UNITE: A Modular Testbed for Reproducible Delayed Space Teleoperation

Dries Cardinaels[1] [0009−0002−1255−0838] ✉, Thomas Pietrzak[2] [0000−0002−2013−7253], Raf Ramakers[1] [0000−0001−6466−0663], and Kris Luyten[1] [0000−0002−4194−1101]

[1] Digital Future Lab, UHasselt - Flanders Make, Diepenbeek, Belgium
{dries.cardinaels,raf.ramakers,kris.luyten}@uhasselt.be
[2] Univ. Lille, Inria, CNRS, Centrale Lille, UMR 9189 CRIStAL, Lille, France
thomas.pietrzak@univ-lille.fr

Abstract. Future lunar and planetary missions are expected to rely on remotely operated surface robots, making communication delay unavoidable. Because this delay degrades teleoperation, researchers develop and evaluate interfaces and control techniques for working under delayed feedback. These evaluations often use custom setups that are difficult to reconstruct, compare, and reuse. To identify the parts a reusable setup must capture, we conducted a scoping review of 83 studies of delayed human-in-the-loop mobile teleoperation. The review identified seven requirements, covering communication, input, feedback, vehicle behavior, assistance, task and environment, and data capture. We used these requirements to build UNITE, an openly available modular Unity simulation testbed for human-in-the-loop space teleoperation that captures the apparatus as a shareable configuration. We demonstrate UNITE by reconstructing a prior lunar-scale delay study. In the reconstruction, UNITE represented it without extension, while making three apparatus choices explicit that the original study had left underspecified. UNITE is available at `https://github.com/UHasselt-DigitalFutureLab/unite-teleoperation`.

Keywords: Space teleoperation · communication delay · human-in-the-loop teleoperation · mobile robot simulation · experimental testbeds · reproducible research

1 Introduction

Future space exploration is expected to rely on robots for surface work between and alongside crewed missions, including inspection, transport, maintenance, construction, and science operations [43,36]. Depending on the task and operating context, a human operator may directly control the robot, share control with autonomous functions, or supervise actions performed with limited intervention [59,20]. Even as rovers gain greater onboard autonomy in areas such as navigation and fault response, human teams remain responsible for defining goals and constraints, assessing returned information, resolving failures, and preparing subsequent plans [21,54]. When this involvement occurs remotely, commands and sensor feedback are subject to round-trip communication delays ranging from seconds for Earth–Moon operations to minutes for Earth–Mars operations [37,57]. Because this delay results from physical distance rather than limitations in communication technology, it cannot be eliminated and remains a central challenge for effective remote robotic operation.

Because this delay cannot be removed, research aims not to eliminate it but to help operators stay in control [39,30]. Delay has long been shown to degrade remote operation of a mobile robot: it slows operators down, increases error, and leads them to adopt a cautious move-and-wait strategy [13,18]. In response, previous research developed ways to compensate for delay. A long-standing approach is the predictive display, which shows the operator a prediction of the robot's near-future state so they can act on where the robot will be rather than where it was [4,45]. Others share control with onboard autonomy or augment the operator's view [46,63]. These techniques are studied through human-participant experiments, in which participants operate a robot under a chosen delay while their performance is measured [9]. Running such an experiment requires building an apparatus: a vehicle, an environment, a way of applying the delay, and a way of presenting the task to the operator.

Experimental apparatuses for delayed teleoperation are typically developed for the needs of individual studies. This is understandable, given differences in research questions, tasks, vehicles, environments, and assistance techniques. However, such study-specific development can make it difficult to reuse or directly compare experimental setups across studies. Researchers wishing to build on earlier work may need to reconstruct important aspects of a setup from published descriptions, which may not fully specify details such as delay implementation, vehicle dynamics, camera views, control mappings, training procedures, or metric definitions. Our scoping review of 83 empirical studies of delayed ground-vehicle teleoperation indicates that many of these experimental choices recur across the literature, but are built and described differently from study to study. As a result, reproducing a prior setup, evaluating a method under matched conditions, or extending it with a new assistance technique can require substantial reimplementation and interpretation. Similar concerns have been raised more broadly in human–robot interaction, where custom study designs, inconsistent metrics, and inconsistent reporting have been identified as barriers to replication, comparison, and reuse [72], prompting recent calls for more formalized and standardized reporting frameworks [3].

Reporting standards do not make experimental setups reusable; researchers also need shared technical infrastructure. Gazebo and CARLA provide robotics and driving simulation [32,14], DUST provides a lunar environment [5], and TELECARLA extends CARLA for networked teleoperation [25]. PsychoPy and UXF structure trials and participant-data collection [50,6]. These tools address separate parts of the apparatus but do not provide a common representation of the simulator, communication delay, operator interface, task logic, and logging.

The missing layer is a reusable substrate that captures the complete delayed-teleoperation apparatus as a shareable configuration.

To support more comparable, reproducible, and reusable research on delayed space teleoperation, we make three contributions that together give researchers an evidence-based foundation for specifying, sharing, reproducing, and extending experiments in space teleoperation under communication delay:

- a scoping review of 83 empirical studies of delayed human-in-the-loop mobile teleoperation, characterizing recurring experimental elements and variation that matters for reuse in how those elements are implemented and reported;
- UNITE, an openly available modular testbed that represents communication conditions, operator input, feedback, vehicle behavior, assistance techniques, task and environment, and data capture as configurable components of the delayed-teleoperation apparatus; and
- a demonstration of UNITE through the reconstruction of an experimental setup grounded in a prior human-participant study conducted under lunar-scale delay.

2 Related Work

UNITE lies at the intersection of three topics: interaction techniques for operating robots under communication delay, operational arrangements for human involvement in space robotics, and infrastructure for constructing and reproducing human-participant experiments.

2.1 Telerobotics and Communication Delay

Communication delay decouples operator action from observed robot response: commands take effect later, while feedback represents an earlier system state [18,59]. Its effects depend not only on magnitude, but also on task, vehicle speed and control behavior, and whether communication delay is fixed or variable [66,28,30]. As delay increases, continuous control degrades and operators may adopt move-and-wait control [18,30].

Predictive displays address delay by presenting an estimated future state of the remote system. Early phantom-robot displays visualized a predicted robot pose [4]; later implementations evaluated predictive vehicle overlays for lunar-rover driving [39], low-cost predictive displays [15], transformed delayed video showing an estimated future view [45], and predictive corridors [22]. Their effectiveness depends on what is externalized: under a fixed 2.56-second round-trip delay, trajectory visualization shortened task time and reduced perceived cognitive load, whereas command-timing visualization had no measurable effect [9].

A second strategy reallocates control between operator and robot. Supervisory and collaborative control redistribute authority between operator and robot: humans provide goals, commands, or advice, while autonomous functions execute or safeguard parts of the task [59,20]. Local autonomy can improve teleoperation productivity [33]; adjustable-autonomy interfaces support direct, skill-level, and task-level control [58]; and shared control blends operator and automated commands for obstacle avoidance under delay [63]. Model-based approaches instead compensate for delayed execution by predicting or tracking reference vehicle states [53,52]. These approaches occupy different points in the teleoperation loop: prediction acts on operator-side feedback, command assistance may act before or after transmission, and local autonomy executes onboard.

Their evaluation also depends on the surrounding apparatus. Delay may affect command or feedback paths, remain fixed or vary over time, and be allocated differently between uplink and downlink [66,30,53]. Vehicle dynamics, command mappings, viewpoints, environment geometry, and metric definitions introduce further variation. Meaningful comparison therefore requires both the interaction technique and its experimental conditions to be specified.

