Effects of Onset Latency and Robot Speed Delays on Mimicry-Control Teleoperation

Daniel Rakita, Bilge Mutlu, Michael Gleicher
University of Wisconsin–Madison, Department of Computer Sciences
{rakita,bilge,gleicher}@cs.wisc.edu

ABSTRACT

In this paper, we study the effects of *delays* in a mimicry-control robot teleoperation interface which involves a user moving their arms to directly show the robot how to move and the robot follows in real time. Unlike prior work considering delays in other teleoperation systems, we consider delays due to robot *slowness* in addition to *latency* in the onset of movement commands. We present a human-subjects study that shows how different amounts and types of delays have different effects on task performance. We compare the movements under different delays to reveal the strategies that operators use to adapt to delay conditions and to explain performance differences. Our results show that users can quickly develop strategies to adapt to slowness delays but not onset latency delays. We discuss the implications of our results for the future development of methods designed to mitigate the effects of delays.

CCS CONCEPTS

• **Human-centered computing** → **Interaction paradigms**; • **Computer systems organization** → **Robotics**.

KEYWORDS

Teleoperation, motion retargeting, latency analysis

ACM Reference Format:
Daniel Rakita, Bilge Mutlu, Michael Gleicher. 2020. Effects of Onset Latency and Robot Speed Delays on Mimicry-Control Teleoperation. In *Proceedings of the 2020 ACM/IEEE International Conference on Human-Robot Interaction (HRI'20), March 23–26, 2020, Cambridge, United Kingdom.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3319502.3374838

1 INTRODUCTION

Teleoperation systems offer the ability to project user manipulations into remote, dangerous, or high-precision settings [13]. These systems have a long history of success in high-value applications. Extensive work has documented the demands of system performance on task performance. In particular, *delay* between user command and system action has been examined as a cause of performance degradation, and many strategies have been proposed to mitigate its effects. However, existing work has not explored the effects of different types of delays which may lead to different user coping strategies. Further, prior work has not considered natural, arm-scale teleoperation systems which hold potential for new applications.

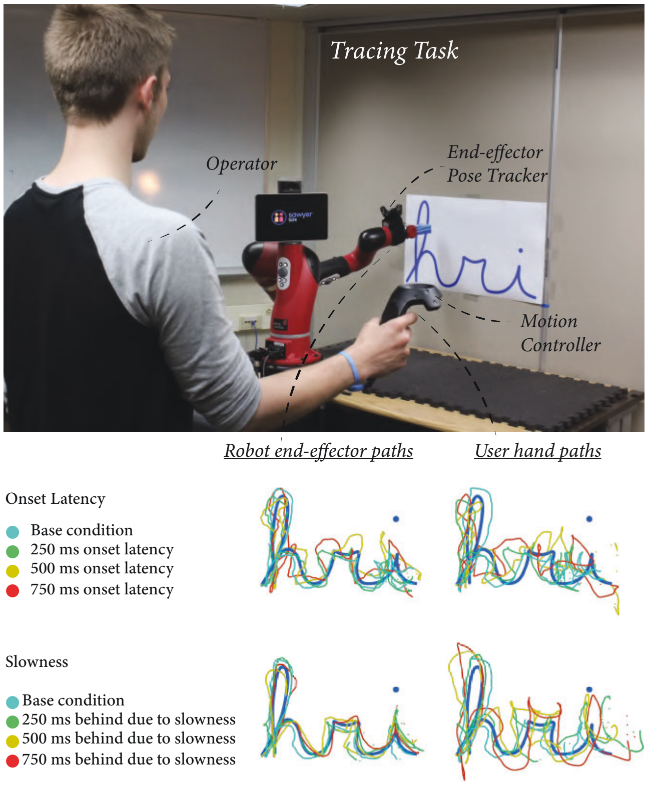

Figure 1: In this paper, we investigate the effects of *delays* in a mimicry-control robot teleoperation interface. (Top) In one of our evaluation tasks, participants controlled the robot to trace the cursive letters "hri" with varying degrees of onset latency and robot slowness. (Bottom) We analyzed the robot's end effector path as controlled by the user's input hand paths. Users adopted alternate motor strategies to control a robot arm considerably slower than their own arm, but no strategy emerged to overcome significant onset latency.

In this paper, we explore the effects of delays in teleoperation by studying an arm-scale teleoperation system with a direct mapping interface. Such *mimicry-control* systems map the six-DOF movement of the user's hand to the robot's end effector in real time. They are an emerging class of telemanipulation that has been shown to be successful even with novice users [16, 17, 22, 23]. Prior research on teleoperation delays has not considered such interfaces. The directness of these interfaces potentially places a high demand on the system performance; the slave robot must be responsive enough to the user's movements so that their natural arm control strategies still apply. Such demands may exacerbate the ill-effects of delays. However, the natural control strategies used by operators in such systems also offer a way to understand the effects of delay by examining how control under delay deviates.

Our premise is that different types of delays cause users to adapt in different ways, which in turn yields different task performance and may suggest different mitigation strategies. We consider two types of delays: *onset latency*, which is the time between when a command is issued and when the slave robot begins to execute the command, and *robot slowness*, which includes delays caused by the robot not executing movements as quickly as the user does. These types of delays occur, to varying degrees, in any practical system. Prior work has explored different amounts of delay, but not distinguished among different types of delays.

We conducted a human-subjects study that investigated how delay affects arm-scale teleoperation systems with a natural mapping interface. Participants performed tasks using a teleoperation system to which we systematically added different amounts and types of delays. We confirmed that the well-known negative effects of delays can be measured in task performance. More importantly, we see differences in how users are affected by different types of delay. These differences appear in task performance but can also be identified in the details of their movements. By analyzing the detailed movements, we were able to identify different strategies used by participants to control the robot in delayed conditions. The strategies apply to certain types of delays, explaining differences in task performance. Our results show that while users exhibit strategies to address slowness delays, they are much less able to apply strategies that address latency delays.

The central contribution of this paper is an extensive characterization of the effect of delays in a telemanipulation system. We provide a systematic study of a specific mimicry-control system that allows us to understand the effects of different types and amounts of delay on task performance and user perception. We identify motor adjustment strategies used by operators to adapt to delays that help explain the connection between delays and task performance.

