboratory: Autonomy * Perception * Robotics * Interfaces * Learning, http://april.eecs.umich.edu (Mar. 2012).
39. A. Huang, E. Olson and D. Moore, "LCM: Lightweight Communications and Marshalling," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems IROS* (Oct. 2010).
40. P. Ögren, P. Svenmarck, P. Lif, M. Norberg and N. E. Söderbäck, "Design and Implementation of a New Teleoperation Control Mode for Differential Drive UGVs," *Auton. Robots* (Nov. 2013), pp. 1–9.
41. iRobot Corporation, iRobot 510 PackBot – specifications (2012). Accessed online 2014.
42. J. G. Hollands and M. Lamb, "Viewpoint tethering for remotely operated vehicles effects on complex terrain navigation and spatial awareness," *Human Factors: J. Human Factors Ergon. Soc.* **53**(2), 154–167 (Apr. 2011).
43. D. Anand, M. Bhatia, J. Moyne, W. Shahid and D. Tilbury, "Wireless test results booklet," Technical report, University of Michigan ERC/RMS (2010).
44. E54 Committee, "Test method for evaluating emergency response robot capabilities: Mobility: Maneuvering tasks: Sustained speed," Technical report, ASTM International (2011).
45. H. I. Son, L. Chuang, A. Franchi, J. Kim, D. Lee, S.-W. Lee, H. Bulthoff and P. Giordano, "Measuring an Operator's Maneuverability Performance in the Haptic Teleoperation of Multiple Robots," *Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems IROS)* (2011) pp. 3039–3046.
46. V. Grabe, P. Pretto, P. R. Giordano and H. H. Bülthoff, "Influence of Display Type on Drivers Performance in a Motion-Based Driving Simulator," *Proceedings of the Driving Simulation Conference* (2010).
47. AUVSI, "Unmanned ground vehicles: Core capabilities & market background," Technical report, The Association for Unmanned Vehicle Systems International (Aug. 2013).
48. M. Frigge, D. C. Hoaglin and B. Iglewicz, "Some implementations of the boxplot," *Am. Statistician* **43**(1), 50–54 (Feb. 1989).
49. R. Rouse, III, "What's your perspective? *SIGGRAPH Comput. Graph.* **33**(3), 9–12 (Aug. 1999).
50. S. L. Pazuchanics, "The Effects of Camera Perspective and Field of View on Performance in Teleoperated Navigation," *Proc. Human Factors Ergon. Soc. Annu. Meet.* **50**(16):1528–1532 (Oct. 2006).
51. A. G. Ulsoy, H. Peng and M. Çakmakci, *Automotive Control Systems* (New York, NY, Cambridge University Press, 2012).
52. N. S. Nise, *Control Systems Engineering*, 4th ed. (Hoboken, NJ, John Wiley & Sons, 2004).
53. A. Ungoren and H. Peng, "An adaptive lateral preview driver model," *Veh. Syst. Dyn.* **43**(4), 245–259 (2005).
54. D. Toffin, G. Reymond, A. Kemeny and J. Droulez, "Role of steering wheel feedback on driver performance: Driving simulator and modeling analysis," *Veh. Syst. Dyn.* **45**(4), 375–388 (2007).
55. R. J. Jagacinski, "A qualitative look at feedback control theory as a style of describing behavior," *Human Factors: J. Human Factors Ergon. Soc.* **19**(4), 331–347 (1977).

https://doi.org/10.1017/S0263574716000679 Published online by Cambridge University Press