2.2 Operational Contexts for Human-in-the-loop Space Teleoperation

Operator location determines how responsibilities are divided between humans and robots. Earth-based Mars rover teams use downlinked data, visualization and planning tools, and distributed operations to plan rover activities [47,67]. This process requires substantial operational effort and constrains mission productivity [21]. During drives, real-time intervention is impossible, so Curiosity relies on onboard navigation, visual odometry, and fault protection [54].

Nearby orbital control permits a different allocation. HERRO and lunar Gateway concepts place astronauts in orbit to operate surface robots with low communication delay without requiring crew to land at the worksite [57,8]. NASA's Human Exploration Telerobotics project and subsequent ISS tests evaluated communications, control modes, and operations for crew-controlled surface robots [19,7]. ESA's ANALOG-1 similarly evaluated an ISS-based interface for controlling a mobile robot and coordinating an astronaut with ground scientists [68].

Mission demonstrations provide operationally grounded evidence, but rely on mission-specific hardware, procedures, and operational contexts [19,7,68]. Terrestrial analogs instead test technologies, communication conditions, and operational concepts under selected mission constraints [55,16], while simulation permits tighter control of the apparatus. Litaker et al. [37], for example, evaluated a simulated Lunar Terrain Vehicle under 0-, 4-, 6-, and 8-second delays and identified workload, driving strategy, workstation ergonomics, and training as factors in remote-driving performance. Recent SpaceCHI position papers further propose human-in-the-loop AI, digital twins, and edge AI [41], and dynamically varying autonomy and interface immersion [29]. These proposals broaden the conditions an experimental system must represent, but do not provide a shared basis for comparing them under controlled conditions.

2.3 Experimental Infrastructure and Reproducibility

Robotics simulators provide important parts of a teleoperation apparatus. Gazebo supports physics-based robot and environment simulation [32], while CARLA provides vehicles, configurable sensors and environmental conditions, and controlled driving scenarios [14]. TELECARLA extends CARLA for teleoperated driving and compressed-video stream configuration [25], and DUST models lunar south-pole terrain and lighting for mission analysis [5]. Simulator capabilities and intended uses differ substantially [10]; these systems therefore provide apparatus components, but do not themselves define a reusable delayed-teleoperation experiment.

Behavioral-experiment software addresses a complementary layer. PsychoPy and OpenSesame support graphical or scripted experiment construction and stimulus–response handling [50,40], while jsPsych defines and executes browser-based task sequences [34]. Within Unity, UXF represents studies as sessions, blocks, and trials and supports behavioral and continuous-data capture [6]. These tools organize study execution, but do not specify the teleoperation pipeline connecting operator input, communication delay, vehicle behavior, assistance, feedback, and task state. UNITE complements them by representing that apparatus within each experimental condition.

Reproducibility requires both explicit reporting and executable configurations. Steinfeld et al. [62] call for common HRI metrics and attention to biasing factors; Zimmerman et al. [72] show that decentralized methods and custom measures limit baseline comparison; and Bagchi et al. [3] propose formalized reporting for study details often omitted from publications. These works clarify what should be reported, but do not provide a technical system for instantiating and sharing the reported apparatus.

Adjacent benchmarks address narrower infrastructure problems. M4Bench supports comparison across users, robots, goals, and input devices in simulated manipulation [69], while the Space Robotics Bench provides an open-source simulation benchmark for autonomous robot learning in space environments [48]. Neither targets delayed, human-operated mobile robots. The remaining gap is an executable layer that makes communication conditions, input mapping, feedback, vehicle behavior, assistance, task and environment, and synchronized logging independently configurable and shareable. This is the layer implemented by UNITE and derived from the scoping review that follows.

3 Scoping Review: Characterizing Experimental Design

To base UNITE's modules on what actually varies between studies rather than on our own assumptions, we conducted a scoping review of how empirical delayed human-in-the-loop (HITL) mobile-teleoperation studies are built. The review characterized the design space of these experiments and identified the recurring components that UNITE represents as configurable modules.

3.1 Objective and Research Question

We looked at how delayed HITL mobile-teleoperation studies describe the conditions under which a technique is tested. Our goal was to find out what a study must report for others to do three things: rebuild the original setup, test a new technique against an existing one under the same conditions, and change one factor, such as communication delay or environment, without unintentionally changing anything else. We therefore asked:

RQ: Which experimental elements recur across delayed HITL mobile-teleoperation studies, and how does variation in how those elements are realized affect the reproduction, comparison, and reuse of teleoperation techniques?

3.2 Review Methodology

We searched ACM Digital Library (n=30), IEEE Xplore (n=371), Scopus (n=341), and Web of Science (n=176) for all records indexed up to and including 2025, resulting in 918 records before deduplication. Additional eligible studies were identified through other search methods, including targeted manual searching, and were recorded separately from the database-search output. We restricted the upper bound to 2025 so that the search captured only complete publication years. Duplicates were removed in two stages: automatic deduplication with ASySD [23] removed 318 records, and a subsequent manual review removed a further 9 that the automatic stage had missed, leaving 591 unique records for screening.

Inclusion Criteria. We defined eligibility as five inclusion criteria. A record was eligible if it (IC1) involved teleoperation, remote operation, or supervisory control by a human operator; (IC2) used a ground or surface mobile platform such as a rover, unmanned ground vehicle, or comparable embodied system; (IC3) reported an empirical study in which real human operators performed the task, shown by a described procedure, a description of the participants, participant-level results, or a reported number of operators; (IC4) treated communication delay as an experimental condition rather than as motivation or as an incidentally measured network property; and (IC5) was a peer-reviewed publication available in English. We excluded work that addressed only autonomous learning, that evaluated control or stability methods without a participant study, or that mentioned delay merely as motivation without reporting an experimental setup.

Screening. As illustrated in Fig. 1, all 591 records were screened against the eligibility criteria; records that could not be judged from the title and abstract were retained for full-text review [35]. This stage excluded 399 records, leaving 192 reports sought for retrieval. Full texts could not be retrieved for 21 reports, which were excluded. The remaining 171 reports were assessed in full text, of which 79 were excluded. Fig. 1 gives the number excluded under each criterion.

This left 92 studies eligible from the database search. We then used other search methods, including manual search, to identify 6 additional eligible records, giving 98 studies for full-text data charting.