2 RELATED WORKS

Analysis of Delays in Teleoperation—The value of teleoperation in performance-critical applications with communication delays has motivated a rich history of assessing effects when delays are present. Sheridan and Ferrell [25] first studied the effects of communication latency in a simple remote manipulation task. They showed that latency did affect performance, but people were still able to complete the task by adopting an alternate "stop-and-wait" strategy when delays were present. Hristu et al. [9] compared the effects of latency and *bandwidth*, i.e., the frequency with which information is sent back to a master device, on a finger-scale haptic device. They showed both sources of delay to be detrimental to task performance. Held and Durlach [6] also note the tendency for people to adapt their input strategies when objects are spatially or temporally skewed. Our work aims to extend this adaptation concept to mimicry-control and characterize input strategy adaptations that people exhibit in the face of onset latency or speed delays.

Significant work has attempted to characterize performance degradation in critical applications to assess when teleoperation may be feasible. See, for example, Lum [12] for a survey of surgical applications. Notably, Anvari et al. [1] show that people can successfully complete robot surgery tasks with 500 milliseconds of system delay, deviating from the common convention that exceeding 250 milliseconds of delay would degrade performance too much to be useful. The authors also mention that control strategies and performance significantly vary between people. Work has also investigated time delays in underwater and space teleoperation scenarios, as it takes a set amount of time for the signal to physically reach the controller a great distance away [8, 14, 24].

Other work devises methods for mitigating performance degradation due to delays. For example, Lee and Spong [11] introduced a controller to overcome fixed time interval delays. Another approach is the use of predictive displays that bridge the delay gap by showing a predicted view of what the robot will likely do in the near future [2, 3, 21]. We believe our work can inform the design of future mitigation strategies by helping better understand how and why delays lead to performance degradation.

Our work draws on this literature of characterizing effects of delays in teleoperation, but it differs in two main ways: (1) we investigate a different interface, namely mimicry-control, as opposed to standard interfaces such as six-DOF stylus devices or touchscreens; (2) we manipulate different types of delays, all prior work only manipulates communication delays. These differences allow us to identify user strategies that explain performance degradation.

Robot Control with Human Motion—Moving robots using human motion has a long history dating back to the early turn-crank master-follower device developed by Goertz et al. [4]. Since then, research has explored various scenarios that involve moving robots using human motion. For example, work in robotic surgery investigates systems that use specialized input devices to map and scale surgeons' motions to robots to perform minimally invasive procedures (see the work by Lanfranco et al. [10] for an overview). Pollard et al. [15] present a method to transfer human motion data and accompanying stylistic motion qualities to a humanoid robot, despite the discrepancy between human and robot degrees of freedom, joint velocity limits, and joint rotation ranges. Suleiman et al. [26] present a robot imitation approach and show that their analytical solution to the optimization converges quickly and effectively. Our prior work presents a non-linear optimization-based framework for mapping arm motion onto a robot manipulator. The key insight in this work is that exact mapping solutions are not feasible, but relaxed solutions close to exact solutions are sufficient and effective for facilitating motion-retargeting-based teleoperation [17]. We extended this method to add in shared-control aspects for motor task training as well as bimanual robot manipulators [19, 20]. Our current work uses this motion retargeting framework to study effects of delays on mimicry-control.

3 PRELIMINARIES

In this section, we define terms and concepts used throughout our work. First, we consider a *delay* during teleoperation as any

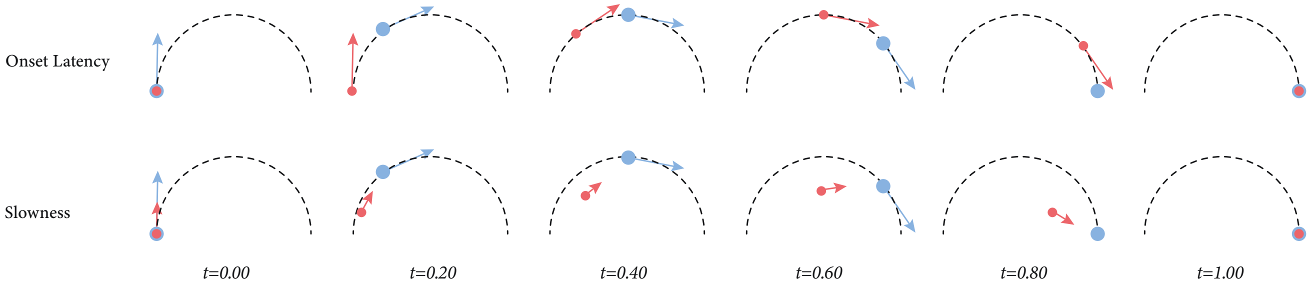

Figure 2: Comparison between (top) onset latency and (bottom) slowness delays.

temporal gap causing a misalignment between input commands and output motions. As discussed in §1, we consider *two* potential sources of delay in this work: (1) *onset latency* and (2) *robot slowness*. *Onset latency* is the delay between the instant that a command is issued and when the command begins to be executed. *Slowness* is the rate at which the slave moves toward the master. At any instant, the slave is moving toward the "current" (subject to the latency) estimate of the master position, at a velocity bounded by its slowness. We note that our definition of slowness delays differ from the concept of *bandwidth* by Hristu et al. [9] as slowness in our work is unrelated to the communication signal and related only to the capabilities of the robot as dictated by its joint velocity limits.

To illustrate the differences between these delays, consider an *n*-dimensional trajectory that a master device traces over time. As an example, Figure 2 shows a two-dimensional semi-circle trajectory traced by the "master," i.e., the blue dot, over the time of one second. The top row of this figure illustrates the effect of a 200 millisecond *onset* delay. Here, the "follower," i.e., the red dot, follows the same curve with the same velocities over time, but because the initial motion onset of the red dot was 200 milliseconds late, it remains this amount of time behind on the trajectory throughout its traversal.

The bottom row in Figure 3 illustrates the effect of a *slowness* delay. Here, both the master and follower dots start at the same time, but the velocity of the red dot is less than the velocity of the blue dot. At each time-step, the red dot greedily uses all of its allotted velocity to try to catch the blue dot, but because it is slower, it lags behind. It would be infeasible for the follower to trace the same path as the master in the slowness case since a follower with half the velocity would take *double* the time to follow the same trajectory. Note that *both* the onset latency and slowness examples illustrated in Figure 2 result in a 200 millisecond delay at the end of the trajectory, but they result from vastly different sources.