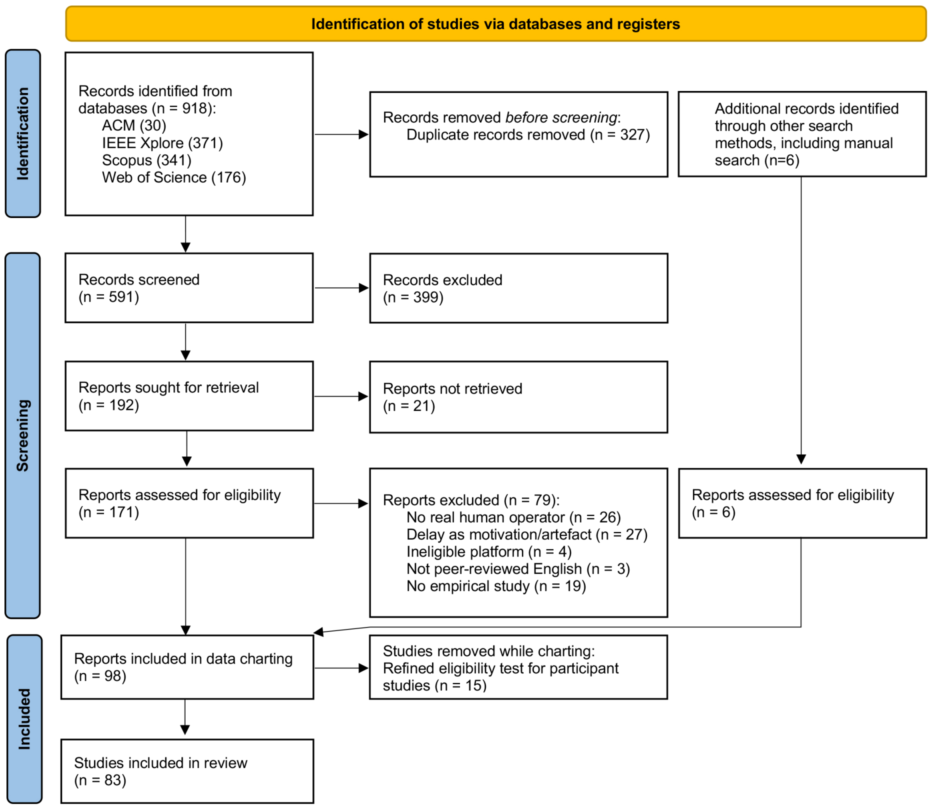

Fig. 1. PRISMA 2020 flow diagram [49] detailing the database searches, deduplication, screening, and final inclusion of studies.

While charting the full texts, we found that IC3 needed a clearer test. Some papers had a real human operating the system, but only to demonstrate a control or interface method, with no study of the operator. We therefore required each included study to show at least one clear sign of a user study: a described procedure, a description of the participants, participant-level results, or a reported number of operators. Applying this test to the 98 charted reports, we found 15 that were demonstrations of a control, stability, or interface method: a human closed the control loop, but no user study was reported, and the reported results described the method rather than the operator. We removed these 15 reports, so the final corpus comprised 83 studies for data charting.

Data Charting. We charted each study at the level of its explicitly reported experimental choices. Studies containing multiple experiments were charted separately. Charting proceeded in two passes. First, one reviewer conducted an open discovery pass over 10 studies randomly selected from the 83 included studies. Within a fixed set of components, we recorded the properties reported by each study using the study's own terminology and retained supporting quotations. These components correspond to stages of the human-in-the-loop teleoperation loop [59], extended with the technique evaluated and the outcomes measured.

Semantically equivalent properties were consolidated into a provisional codebook. Fields were retained when they described an experimental choice relevant to reconstructing a study, comparing studies, or understanding how a technique was implemented or evaluated. The same reviewer then re-charted the 10 pilot studies using this codebook to validate its definitions and coverage. In the second pass, the codebook was applied to all 83 studies. Reported values were extracted and normalized into separate fields together with the supporting quotation. Information absent from a study was coded as not reported. Cases that could not be mapped reliably to a codebook field were provisionally marked as unclear and subsequently revisited using the field definition and relevant evidence. All such cases were resolved as either reported or not reported.

Synthesis and Requirement Derivation. We compared the charted data across studies using the components of the teleoperation loop. We identified recurring elements, variations that affect reconstruction, and reporting gaps. We retained only information explicitly reported by each source at the required level of specificity; missing mappings, viewpoints, communication directions, vehicle geometries, and feedback modalities were not derived from hardware descriptions or contextual assumptions. Recurring elements and variations informed the scope and configuration options of seven general requirements for UNITE. These requirements were subsequently used to define UNITE's modules and their configurable interfaces.

4 Design Requirements

From the synthesis of the charted data, we derived seven requirements covering communication conditions, operator input, operator feedback, vehicle behavior, assistance techniques, task and environment, and data capture. The studies cited below illustrate the variation underlying each requirement; they are examples from the corpus rather than its sole basis. The charting also recorded study-protocol properties, such as participant numbers, training, and experimental design, together with subjective measures. These describe how a study is run and analyzed rather than the apparatus it runs on, and are supported by dedicated experiment-management software. UNITE therefore represents the conditions under which a study is run, not the study protocol. UNITE must allow the elements covered by the requirements to be configured independently while saving the fully specified setup as a shareable Unity project. This enables researchers to reproduce an existing setup and extend or adjust it by modifying only the elements relevant to a new study.

DR1: Configurable communication conditions. Delay was applied to control commands [64], video feedback [39], and vehicle telemetry [71]. Delay magnitude and directional allocation also varied. Maag et al. [38] compared 200/0, 0/200, and 100/100 ms uplink/downlink conditions, demonstrating that the same total delay can be distributed differently across the teleoperation loop. Delay could be fixed [64], follow predefined time-varying functions [61], or be sampled from distributions [64]; some studies additionally measured or manipulated other communication impairments, such as packet loss [65]. Reporting conventions also differed, with delay expressed either as separate directional components [38,63] or as a round-trip communication time [42]. A reported delay magnitude alone is therefore insufficient to reconstruct the implemented condition. UNITE must represent each communication condition through explicit configuration of the affected stream, communication direction, delay magnitude, temporal behavior, and any additional implemented impairment, such as packet loss. Separate uplink and downlink configurations must preserve the directional allocation of delay. UNITE must generate these conditions within Unity to avoid dependence on external network infrastructure.

DR2: Operator input and command mapping. Studies varied in the physical input device, the control modality, and the mapping from device signals to vehicle or subsystem commands. Matheson et al. [39] explicitly mapped the left analog stick of an Xbox controller to longitudinal translation and the right stick to yaw. Storms et al. [63] used condition-specific mappings from a controller trigger and joysticks to forward speed, turn rate, and waypoint position. Abubakar et al. [1] reported velocity-to-velocity coordination between a haptic device and rover linear and angular velocities. Other studies identified an input device without reporting a complete mapping. Dybvik et al. [15], for example, reported keyboard arrow-key control but not how presses translated into motion commands, such as the commanded speed or whether input was held or toggled. A device name or general command category is therefore insufficient to reconstruct control behavior. UNITE must support independently configurable input devices and control modalities together with explicit device-to-command mappings, including the relevant keys, axes, buttons, scaling, and command variables.

DR3: Operator feedback channels and viewpoint. Operator feedback varied in modality, delivery device, displayed information, and camera viewpoint. Visual feedback included reconstructed 3D views [31], communication delay indicators [56], and telemetry such as current speed [38]. Viewpoints included two displayed camera feeds, one from an operator-steerable top camera [42], additional mirror views [38], and explicitly manipulated elevated and normal views [27]. Haptic and auditory channels were also manipulated experimentally: studies compared force-feedback conditions under delay [60], haptic steering assistance on and off [26], and visual with visual-plus-auditory feedback across delay levels [17]. In bilateral systems, force feedback could itself be delayed [51]. UNITE must therefore support explicitly configured visual, haptic, and auditory output channels, their delivery devices, displayed information, interface layout, and selectable camera viewpoints. Camera placement and optical parameters such as field of view must be stored when configured, and each feedback stream must be independently connectable to the communication configuration in DR1.

DR4: Configurable vehicle behavior. The same operator command can produce different motion depending on the platform, steering geometry, dynamic model, actuator response, and wheel–terrain interaction. Some papers specified these elements in detail: a simulated large UGV with a 14-DoF model, Pacejka tires, a diesel engine, and an automatic transmission [44], soft-terrain slippage modeled as part of the vehicle [1], and a deliberate distinction between the controller-internal model and the simulated plant [63]. Other papers described the apparatus without specifying its motion behavior: a three-wheel robot with two motor-driven wheels and a caster, without explicitly naming the steering mode or specifying how commands produced turning [15], or game physics characterized as realistic without a model type or parameters [45]. In both cases the vehicle behavior cannot be reconstructed from the description. UNITE must support configurable vehicle or robot models with parameterized dynamic behavior, including actuator response and wheel–terrain interaction where applicable, and must treat steering geometry as an explicit configuration field rather than a property implied by vehicle construction.

DR5: Configurable assistance techniques. Assistance techniques differed in their function, attachment point in the teleoperation loop, and required inputs. Predictive displays transformed or overlaid visual feedback [39,22]; shared control corrected steering on the operator side before transmission [12]; model predictive control compensated delay and avoided obstacles on the remote side [24]; and predictor frameworks placed coordinated functions on both sides of the loop [1]. Their reported dependencies included command history and LiDAR point clouds [70], as well as vehicle state, route information, and current delay [12]. Placement may also vary by experimental condition: Storms et al. [63] evaluated the same shared-control function located onboard the robot and at the operator control unit as separate interfaces. UNITE must therefore support assistance techniques as configurable modules with a loop attachment point, required inputs, and dependencies.

DR6: Configurable task and environment. Task and environment definitions combined goals, course geometry, terrain, obstacles, and ambient conditions. Cross et al. [11] varied corridor width and lighting as experimental factors;

Storms et al. [63] specified arena geometry, obstacle placement, and a search goal; Mellinkoff et al. [42] distributed target rocks across a lunar-analog crater course and ended the trial on target identification. UNITE must therefore support configurable task goals and explicit completion, failure, and termination rules. Course content is provided as shareable Unity assets, with each condition shared as a self-contained scene. Properties stored as component values, such as lighting intensity, are recoverable from that scene. Properties that emerge only from authored geometry, such as corridor width, are not automatically represented as named experimental factors or levels. Recording those semantics remains a study-reporting requirement rather than a system requirement.

DR7: Data capture and metric definition. A path-following score was calculated by averaging normalized robot scores at each timestep; a score of 1 indicated perfect center-line following and 0 indicated that the robot was off-track for the entire path [64]. Cornering time was computed as exit time minus entry time after excluding curves with minimum speed at or below 0.005 m/s [2]. Time to discovery was measured from operator movement start to successful target identification [42]. For failed tasks, adjusted completion times were assigned using successful times from the same condition [11]. The required data capture depends on the metric definition and requires synchronized logging of the operator commands, vehicle states, and task events used by those definitions. UNITE must therefore record timestamped raw variables and events required for selected objective and system measures and retain each metric's definition, units, formula or calculation procedure, thresholds, aggregation rule, and start and stop events.

5 The UNITE Environment

We implement the teleoperation loop as a staged orchestrator with separate uplink and downlink paths. Uplink denotes information flowing from the operator station to the robot, while downlink denotes information flowing from the robot to the operator station. Figure 2 provides a conceptual overview of the operator station, communication paths, and remote platform in this architecture.

5.1 Input Provider

What. Acquires device-specific operator actions and exposes typed signals. **Why.** Implements DR2 by separating hardware access from command semantics. **How.** Reports relevant inputs, values, states, and timestamps through a declared output contract. **Boundary.** It does not assign command meaning, plan motion, or control the vehicle. Providers are directly interchangeable only when their output contracts match.

5.2 Command Mapping and Encoding

What. Converts typed provider signals into a study-defined command representation. **Why.** Implements DR2, as device signals do not determine their control

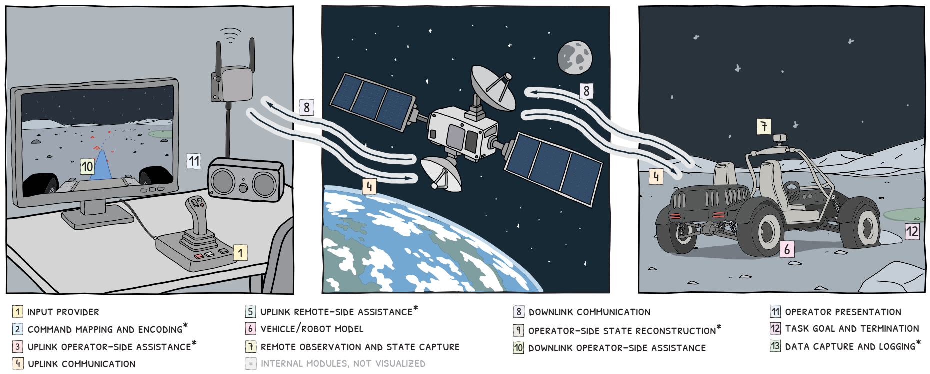

Fig. 2. Conceptual overview of the UNITE telerobotic loop. Numbers identify module contracts rather than execution order; asterisks denote internal modules not shown.

meaning. **How.** A selected implementation consumes the output contract of the Input Provider (5.1) and explicitly configures the relevant keys, axes, buttons, input states, scaling, and resulting command variables. It publishes the mapped commands through a declared output contract. **Boundary.** It represents operator intent but does not plan, constrain, transmit, or execute motion. Provider–mapping combinations are compatible only when their declared contracts match.

5.3 Uplink Operator-side Assistance

What. Optionally processes mapped commands before uplink transmission. **Why.** Implements the operator-side attachment point of DR5. **How.** A selected implementation processes packages from Command Mapping and Encoding (5.2) using its declared dependencies. It may suppress each package or publish one with a preserved or transformed payload. **Boundary.** It neither generates command sequences, manages communication, nor executes motion. Without assistance, mapped packages pass unchanged to the Uplink Communication module (5.4).

5.4 Uplink Communication

What. Applies a configurable communication condition to local-to-remote packages. **Why.** Implements the uplink component of DR1. **How.** Consumes packages from the Operator-side Assistance module (5.3), or directly from Command Mapping and Encoding when no assistance is selected. The selected implementation defines delay magnitude, temporal behavior, and supported non-delay degradation. The module schedules and queues the resulting transmissions, releasing all due packages; outcomes may also suppress transmission, such as for packet loss. **Boundary.** It treats payloads as opaque and owns only local-to-remote communication. It neither executes commands nor represents the downlink.

5.5 Uplink Remote-side Assistance

What. Optionally processes received command packages before vehicle control. **Why.** Implements the remote-side uplink attachment point of DR5, including shared control, safeguarded execution, delay compensation, and obstacle avoidance. **How.** A selected implementation processes packages released by the Uplink Communication module (5.4) and may read declared dependencies such as vehicle state, sensor data, or environment models. It may preserve, modify, replace, or suppress a command. **Boundary.** It does not manage communication, own remote state, or generate actuator outputs. Without assistance, packages pass directly to the Vehicle/Robot Model (5.6).

5.6 Vehicle/Robot Model

What. Applies received commands through the selected vehicle behavior and updates the authoritative remote state. **Why.** Implements the configurable steering geometry, dynamic behavior, actuator response, and wheel–terrain interaction required by DR4. **How.** It consumes packages from the Uplink Remote-side Assistance module (5.5), or directly from the Uplink Communication module (5.4) when no assistance is selected. Each implementation declares its accepted command contract and explicitly configures its steering geometry, kinematic or dynamic model, actuator response, and wheel–terrain interaction where applicable. **Boundary.** It does not map operator-device signals, manage communication, or implement assistance techniques. Its authoritative state may be read by remote-side assistance and observation modules.

5.7 Remote Observation and State Capture

What. Coordinates the capture of configured remote observations as typed, stream-labeled packages. **Why.** Implements the configurable feedback sources required by DR3 while distinguishing the robot's local state from the delayed information available to the operator. **How.** Each control step uses two observation refreshes. Before remote-side assistance processes released commands, the module captures the current remote state and makes the resulting observations available locally without downlink delay. After the Vehicle/Robot Model updates, it captures the resulting state again, replaces the local observation, and optionally publishes the post-control package to the downlink. A source may be configured as local-only, or may publish its post-control package to a selected downlink stream. **Boundary.** It defines the capture and publication lifecycle, while each source implementation defines the observed data and sampling behavior. It does not apply communication conditions, reconstruct operator-side state, or present feedback.