These onset latency and slowness definitions extend to trajectory following in any dimension. Notably, in this work we reason over delays in seven-dimensions to assess robot manipulator joint-space trajectories. Additionally, by mapping these high-dimensional joint-space configurations through the robot's forward kinematics model, we are able to visualize and analyze onset latency and robot slowness in three-dimensions by comparing the user's hand translation trajectories and the robot's end-effector translation trajectories, as seen in Figure 1. These analyses are covered in depth in §5.

4 USER STUDY

We conducted a user study to characterize the performance and user perception effects influenced by onset latency and robot speed in a mimicry-control interface. In this section, we provide an overview of our implementation details, study design, tasks that participants performed using the control interface, our measures, and the results. Throughout the study, we also collected a motion dataset that was used for additional analyses that are described in §5.

4.1 Implementation Details

The mimicry-control system used in our user study followed the same implementation details specified in our prior work [17]. Participants stood behind the robot and used an HTC Vive motion controller to guide the six-DOF pose of the robot's end-effector positions and rotations in real-time using their own arm motions. The robot's gripper could be opened and closed using the trigger on the controller. The participants' standing location was kept consistent for all tasks across all participants and was situated such that participants were always outside of the reach of the robot and could consistently see the workspace. The Vive motion controller pose information was collected in a Unity game engine environment on a Windows computer using the `SteamVR` library.

At each control update, a separate computer running Linux Ubuntu and using ROS for all message transferring collected the current translation, rotation, and button-interface information from the Windows computer over a UDP socket. The translation and rotation information from the motion controller was used to compute a new robot joint configuration that exhibits an end-effector pose that sufficiently matches the user's current hand pose. New joint configurations were calculated per-update using an optimization-based inverse-kinematics solver called RelaxedIK [18]. RelaxedIK is designed to solve for joint configurations that are both *accurate*, i.e., they sufficiently follow the end-effector pose goals dictated by the user's hand motions, as well as maintain *motion feasibility* over a sequence of solutions, i.e., the solver will avoid self-collisions, kinematic singularities, and joint-space discontinuities. Our system used the open-source implementation of RelaxedIK[1] and solved for new configurations at approximately 300 Hz.

We used a Rethink Robotics Sawyer in our study. The robot was actuated at 100 Hz using the velocity controller provided in the Intera API. Velocities were computed per update such that the robot at

[1] https://github.com/uwgraphics/relaxed_ik

its current joint configuration actuated toward the most recent joint configuration goal provided by RelaxedIK as fast as possible within its given velocity limits. Because our work aims to analyze how people adapt their motion inputs based on the modulated motion properties of the robot, we also recorded the robot's motion over time. We recorded the robot's joint configurations as specified by its encoders at controller each update, and also redundantly recorded the robot's end-effector position and rotation information using an external Vive Tracker sensor. Recording the robot's end-effector allowed us to ensure that the robot's joint states were accurately time aligned with the user's motion controller and to make precise, end-to-end latency measurements. Using this implementation, our evaluation of delay included the *total latency* of the system, including things like how long it takes to actuate the robot's joints, how long it takes to overcome the robot's inertia from a stopped position, etc. Prior works tend to only consider the *communication* latency of the system, and does not consider the full stack of effects that could cause a misalignment between inputs and outputs.

4.2 Hypothesis

Our central hypothesis was that delays would have detrimental effects, as suggested by the existing literature. We hypothesized that both task performance and user perceptions would decrease as delays increase, which is strongly supported by prior studies. We hypothesized that different types of delays would affect performance differently, based on our experience with different systems and our intuitions of why mimicry control is successful (e.g., different types of delays break the illusion of naturalness in different ways). We also hypothesized that we would see different strategies for adapting to delays, either in response to different amounts and types of delays or due to individual differences in participants.

4.3 Study Design

To test our hypothesis, we designed a 7×1 within-participants experiment in which naïve participants used seven control conditions in a random order to complete three physical tasks. Our conditions included a *base* condition where the system ran at its best performance with no artificially added onset latency or velocity limits. We included three conditions with different levels of added onset latency, which we call *latency1*, *latency2*, and *latency3*, and three different levels of slowness delays *speed1*, *speed2*, and *speed3*.

The latency conditions artificially added a delay between when a motion is observed and when it is communicated to the robot. We measured *total system latency* using a Vive tracking system by calculating the amount of time between when the controller moves and when a corresponding movement by the robot occurs. We measured the base condition to be approximately 90 milliseconds (the 80Hz sampling rate of the Vive limits the fidelity of the measurement). For the *latency* conditions, we added programmatic delays to cause the total system latency to reach a target value that was verified by measurement. We chose 250 milliseconds for *latency1* based on prior studies [e.g., 12] that report it as the level of delay where performance starts degrading, and 500 and 750 milliseconds were chosen for *latency2* and *latency3*, respectively, as equally spaced points to observe further degradation in performance.

The robot speed conditions synthetically slowed the robot down by reducing the robot's joint velocity limits in the velocity controller. Joint velocity limits were uniformly reduced to be approximately equivalent to the 250, 500, and 750 millisecond delays implemented in the latency conditions. To calculate these joint velocity scaling factors for a given delay, we drew 100 random configurations, and computed the output end-effector translation velocity limits at those configurations by mapping through the Moore-Penrose pseudoinverse of the robot's Jacobian matrix: $\mathbf{J}(\theta)^{\dagger} k\dot{\mathbf{v}} = \dot{\mathbf{e}}$. We then manually adjusted the value k such that the average end-effector translation velocities over all configurations results in the desired delay in the robot's physical joint velocity limits. The *base* condition used the robot's actual joint velocity limits, i.e., $k = 1$.

4.4 Tasks

To improve the applicability and generalizability of our findings, we developed three tasks that cover a wide range of manipulation scenarios. These tasks were *toy cleanup*, *stirring*, and *tracing*. All tasks had a maximum time limit of one minute. In the *toy cleanup* task, participants picked up two foam blocks and put them in the toy bin. This task was included as a pick-and-place task to assess both precise manipulation and grasping acuity.

The *stirring* task involved participants stirring five times clockwise in a bundt cake pan. The pan is approximately 0.3 meters in diameter and has a circular inner boundary of about 0.1 meters in diameter at the center. This task was included to observe how fluently participants could move along a simple circular curve.

The *tracing* task involved participants tracing the letters "hri" in lowercase cursive on a poster in front of the robot (as seen in Figure 1). The robot held a small foam block pointer in its gripper to indicate where the robot was pointing, and participants were instructed to follow the curve as closely as possible near the wall, while preferably avoiding collisions with the wall. The "hri" cursive writing was 0.65 meters wide and 0.4 meters tall. While the "i" was dotted on the acronym for recongnizability, participants were instructed not to dot the "i" such that the output end-effector curve was continuous from lower left to lower right. This task was included to assess the participant's fine-motor control accuracy over the robot's end-effector through a complex geometrical curve.

4.5 Procedure

Following informed consent, participants were provided with detail on the study goals and tasks. Before controlling the robot, participants were instructed on how to use the control interface. After expressing readiness, participants were presented with a training task of picking up a foam block from the table. This training task was selected due to its simplicity and the preparation it offered for the first study task. Each participant performed the training task at least one time and had the option of performing it up to two additional times. After the training, participants were introduced to the study tasks outlined above. Participants were then asked to perform the tasks with their hands to demonstrate their understanding. They then performed the tasks for each control condition on the robot in the order *toys*, *stirring*, then *tracing*. Condition order was randomized between participants. After each task, the experimenter reset the robot to the same initial configuration to standardize the starting point. After completing the experimental tasks using a particular control condition, participants filled out a questionnaire pertaining to that condition. At the end of the study, participants were debriefed about what we were studying, given the opportunity to ask any additional questions, and compensated.

4.6 Participants

We collected data from 21 participants (7 male, 14 female) from the University of Wisconsin–Madison campus with ages 19–41 ($M = 24.07$, $SD = 6.61$). Due to robot gripper malfunctions during some study trials, data from 24 of 147 total condition trials were lost (three or four conditions were affected for seven participants). Due to the randomized order and our within-participants study design, we do not expect this data loss to introduce systematic bias. Participants reported low-to-moderate familiarity with robots ($M = 3.08$, $SD = 1.44$, measured on a seven-point scale). Seven participants reported participating in prior robotics research studies. The study took 60 minutes, and each participant received $10 for their time.

4.7 Measures and Analyses

We used several metrics to characterize performance on the three tasks described in §4.4. In the *toy cleanup* task, we measured completion time and the number of times an object was knocked over (resulting in an object reset). During *stirring*, we measured completion time and the number of contacts between the spatula and either the outside or inside of the bundt pan. In the *tracing* task, we measured completion time and formulated a "tracing accuracy" metric that assessed how well participants followed the given curve.

The tracing accuracy metric, T is a sum of two sub-scores: a Cartesian accuracy score, c, and an order accuracy score, o. The Cartesian accuracy score projects each end-effector point onto the tracing curve and takes the average error distance of these projections. Because good tracing also consists of following the curve points *in order*, the order accuracy score, o, penalizes end-effector paths that jump around between various points on the path even if they are consistently geometrically close to the curve. We take the arc-length parameterization of the tracing curve, and, after projecting the end-effector point at each time-step onto the curve, associate an arc-length parameter value in $[0, 1]$ to the sequence of projection points. We then go through this sequence of arc-length parameter values and iteratively sum up errors to compute o using the following rule: if the arc-length parameter value at a given index is greater than its predecessor and less than its successor, i.e., it is in order, do not add anything to o; however, if the arc-length parameter value is *less than* than its predecessor or *greater than* than its successor, add the absolute value of the displacement between these arc-length parameter values to o. We normalize o and c and sum them together to get the final tracing accuracy metric: $T = o+c$, thus meaning the range of T is $[0, 2]$ where lower values are better.

To measure participants' perception of the different control methods, we administered a questionnaire based on prior research on measuring user preferences [7], including scales to measure fluency, trust, and predictability. Each scale included items measured on a seven-point rating scale (1 = strongly disagree; 7 = strongly agree). Fluency was measured using items "The robot and I worked fluently together as a team," and "The robot contributed to the fluency of the interaction" (Cronbach's $\alpha = 0.81$). The trust scale included items "I trusted the robot to do the right thing at the right time" and "The robot was trustworthy" (Cronbach's $\alpha = 0.91$). Finally, predictability was measured using items "The robot consistently moved in a way that I expected," "The robot's motion was not surprising," and "The robot responded to my motion inputs in a predictable way" (Cronbach's $\alpha = 0.93$). We also included a non-weighted TLX [5].

We analyzed data from all measures using one-way repeated-measures analyses of variance (ANOVA). All pairwise comparisons used Bonferroni Correction by multiplying the p-value generated from Student's t-test by six.

4.8 Results

Our results are summarized in Figure 3. The analysis provides partial support for our hypothesis, as at a high level onset latency delays were shown to significantly degrade performance and user perceptions across all tasks and measures, while significant differences were not observed as the robot was slowed down. For instance, the tracing accuracy measure shows that participants got significantly worse at tracing as more onset latency was present; conversely, while tracing time logically slowed down as the robot slowed down, tracing accuracy actually trended *better* as the robot's speed decreased. This same effect was observed in the stirring task and toys task based on the significant differences in the stirring time, number of edge hits, and toys time metrics. In fact, no participants were able to complete the toys task in the allotted time in the *latency2* and *latency3* conditions, while the robot slowness conditions exhibited no such performance degradation.

User perception effects, such as fluency, trust, predictability, and numerous factors of the TLX questionnaire, were shown be significantly affected by onset latency, but even the slowest robot condition generally trended favorably compared to the base condition across all measures. In the following section, we further investigate *why* these two sources of delay have vastly different effects on user performance and perceptions.

5 DATA ANALYSIS

Our results discussed in the previous section suggest that delays from communication latency more negatively affect performance and user perceptions than robot slowness. Our goal throughout this work was not only to determine whether these delays do or do not have an effect but to analyze *how* these types of delays affect performance. To this end, we conducted a data analysis procedure to investigate the differences between these two sources of delay at the motor strategy level. This section provides an overview of this data analysis procedure and summarize our findings.