5.8 Downlink Communication

What. Applies configurable communication conditions to remote-to-local feedback packages. **Why.** Implements the downlink component of DR1 and the independently configurable feedback-stream connections required by DR3. **How.** The module routes each package to a configured downlink channel according to its declared feedback stream. Each channel applies its selected communication-condition implementation, queues the resulting transmissions, and releases all due packages. Only observations selected for downlink publication enter this module; locally consumed observation streams remain outside the delayed feedback path. **Boundary.** It routes and schedules packages using their metadata without inspecting or modifying the contained feedback data. Each channel implementation defines its delay and supported non-delay degradation.

5.9 Operator-side State Reconstruction

What. Reconstructs operator-side information from feedback packages released by the Downlink Communication module (5.8). **Why.** Provides the delayed remote information required for reconstructed views and other visual feedback under DR3. **How.** Selected reconstruction implementations consume declared feedback-package contracts and update corresponding local data representations. For example, vehicle-state, depth-data, and video-frame packages may update a local pose, reconstructed map, and frame buffer, respectively. **Boundary.** It does not apply communication conditions, process assistance, update operator-visible Unity objects, select interface layouts, or access current authoritative remote state. Each implementation defines how released observations are converted into operator-side information, which is made available to Operator Presentation (5.11) and, where needed, Downlink Operator-side Assistance (5.10).

5.10 Downlink Operator-side Assistance

What. Optionally processes or augments feedback available on the operator side before presentation. **Why.** Implements the operator-side downlink attachment point of DR5, including prediction, state estimation, feedback filtering, and warning generation. **How.** A selected implementation may consume reconstructed operator-side information from the Operator-side State Reconstruction module (5.9) and may read declared local dependencies such as command history, pending uplink commands, vehicle models, route information, configured communication delay, or reconstructed maps. It may publish transformed or augmented information, or one or more supplementary packages; any replacement or suppression of reconstructed feedback is defined by the study-specific presentation implementation. Each output package is a stream-labeled `Package` carrying a study-defined payload. **Boundary.** It does not apply communication conditions, update authoritative remote state, or directly determine the interface layout. Without assistance, reconstructed operator-side information is made directly available to the Operator Presentation module (5.11).

5.11 Operator Presentation

What. Maps reconstructed or assisted operator-side information to configured operator-side output sources. **Why.** Implements the feedback modalities, displayed information, interface layout, output-device connections, and selectable viewpoints required by DR3. **How.** The module receives reconstructed operator-side information from the Operator-side State Reconstruction module (5.9) and combines it with output from the Downlink Operator-side Assistance module (5.10) when selected. A study-specific presentation implementation routes the available operator-side information to configured local sources and updates those sources during subsequent presentation steps. These sources may provide visual, auditory, or haptic output, but their device-specific behavior and layout are defined by the selected study implementation. **Boundary.** It determines how available feedback is delivered to the operator but does not capture remote observations, apply communication conditions, update authoritative remote state, or compute assistance.

5.12 Task Goal and Termination

What. Evaluates configured task criteria and terminates the active trial when a terminal condition is met. **Why.** Implements the completion, failure, and termination rules of DR6. **How.** Selected criterion implementations are evaluated during each fixed-rate control step. When a criterion returns a terminal result, the module accepts the first result, ends the active trial through the staged orchestrator, and makes a timestamped trial-ended event available to other modules. **Boundary.** It defines the criterion interface and termination lifecycle, while each implementation defines the condition that constitutes success, failure, timeout, or abort.

5.13 Data Capture and Logging

What. Records configured raw variables and events produced during each trial. **Why.** Implements the synchronized data-capture and metric-definition requirements of DR7. **How.** Selected capture implementations record timestamped observations from declared module outputs, state variables, and experiment events. The module stores metric definitions, including units, calculation procedures, thresholds, aggregation rules, and start and stop events, for downstream analysis. **Boundary.** It records the experiment without controlling communication, vehicle behavior, assistance, task execution, or termination; downstream analysis converts observations into reported measures.

5.14 Coverage

Table 1 maps each requirement to the modules or configuration assets that implement or represent it. Four requirements (DR1, DR2, DR3, and DR5) span

Table 1. Mapping from design requirements to implementing modules and represented scene content.

DR	Implemented or represented by
DR1	Uplink Communication (5.4), Downlink Communication (5.8)
DR2	Input Provider (5.1), Command Mapping and Encoding (5.2)
DR3	Remote Observation and State Capture (5.7), Downlink Communication (5.8), Operator-side State Reconstruction (5.9), Operator Presentation (5.11)
DR4	Vehicle/Robot Model (5.6)
DR5	Uplink Operator-side Assistance (5.3), Uplink Remote-side Assistance (5.5), Downlink Operator-side Assistance (5.10)
DR6	Task Goal and Termination (5.12), condition scene
DR7	Data Capture and Logging (5.13)

several modules because the review identified separable stages within those parts: delay can be configured independently on the uplink and downlink; input devices are separated from command mapping; feedback is observed, transmitted, reconstructed, and presented as distinct stages; and assistance can attach before uplink transmission, after uplink reception on the remote side, or after downlink reception on the operator side. The Downlink Communication module is shared between DR1 and DR3 because the same channel carries delayed feedback. DR6 is split between the Task Goal and Termination module and the condition scene: the module defines task goals and termination rules, while the scene holds course and environment content. No dedicated module owns that scene content, although vehicle, assistance, observation, and task-criterion implementations may read or act on it.

6 Demonstration

To demonstrate UNITE, we used it to reconstruct an existing study: Cardinaels et al. [9], which compared three feedforward visualizations (Network, Path, and Envelope) against a baseline with delayed video feedback, with 24 participants driving a simulated rover across lunar terrain under a fixed delay. It was not part of our corpus, so the study demonstrating the requirements was independent of those used to derive them. We charted it with the same codebook as in the review and mapped each charted element onto a UNITE module. The apparatus fit the UNITE Core and Kernel without architectural extension. The four conditions were reproduced as separate scenes differing only in the selected downlink operator-side assistance component; the shared Operator Presentation module renders its output. Every other module was configured identically, and all four scenes shared the same vehicle and terrain. Reconstruction also revealed three points where the study's description, though otherwise detailed, does not fully specify the apparatus, each corresponding to a requirement that calls for the missing information: DR1, DR4, and DR5.

Directional allocation (DR1). The study reports a fixed 2.56 s round-trip delay but does not state how it divides between the uplink and downlink. Reconstruction requires a specific split, so we assumed an uplink-only allocation. In our UNITE reconstruction, we consequently configure a 2.56 s uplink delay and a 0 s downlink delay, assigning the full delay to the command path. The resulting rover response and video feedback are therefore delayed by the command-path communication delay, even though the configured downlink delay is zero.

Disturbance parameters (DR4). The study describes its disturbance model in structural detail, naming six sources and identifying which use Perlin noise, but reports no parameter values, describing them as iteratively calibrated. Our UNITE implementation ports the described disturbance mechanisms using the original implementation, available to the author team, as a provenance reference. However, the configuration values used in the reconstruction are explicit choices rather than values reported in the paper. They are intended to produce comparable trajectory variation, not to claim the original calibration.

Route keeping (DR5). The study attributes the constraint that keeps operators on the route to course geometry, stating that cliffs and boundaries prevented off-path driving. It does not specify whether this is enforced by terrain alone or by an assistance mechanism acting on operator commands. Using the original source project available to the author team, we resolved this ambiguity in our UNITE implementation by porting the boundary-detection behavior as a slope-based remote-side assistance component, including its threshold and reverse-on-contact response. A reader with the paper alone could not recover that implementation choice.