Prior work has demonstrated that people adjust their motor strategies when visual feedback is either spatially or temporally skewed [6, 25, 27]. Building on this principle, we speculated that as people control the robot using their own motions as inputs in the face of onset latency or robot speed delays, they would show some *adaptation* in their motor input command strategy. Our goal was to find patterns in the data collected in our user study to indicate signs of such motor adaptation strategies when delays are present.

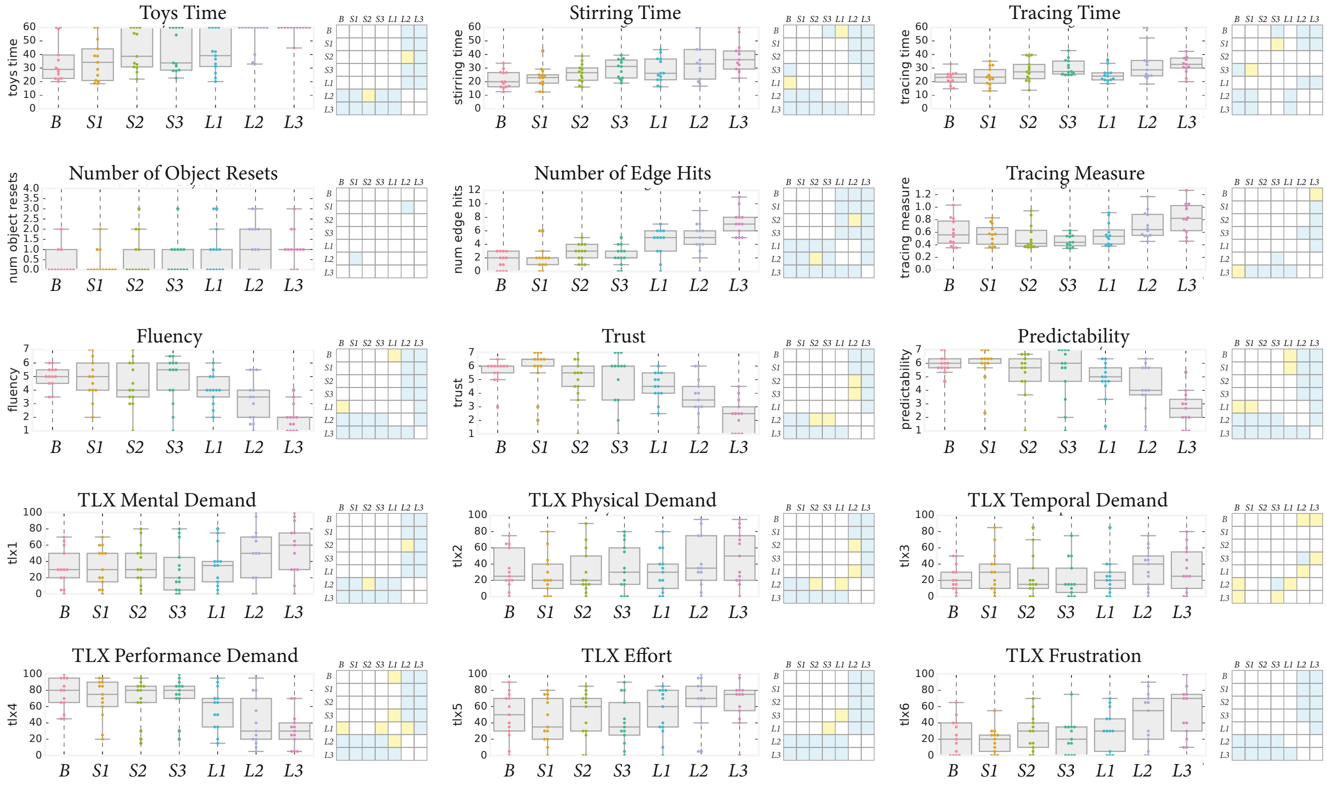

Figure 3: Tukey boxplots of data from the performance and user perception measures for each control condition across tasks. *Base* (B), *Speed1* (S1), *Speed2* (S2), *Speed3* (S3), *Latency1* (L1), *Latency2* (L2), and *Latency3* (L3). Range values signify standard error. The grids to the right of each graph denote pairwise significance. A square is filled if the two conditions are significantly different, with blue squares denoting $p < .01$ and yellow squares denoting $p < .05$.

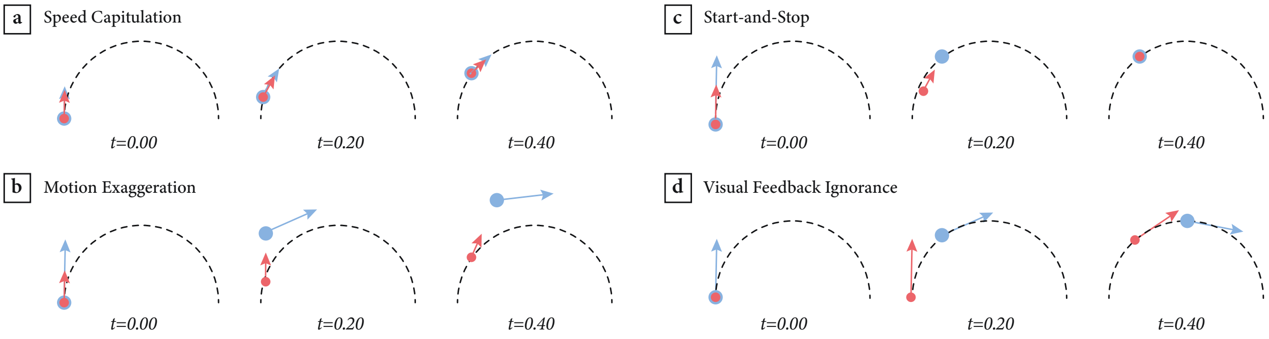

Figure 4: Candidate motor adjustment strategies used to guide our motion data analysis. The "master" (blue dot) illustrates various motion strategies to get the "follower" (red dot) to follow the semi-circle trajectory, even with slowness or onset latency.