None of these gaps required extending the UNITE Core or Kernel; each required an explicit value or implementation choice to run the reconstruction. Even a recent, carefully reported study left three apparatus details underdetermined in the publication. This is exactly the reproduction problem the requirements target, now concrete: specific configuration fields rather than a vague concern.

7 Conclusion

Research on delayed space teleoperation often relies on custom experimental setups, making prior work difficult to reconstruct, compare, and reuse. UNITE addresses this problem with a shared foundation for delayed human-in-the-loop mobile teleoperation. We grounded its design in a scoping review of 83 empirical studies, identifying recurring variation in communication, operator input, feedback, vehicle behavior, assistance, task and environment, and data capture. UNITE translates these aspects into independently configurable components. Reconstructing a previous lunar-scale delay experiment showed that UNITE could represent the setup without extending its architecture, while making explicit three details left open in the original study: the allocation of delay between uplink and downlink, the disturbance parameters, and the route-keeping implementation.

The literature review was charted by a single reviewer, which may introduce subjective judgment. To reduce this risk, we used a fixed codebook, distinguishing unreported from unclear information, and applying consistent criteria across all 83 studies. The reconstruction provides initial evidence of UNITE's applicability, but broader validation remains necessary. Future work should evaluate UNITE across different vehicles, interaction techniques, communication conditions, and tasks. As an openly available testbed, UNITE enables researchers to share setups as configurations rather than prose descriptions, supporting the reconstruction, reproduction, and extension of prior experiments.

Acknowledgments. This work was funded by the Flemish Government under the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" program, R-13509, and by the Special Research Fund (BOF) of Hasselt University, BOF23OWB29. The infrastructure for this work is funded by the European Union – NextGenerationEU project *MAXVR-INFRA* and the Flemish government.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Abubakar, A., Zweiri, Y., Alhammadi, R., Mohiuddin, M.B., Yakubu, M., Seneviratne, L.: Predictor-based control for delay compensation in bilateral teleoperation of wheeled rovers on soft terrains. IEEE Access **12**, 111593–111610 (2024). https://doi.org/10.1109/ACCESS.2024.3443204
2. Arend, M.G., Benz, T.M., Mertens, A., Brandl, C., Nitsch, V.: Do multimodal search cues help or hinder teleoperated search and rescue missions? Ergonomics **66**(9), 1255–1269 (2023). https://doi.org/10.1080/00140139.2022.2144646, https://doi.org/10.1080/00140139.2022.2144646, pMID: 36369787
3. Bagchi, S., Holthaus, P., Beraldo, G., Senft, E., Hernandez Garcia, D., Han, Z., Jayaraman, S.K., Rossi, A., Esterwood, C., Andriella, A., Pridham, P.: Towards improved replicability of human studies in human-robot interaction: Recommendations for formalized reporting. In: Companion Proceedings of the 2023 ACM/IEEE International Conference on Human-Robot Interaction. p. 629–633. HRI Companion '23, Association for Computing Machinery, New York, NY, USA (2023). https://doi.org/10.1145/3568294.3580162, https://doi.org/10.1145/3568294.3580162
4. Bejczy, A., Kim, W., Venema, S.: The phantom robot: predictive displays for teleoperation with time delay. In: Proceedings., IEEE International Conference on Robotics and Automation. pp. 546–551 vol.1 (1990). https://doi.org/10.1109/ROBOT.1990.126037
5. Bingham, L., Kincaid, J., Weno, B., Davis, N., Paddock, E., Foreman, C.: Digital lunar exploration sites unreal simulation tool (dust). In: 2023 IEEE Aerospace Conference. pp. 1–12 (2023). https://doi.org/10.1109/AERO55745.2023.10115607

6. Brookes, J., Warburton, M., Alghadier, M., Mon-Williams, M., Mushtaq, F.: Studying human behavior with virtual reality: The unity experiment framework. Behavior Research Methods **52**(2), 455–463 (2020). https://doi.org/10.3758/s13428-019-01242-0, https://doi.org/10.3758/s13428-019-01242-0
7. Bualat, M., Schreckenghost, D., Pacis, E., Fong, T., Kalar, D., Beutter, B.: Results from testing crew-controlled surface telerobotics on the international space station. In: iSAIRAS-International Symposium on Artificial Intelligence, Robotics and Automation in Space (2014), nASA report no. ARC-E-DAA-TN14584
8. Burns, J.O., Mellinkoff, B., Spydell, M., Fong, T., Kring, D.A., Pratt, W.D., Cichan, T., Edwards, C.M.: Science on the lunar surface facilitated by low latency telerobotics from a lunar orbital platform - gateway. Acta Astronautica **154**, 195–203 (2019). https://doi.org/10.1016/j.actaastro.2018.04.031, https://www.sciencedirect.com/science/article/pii/S0094576517317307
9. Cardinaels, D., Ramakers, R., Veuskens, T., Pietrzak, T., Ruiz, G.A.R., Luyten, K.: Every move you make: Visualizing near-future motion under delay for telerobotics. In: Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems (CHI '26), April 13–17, 2026, Barcelona, Spain (2026). https://doi.org/10.1145/3772318.3791452, https://doi.org/10.1145/3772318.3791452
10. Collins, J., Chand, S., Vanderkop, A., Howard, D.: A review of physics simulators for robotic applications. IEEE Access **9**, 51416–51431 (2021). https://doi.org/10.1109/ACCESS.2021.3068769
11. Cross, M., McIsaac, K.A., Dudley, B., Choi, W.: Negotiating corners with teleoperated mobile robots with time delay. IEEE Transactions on Human-Machine Systems **48**(6), 682–690 (2018). https://doi.org/10.1109/THMS.2018.2849024
12. Daniel Gingell, P.K., Park, S.: Fuzzy shared-control for digital twin-based teleoperated driving under network delay. In: 2024 IEEE 27th International Conference on Intelligent Transportation Systems (ITSC). pp. 3667–3672 (2024). https://doi.org/10.1109/ITSC58415.2024.10919705
13. Davis, J., Smyth, C., McDowell, K.: The effects of time lag on driving performance and a possible mitigation. Trans. Rob. **26**(3), 590–593 (Jun 2010). https://doi.org/10.1109/TRO.2010.2046695, https://doi.org/10.1109/TRO.2010.2046695
14. Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V.: CARLA: An open urban driving simulator. In: Levine, S., Vanhoucke, V., Goldberg, K. (eds.) Proceedings of the 1st Annual Conference on Robot Learning. Proceedings of Machine Learning Research, vol. 78, pp. 1–16. PMLR (13–15 Nov 2017), https://proceedings.mlr.press/v78/dosovitskiy17a.html
15. Dybvik, H., Løland, M., Gerstenberg, A., Slåttsveen, K.B., Steinert, M.: A low-cost predictive display for teleoperation: Investigating effects on human performance and workload. International Journal of Human-Computer Studies **145**, 102536 (2021). https://doi.org/10.1016/j.ijhcs.2020.102536, https://www.sciencedirect.com/science/article/pii/S1071581920301385
16. Eppler, D., Adams, B., Archer, D., Baiden, G., Brown, A., Carey, W., Cohen, B., Condit, C., Evans, C., Fortezzo, C., Garry, B., Graff, T., Gruener, J., Heldmann, J., Hodges, K., Hörz, F., Hurtado, J., Hynek, B., Isaacson, P., Juranek, C., Klaus, K., Kring, D., Lanza, N., Lederer, S., Lofgren, G., Marinova, M., May, L., Meyer, J., Ming, D., Monteleone, B., Morisset, C., Noble, S., Rampe, E., Rice, J., Schutt, J., Skinner, J., Tewksbury-Christle, C.M., Tewksbury, B.J., Vaughan, A., Yingst, A., Young, K.: Desert research and technology studies (drats) 2010 science operations: Operational approaches and lessons learned for managing science during human planetary surface missions. Acta Astronautica **90**(2),

224–241 (2013). https://doi.org/10.1016/j.actaastro.2012.03.009, https://www.sciencedirect.com/science/article/pii/S009457651200080X, nASA's 2010 Desert Research and Technology Studies Mission Objectives and Results

17. Ferraz, M., Ferreira, E., Exter, E.d., van der Hulst, F., Rovina, H., Carey, W., Grenouilleau, J., Krueger, T.: Multisensory real-time space telerobotics. In: Arai, K., Bhatia, R., Kapoor, S. (eds.) Intelligent Computing. pp. 275–298. Springer International Publishing, Cham (2019)
18. Ferrell, W.R.: Remote manipulation with transmission delay. IEEE Transactions on Human Factors in Electronics **HFE-6**(1), 24–32 (1965). https://doi.org/10.1109/THFE.1965.6591253
19. Fong, T., Provencher, C., Micire, M., Diftler, M., Berka, R., Bluethmann, B., Mittman, D.: The human exploration telerobotics project: Objectives, approach, and testing. In: 2012 IEEE Aerospace Conference. pp. 1–9 (2012). https://doi.org/10.1109/AERO.2012.6187043
20. Fong, T., Thorpe, C., Baur, C.: Collaborative control: A robot-centric model for vehicle teleoperation, vol. 1. Carnegie Mellon University, The Robotics Institute Pittsburgh (2001)
21. Gaines, D., Anderson, R., Rabideau, G., Vasavada, A., Verma, V., Estlin, T., Fesq, L., Ingham, M., Maimone, M., Nesnas, I., Doran, G., Huffman, W., Justice, H., Mackey, R.: Productivity challenges for mars rover operations. In: Planning and Robotics Workshop of ICAPS. London, England, United Kingdom (jun 2016), https://ntrs.nasa.gov/citations/20190025594, nTRS Document ID: 20190025594; Report Number: JPL-CL-16-2123
22. Graf, G., Abdelrahman, Y., Xu, H., Abdrabou, Y., Schitz, D., Hußmann, H., Alt, F.: The predictive corridor: A virtual augmented driving assistance system for teleoperated autonomous vehicles. In: Argelaguet, F., McMahan, R., Sugimoto, M. (eds.) ICAT-EGVE 2020 – International Conference on Artificial Reality and Telexistence and Eurographics Symposium on Virtual Environments. The Eurographics Association (2020). https://doi.org/10.2312/egve.20201260
23. Hair, K., Bahor, Z., Macleod, M., Liao, J., Sena, E.S.: The Automated Systematic Search Deduplicator (ASySD): a rapid, open-source, interoperable tool to remove duplicate citations in biomedical systematic reviews. BMC Biology **21**(1), 189 (9 2023). https://doi.org/10.1186/s12915-023-01686-z
24. Hatori, Y., Uchimura, Y.: Obstacle avoidance during teleoperation by model predictive control with time-varying delay. IEEJ Journal of Industry Applications **12**(2), 117–124 (2023). https://doi.org/10.1541/ieejjia.22004524
25. Hofbauer, M., Kuhn, C.B., Petrovic, G., Steinbach, E.: Telecarla: An open source extension of the carla simulator for teleoperated driving research using off-the-shelf components. In: 2020 IEEE Intelligent Vehicles Symposium (IV). pp. 335–340 (2020). https://doi.org/10.1109/IV47402.2020.9304676
26. Hosseini, A., Richthammer, F., Lienkamp, M.: Predictive haptic feedback for safe lateral control of teleoperated road vehicles in urban areas. In: 2016 IEEE 83rd Vehicular Technology Conference (VTC Spring). pp. 1–7 (2016). https://doi.org/10.1109/VTCSpring.2016.7504430
27. Jernberg, C., Andersson, J.: The effect of view, latency, speed and performed task on remote vehicle operators. In: McNally, C., Carroll, P., Martinez-Pastor, B., Ghosh, B., Efthymiou, M., Valantasis-Kanellos, N. (eds.) Transport Transitions: Advancing Sustainable and Inclusive Mobility. pp. 787–794. Springer Nature Switzerland, Cham (2025)

28. Jernberg, C., Sandin, J., Ziemke, T., Andersson, J.: The effect of latency, speed and task on remote operation of vehicles. Transportation Research Interdisciplinary Perspectives **26**, 101152 (2024). https://doi.org/10.1016/j.trip.2024.101152, https://www.sciencedirect.com/science/article/pii/S2590198224001386
29. Jetter, H.C., Raule, L., Gerken, J., Pirk, S.: Mixed-Initiative Dynamic Autonomy Through Variable Levels of Immersion and Control (MIDA-VIC): A New Paradigm for Collaborative Robotic Teleoperation in Space Exploration. In: Bensch, L., Nilsson, T., Nisser, M., Pataranutaporn, P., Schmidt, A., Sumini, V. (eds.) Advancing Human-Computer Interaction for Space Exploration (SpaceCHI 2025). Open Access Series in Informatics (OASIcs), vol. 130, pp. 22:1–22:10. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany (2025). https://doi.org/10.4230/OASIcs.SpaceCHI.2025.22, https://drops.dagstuhl.de/entities/document/10.4230/OASIcs.SpaceCHI.2025.22
30. Kamtam, S.B., Lu, Q., Bouali, F., Haas, O.C.L., Birrell, S.: Network latency in teleoperation of connected and autonomous vehicles: A review of trends, challenges, and mitigation strategies. Sensors **24**(12) (2024). https://doi.org/10.3390/s24123957, https://www.mdpi.com/1424-8220/24/12/3957
31. Kelly, A., Chan, N., Herman, H., Huber, D., Meyers, R., Rander, P., Warner, R., Ziglar, J., Capstick, E.: Real-time photorealistic virtualized reality interface for remote mobile robot control. The International Journal of Robotics Research **30**(3), 384–404 (2011)
32. Koenig, N., Howard, A.: Design and use paradigms for gazebo, an open-source multi-robot simulator. In: 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566). vol. 3, pp. 2149–2154 vol.3 (2004). https://doi.org/10.1109/IROS.2004.1389727
33. Larsson, J., Broxvall, M., Saffiotti, A.: An evaluation of local autonomy applied to teleoperated vehicles in underground mines. In: 2010 IEEE International Conference on Robotics and Automation. pp. 1745–1752 (2010). https://doi.org/10.1109/ROBOT.2010.5509276
34. de Leeuw, J.R.: jspsych: A javascript library for creating behavioral experiments in a web browser. Behavior Research Methods **47**, 1–12 (2015). https://doi.org/10.3758/s13428-014-0458-y, https://doi.org/10.3758/s13428-014-0458-y
35. Levac, D., Colquhoun, H., O'Brien, K.K.: Scoping studies: advancing the methodology. Implementation Science **5**(1), 69 (2010)
36. Lewis, M.E., McCleskey, C.M., Alvarez, V.R., Brown, B.L., Perotti, J.M., Latorella, K.A., Gill, T.R., Gomez, J.D.J., Zeitlin, N.P.: Uncrewed lunar surface operations and support activities. In: 2022 ASCEND. American Institute of Aeronautics and Astronautics (AIAA), Las Vegas, NV (oct 2022). https://doi.org/10.2514/6.2022-4234, https://ntrs.nasa.gov/citations/20220004927, nTRS Document ID: 20220004927
37. Litaker, H.L., Beaton, K.H., Li, Z.Q., Lewis, J.F.: Lunar terrain vehicle (ltv) remote teleoperation studies under four lunar communication latencies. In: 2025 IEEE Aerospace Conference. pp. 1–21 (2025). https://doi.org/10.1109/AERO63441.2025.11068604
38. Maag, C., Gary, S., Merkel, N., Neukum, A.: Studying effects of up- and downlink latency on remote driving using teledriving simulation. In: 2025 IEEE Intelligent Vehicles Symposium (IV). pp. 1321–1326 (2025). https://doi.org/10.1109/IV64158.2025.11097757
39. Matheson, A., Donmez, B., Rehmatullah, F., Jasiobedzki, P., Ng, H.K., Panwar, V., Li, M.: The effects of predictive displays on performance in driving tasks with multi-second latency: Aiding tele-operation of lunar rovers. Proceedings of the Human