5.1 Candidate Motor Adjustment Strategies

To bootstrap our data analysis process, we speculated about possible motion input adaptations that could feasibly counteract the visuomotor dissonance caused by onset latency and robot slowness during mimicry-control. These candidate strategies were speculative and not yet rooted in data, but they served as a starting point for further investigation in the data.

Our prior experience and review of the literature identified four distinct strategies that may be used to overcome robot slowness or onset latency, as seen illustrated in Figure 4: (1) *Speed Capitulation*, that could offset the effects of robot slowness delays by slowing down input motions to match the reduced speed of the robot; (2) *Motion Exaggeration*, that could offset the effects of robot slowness delays by displacing the time needed for the robot to keep up by making larger, yet still fast, motion inputs; (3) *Start-and-Stop*, that could offset the effects of *either* robot slowness or onset latency by moving a little, waiting for the robot to catch up, then moving again. This strategy was observed to occur in the case of onset latency by Sheridan and Ferrell [25]; and (4) *Visual Feedback Ignorance*, that could offset the effects of onset latency by simply doing an open loop motion strategy without adjusting based on where the robot currently is. For example, in the stirring task, a visual feedback ignorance strategy would involve the user blindly tracing a circle in space in front of them without regard for the robot's actions.

5.2 Motion Analysis Framework

After proposing candidate motor adjustment strategies, we next devised two analysis tools that could shed light on the data and indicate whether these motor strategies occurred or how often each strategy occurred. These analysis tools were also designed to be general enough to uncover alternative motor adjustment strategies that we did not anticipate, if such strategies existed.

We designed our first analysis tool, which we will refer to as the *velocity ratio test* (VRT), to search for speed capitulation, motion exaggeration, start-and-stop, or other related motor-adjustment strategies in the data. While these strategies are disparate in many ways, they all share a similarity in terms of exhibiting a unique relationship between the master and follower's *translation velocities* over time. For example, in the case of mimicry-control, a speed capitulation strategy would result in the user's hand and the robot's end-effector velocities being the same over time, a motion exaggeration strategy would exhibit user hand-translation velocities that are consistently larger than the robot's end-effector translation velocities, and a start-and-stop strategy would show the user's hand velocity consistently oscillating between being faster and slower than the robot's end-effector velocity.

Using this observation about the similarity amongst various candidate motor adjustment strategies, the translation velocity ratio test involves four steps. In Step 1, we take the user's hand translation trajectory, $\mathbf{p}_h(t)$, and the robot's end-effector translation trajectory, $\mathbf{p}_r(t)$ over any time frame (e.g., we used full task trials in our user study) and map these translation trajectories to *velocity* signals, $\mathbf{v}_h(t)$ and $\mathbf{v}_r(t)$ respectively. In our work, we approximated velocities using finite differencing. Step 2 involves, at each time step, taking the *ratio* of the user's hand translation velocity and the robot's end-effector translation velocity to create an intermediate *velocity ratio* signal: $\mathbf{r}(t) = \frac{\mathbf{v}_h(t)}{\mathbf{v}_r(t)}, \forall t$. In Step 3, we map the velocity ratio signal $\mathbf{r}(t)$ to a single 3-dimensional point, such that the x-component is the ratio of how many points in $\mathbf{r}(t)$ are above 1.3, the y-component is the ratio of how many points in $\mathbf{r}(t)$ are between 0.7–1.3, and the z-component is the ratio of how many points in $\mathbf{r}(t)$ are below 0.7. We will refer to this point as a *velocity ratio point*. The final step involves statistically analyzing a group of velocity ratio points over many participants, tasks, and types of delay to look for differences.

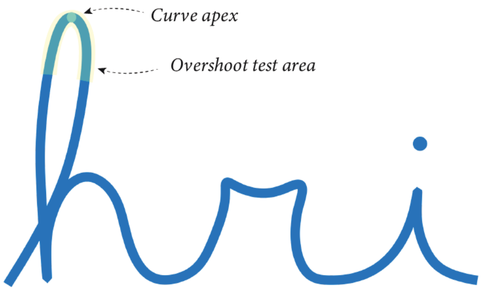

Figure 5: Illustrates where on the cursive "hri" writing the overshoot test was focused.

Note that the mapping procedure described above to create a velocity ratio point takes two entire translation trajectories over any period of time and distills these trajectories down to a three-dimensional space that fully characterizes the speed capitulation, motion exaggeration, start-and-stop, and many other strategy classes that we did not speculate about. To illustrate, the speed capitulation strategy would create a velocity ratio point tending toward $[0, 1, 0]^T$, the motion exaggeration strategy would create a velocity ratio point tending toward $[1, 0, 0]^T$, and the start-and-stop strategy would create a velocity ratio point approximately close to $[0.5, 0.1, 0.4]$. We describe the results of this test on our data in the next section.

Our second analysis tool, which we will refer to as the *overshoot test*, was designed to search for the visual feedback ignorance strategy. Our reasoning here was that overshooting past a goal position when onset latency is present is a strong indication that visual feedback is being used. This overshoot effect occurs because the user is reacting to where they *currently see* the robot, thus causing a late response by the robot (hence, an overshoot) when an onset delay is present. Thus, if participants tended to ignore visual feedback in the onset latency conditions, we would *not* expect there to be any correlation between overshooting past a goal position and the amount of onset latency since the user would be *ignoring* the visual feedback that generally causes overshoot.

The overshoot test involved analyzing the data from the *tracing* task, specifically inspecting the loop at the top of the cursive letter "h" to observe how participants traversed this letter around the apex of the letter. These areas along the curve can be seen in Figure 5. The test here looked at the top point in the writing plane and compared it to the known height of the curve apex point. The higher the user's end-effector went in this region, the greater the measured overshoot. We compare this overshoot measure across all conditions and discuss our results in the next section.

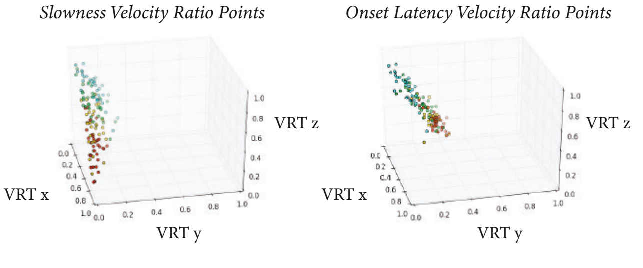

Figure 6: Three-dimensional mapping of the velocity ratio points for slowness (left) and onset latency (right) conditions across all tasks. Blue dots signify the *base* condition; green dots signify *speed1* or *latency1*; yellow dots signify *speed2* or *latency2*; and red dots signify *speed3* or *latency3*. §5.2 discusses the axes for the velocity ratio test (VRT).

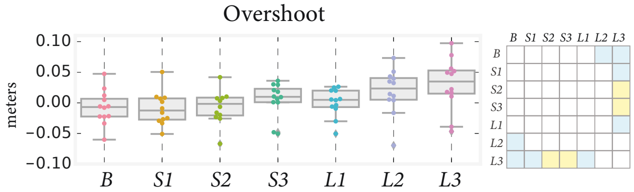

Figure 7: Tukey boxplots of data from the overshoot test. Conditions are *Base* (B), *Speed1* (S1), *Speed2* (S2), *Speed3* (S3), *Latency1* (L1), *Latency2* (L2), and *Latency3* (L3). Range values signify standard error. The grid on the right denotes pairwise significance. A square is filled if the two conditions are significantly different, with blue squares denoting $p < .01$ and yellow squares denoting $p < .05$. Units are in meters.

5.3 Motion Analysis Results

Results from our motion analyses procedures can be seen summarized in Figures 6 and 7. We analyzed the velocity ratio points among all tasks and participants and made comparisons using one-way repeated measures multivariate analysis of variance (MANOVA). There was a significant difference between motor adjustment strategies between robot slowness and onset latency conditions, $F(2, 73) = 11.86$, $p < .0001$, Wilks' $\Lambda = 0.17$. We also found a significant difference between different levels of slowness, $F(2, 70) = 9.04$, $p < .0001$, Wilks' $\Lambda = 0.14$, and onset latency, $F(2, 69) = 8.47$, $p < .0001$, Wilks' $\Lambda = 0.11$. At a high level, participants generally all exhibited a speed capitulation strategy in the base condition. As the robot exhibited more slowness, participants adjusted their movements and adjusted to a motor exaggeration strategy, as seen by points all funneling toward the $[1, 0, 0]^T$ corner of the velocity ratio space. As more onset latency was added, participants adopted more of a start-and-stop strategy, as seen by points tending toward the point $[0.5, 0.1, 0.4]^T$. This finding supports the observation made by Sheridan and Ferrell [25] that people tend to exhibit start-and-stop strategies when facing onset latency.

We analyzed the results from the overshoot analysis using a one-way repeated-measures analyses of variance (ANOVA). Pairwise comparisons used Bonferroni Correction by multiplying the p-value generated from Student's t-test by six. We observed a significant difference in the amount of overshoot between *base* and *latency3* and between *speed1* and *latency3*. We also see a general trend that the more onset latency is added, the more overshoot is present. This observation suggests that participants generally did not exhibit a visual feedback ignorance strategy as onset latency increased, which supports our finding from the previous test that, in this case, people instead favor the start-and-stop strategy.

6 GENERAL DISCUSSION

In this work, we characterized task performance, user perceptions, and motor-control strategies employed in mimicry-control teleoperation when the robot is incapable of matching the user's motions due to *delays* in the system. Specifically, we investigated *two* potential sources of delay: (1) *onset-latency* delays that cause a temporal gap between the user's motion and the onset of the robot's motion; and (2) delays caused by the *robot's speed* being too slow to keep up with the speed of the user's arm motions. We conducted a user study that investigated the effects of multiple levels of onset latency and robot slowness on mimicry-control and characterized the outcome differences between these two sources of delay. We showed that even novice users adeptly found alternate motor strategies to control a robot arm considerably slower than their own arm, but no strategy emerged to sufficiently overcome significant onset latency.

Limitations— Our work has numerous limitations that suggest future extensions. First, our exploration has been limited to mimicry type systems and a specific implementation. Understanding how strategies translate to other systems would help us generalize the results. We also did not examine lower latency conditions as they are not practical with our system. It is possible that even our base condition already represents a significant adaptation strategy. Also, our analysis in §5 only proposed two metrics to characterize motor strategy adjustments. While our analysis provides fruitful observations and clear patterns of strategy modulation, our work only shows that these are *instances* of strategy adjustments during mimicry-control, and we cannot claim that these comprehensively cover all motor strategy adjustments. In future work, we will continue to explore this space through additional evaluations and analysis procedures. Furthermore, our work also only considers strategies to accommodate delays, and we do not consider other factors that may degrade performance such as complex mappings or poor viewpoints. We hope to explore the effects of these other factors, and in particular, understand their relationship to delays.

While our work helps characterize the effects of delays, we have not yet applied these insights to create improved strategies for mitigating the effects. While improving the performance of systems to reduce delays is almost always desirable, practical demands will usually mean that some degree of delay is inherent in any real system. Strategies to mitigate the effects of these delays could be informed by the adaptation strategies that people are using.

Implications—We believe that our work can influence future teleoperation and shared-control systems that utilize motion retargeting. An understanding of onset latency effects using mimicry-control could inform when it is an appropriate or effective interface to use, especially for performance critical applications. For example, mimicry-control may be insufficient to directly control a robot arm on the space station from ground control given the significant time delays involved in sending signals at great distance. In such a scenario, a supervisory control interface where, for example, a user could demonstrate a full motion trace for the robot ahead of time using motion capture, then review, edit, and confirm what the robot's resulting output actions would be from those input motions, might be more desirable. Furthermore, our work could inform future interface design decisions for mimicry-control. For instance, augmented reality or predictive displays could overlay the user's arm over a visualization of the robot arm such that the user could see the effects of time delays on the robot's motion and adapt their input motions accordingly given observed discrepancies.

7 ACKNOWLEDGMENTS

The authors would like to thank Luis Molina for helping run the evaluation. This research was supported by a Microsoft Research PhD Fellowship, National Science Foundation award 1830242, and NASA Cooperative Agreement 80NSSC19M0124.

REFERENCES

[1] Mehran Anvari, Tim Broderick, Harvey Stein, Trevor Chapman, Moji Ghodoussi, Daniel W Birch, Craig Mckinley, Patrick Trudeau, Sanjeev Dutta, and Charles H Goldsmith. 2005. The impact of latency on surgical precision and task completion during robotic-assisted remote telepresence surgery. *Computer Aided Surgery* 10, 2 (2005), 93–99.

[2] Antal K Bejczy, Won S Kim, and Steven C Venema. 1990. The phantom robot: predictive displays for teleoperation with time delay. In *Proceedings., IEEE International Conference on Robotics and Automation*. IEEE, 546–551.

[3] Janez Funda, Thomas S Lindsay, and Richard P Paul. 1992. Teleprogramming: Toward delay-invariant remote manipulation. *Presence: Teleoperators & Virtual Environments* 1, 1 (1992), 29–44.

[4] Raymond C Goertz, Robert A Olsen, and William M Thompson. 1958. Electronic master slave manipulator. US Patent 2,846,084.

[5] Sandra G Hart and Lowell E Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. In *Advances in psychology*. Vol. 52. Elsevier, 139–183.

[6] Richard Held and Nathaniel Durlach. 1991. Telepresence, time delay and adaptation. *Pictorial communication in virtual and real environments* (1991), 232–246.

[7] Guy Hoffman. 2013. Evaluating fluency in human-robot collaboration. In *International conference on human-robot interaction (HRI), workshop on human robot collaboration*, Vol. 381. 1–8.

[8] Peter F Hokayem and Mark W Spong. 2006. Bilateral teleoperation: An historical survey. *Automatica* 42, 12 (2006), 2035–2057.

[9] Dimitrios Hristu, Dimitrios A Kontarinis, and Robert D Howe. 1996. A comparison of delay and bandwidth limitations in teleoperation. *IFAC Proceedings Volumes* 29, 1 (1996), 5709–5714.

[10] Anthony R Lanfranco, Andres E Castellanos, Jaydev P Desai, and William C Meyers. 2004. Robotic surgery: a current perspective. *Annals of surgery* 239, 1 (2004), 14–21.

[11] Dongjun Lee and Mark W Spong. 2006. Passive bilateral teleoperation with constant time delay. *IEEE transactions on robotics* 22, 2 (2006), 269–281.

[12] Mitchell J.H. Lum. 2008. *Quantitative performance assessment of surgical robot systems: TeleRobotic FLS*. Ph.D. Dissertation. University of Washington.

[13] Günter Niemeyer, Carsten Preusche, Stefano Stramigioli, and Dongjun Lee. 2016. Telerobotics. In *Springer handbook of robotics*. Springer, 1085–1108.

[14] Günter Niemeyer and Jean-Jacques E Slotine. 2004. Telemanipulation with time delays. *The International Journal of Robotics Research* 23, 9 (2004), 873–890.

[15] Nancy S Pollard, Jessica K Hodgins, Marcia J Riley, and Christopher G Atkeson. 2002. Adapting human motion for the control of a humanoid robot. In *IEEE International Conference on Robotics and Automation, 2002. Proceedings. ICRA'02.*, Vol. 2. IEEE, 1390–1397.

[16] Oliver Porges, Mathilde Connan, Bernd Henze, Andrea Gigli, Claudio Castellini, and Máximo A Roa. 2019. A wearable, ultralight interface for bimanual teleoperation of a compliant, whole-body-controlled humanoid robot. In *Proceedings of ICRA-International Conference on Robotics and Automation*.

[17] Daniel Rakita, Bilge Mutlu, and Michael Gleicher. 2017. A Motion Retargeting Method for Effective Mimicry-based Teleoperation of Robot Arms. In *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*. ACM, 361–370.

[18] Daniel Rakita, Bilge Mutlu, and Michael Gleicher. 2018. RelaxedIK: Real-time Synthesis of Accurate and Feasible Robot Arm Motion. In *Proceedings of Robotics: Science and Systems*. Pittsburgh, Pennsylvania. https://doi.org/10.15607/RSS.2018.XIV.043

[19] Daniel Rakita, Bilge Mutlu, Michael Gleicher, and Laura M Hiatt. 2018. Shared dynamic curves: A shared-control telemanipulation method for motor task training. In *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*. ACM, 23–31.

[20] Daniel Rakita, Bilge Mutlu, Michael Gleicher, and Laura M Hiatt. 2019. Shared control–based bimanual robot manipulation. *Science Robotics* 4, 30 (2019), eaaw0955.

[21] F. Richter, Y. Zhang, Y. Zhi, R. K. Orosco, and M. C. Yip. 2019. Augmented Reality Predictive Displays to Help Mitigate the Effects of Delayed Telesurgery. In *2019 International Conference on Robotics and Automation (ICRA)*. 444–450. https://doi.org/10.1109/ICRA.2019.8794051

[22] Emily-Jane Rolley-Parnell, Dimitrios Kanoulas, Arturo Laurenzi, Brian Delhaisse, Leonel Rozo, Darwin G Caldwell, and Nikos G Tsagarakis. 2018. Bi-Manual Articulated Robot Teleoperation using an External RGB-D Range Sensor. In *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*. IEEE, 298–304.

[23] Matteo Rubagotti, Tasbolat Taunyazov, Bukeikhan Omarali, and Almas Shintemirov. 2019. Semi-Autonomous Robot Teleoperation With Obstacle Avoidance via Model Predictive Control. *IEEE Robotics and Automation Letters* 4, 3 (2019), 2746–2753.

[24] Thomas B Sheridan. 1993. Space teleoperation through time delay: Review and prognosis. *IEEE Transactions on robotics and Automation* 9, 5 (1993), 592–606.

[25] Thomas B Sheridan and William R Ferrell. 1963. Remote manipulative control with transmission delay. *IEEE Transactions on Human Factors in Electronics* 1 (1963), 25–29.

[26] Wael Suleiman, Eiichi Yoshida, Fumio Kanehiro, Jean-Paul Laumond, and André Monin. 2008. On human motion imitation by humanoid robot. In *International Conference on Robotics and Automation, 2008. ICRA 2008. IEEE*. IEEE, 2697–2704.

[27] Hermann von Helmholtz. 2013. *Treatise on physiological optics*. Vol. 3. Courier Corporation.