Factors and Ergonomics Society Annual Meeting **57**(1), 21–25 (2013). https://doi.org/10.1177/1541931213571007
40. Mathôt, S., Schreij, D., Theeuwes, J.: Opensesame: An open-source, graphical experiment builder for the social sciences. Behavior Research Methods **44**(2), 314–324 (2012). https://doi.org/10.3758/s13428-011-0168-7, https://doi.org/10.3758/s13428-011-0168-7
41. Mavrakis, N., Law, E.L.C., Shum, H.P.H.: Integrating Human-In-The-Loop AI to Tackle Space Communication Delay Challenges. In: Bensch, L., Nilsson, T., Nisser, M., Pataranutaporn, P., Schmidt, A., Sumini, V. (eds.) Advancing Human-Computer Interaction for Space Exploration (SpaceCHI 2025). Open Access Series in Informatics (OASIcs), vol. 130, pp. 15:1–15:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany (2025). https://doi.org/10.4230/OASIcs.SpaceCHI.2025.15, https://drops.dagstuhl.de/entities/document/10.4230/OASIcs.SpaceCHI.2025.15
42. Mellinkoff, B.J., Spydell, M.M., Bailey, W., Burns, J.O.: Quantifying operational constraints of low-latency telerobotics for planetary surface operations. In: 2018 IEEE Aerospace Conference. pp. 1–10 (2018). https://doi.org/10.1109/AERO.2018.8396673
43. Merancy, N., Grantier, J.A., Nelson, G., Bhakta, S., Kessler, P.D., Chai, P.R., Hill, J.J., Nickens, T.M., Cahill, S.A., Bleacher, J.E., Morris-Eckart, A., Cottonham, A., McIntyre, N., Mercer, G., Baird, D.: Moon to mars architecture definition document, revision c. Technical Report ESDMD-001, NASA/TP-20250010956, National Aeronautics and Space Administration (2025)
44. Mirinejad, H., Jayakumar, P., Ersal, T.: Modeling human steering behavior during path following in teleoperation of unmanned ground vehicles. Human Factors **60**(5), 669–684 (2018). https://doi.org/10.1177/0018720818769260, https://doi.org/10.1177/0018720818769260, pMID: 29664713
45. Moniruzzaman, M., Rassau, A., Chai, D., et al.: High latency unmanned ground vehicle teleoperation enhancement by presentation of estimated future through video transformation. Journal of Intelligent & Robotic Systems **106**, 48 (2022). https://doi.org/10.1007/s10846-022-01749-3, https://doi.org/10.1007/s10846-022-01749-3
46. Nielsen, C.W., Goodrich, M.A., Ricks, R.W.: Ecological interfaces for improving mobile robot teleoperation. IEEE Transactions on Robotics **23**(5), 927–941 (2007). https://doi.org/10.1109/TRO.2007.907479
47. Norris, J., Powell, M., Vona, M., Backes, P., Wick, J.: Mars exploration rover operations with the science activity planner. In: Proceedings of the 2005 IEEE International Conference on Robotics and Automation. pp. 4618–4623 (2005). https://doi.org/10.1109/ROBOT.2005.1570832
48. Orsula, A., Geist, M., Olivares-Mendez, M., Martinez, C.: Space robotics bench: Robot learning beyond earth (2025), https://arxiv.org/abs/2509.23328
49. Page, M.J., McKenzie, J.E., Bossuyt, P.M., Boutron, I., Hoffmann, T.C., Mulrow, C.D., Shamseer, L., Tetzlaff, J.M., Akl, E.A., Brennan, S.E., Chou, R., Glanville, J., Grimshaw, J.M., Hróbjartsson, A., Lalu, M.M., Li, T., Loder, E.W., Mayo-Wilson, E., McDonald, S., McGuinness, L.A., Stewart, L.A., Thomas, J., Tricco, A.C., Welch, V.A., Whiting, P., Moher, D.: The prisma 2020 statement: an updated guideline for reporting systematic reviews. BMJ **372** (2021). https://doi.org/10.1136/bmj.n71, https://www.bmj.com/content/372/bmj.n71
50. Peirce, J., Gray, J.R., Simpson, S., MacAskill, M., Höchenberger, R., Sogo, H., Kastman, E., Lindeløv, J.K.: Psychopy2: Experiments in behavior made easy.

Behavior Research Methods **51**(1), 195–203 (2019). https://doi.org/10.3758/s13428-018-01193-y, https://doi.org/10.3758/s13428-018-01193-y
51. Penizzotto, F., García, S., Slawiñski, E., Mut, V.: Delayed bilateral teleoperation of wheeled robots including a command metric. Mathematical Problems in Engineering **2015**(1), 460476 (2015). https://doi.org/10.1155/2015/460476, https://onlinelibrary.wiley.com/doi/abs/10.1155/2015/460476
52. Prakash, J., Vignati, M., Sabbioni, E.: Performance of successive reference pose tracking vs smith predictor approach for direct vehicle teleoperation under variable network delays. IEEE Transactions on Vehicular Technology **73**(4), 4636–4645 (2024). https://doi.org/10.1109/TVT.2023.3339877
53. Prakash, J., Vignati, M., Sabbioni, E., Cheli, F.: Vehicle teleoperation: Human in the loop performance comparison of smith predictor with novel successive reference-pose tracking approach. Sensors **22**(23) (2022). https://doi.org/10.3390/s22239119, https://www.mdpi.com/1424-8220/22/23/9119
54. Rankin, A., Maimone, M., Biesiadecki, J., Patel, N., Levine, D., Toupet, O.: Mars curiosity rover mobility trends during the first 7 years. Journal of Field Robotics **38**(5), 759–800 (2021). https://doi.org/10.1002/rob.22011, https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.22011
55. Reagan, M.L., Janoiko, B.A., Parker, M.L., Johnson, J.E., Chappell, S.P., Abercromby, A.F.: Nasa's analog missions: Driving exploration through innovative testing. In: AIAA SPACE 2012 Conference & Exposition. American Institute of Aeronautics and Astronautics (AIAA), Pasadena, CA (sep 2012). https://doi.org/10.2514/6.2012-5238, https://www.nasa.gov/wp-content/uploads/2016/04/nasa_analog_missions_2012.pdf
56. Sato, Y., Kashihara, S., Ogishi, T.: Implementation and evaluation of latency visualization method for teleoperated vehicle. In: 2021 IEEE Intelligent Vehicles Symposium (IV). pp. 1–7 (2021). https://doi.org/10.1109/IV48863.2021.9575817
57. Schmidt, G.R., Landis, G.A., Oleson, S.R.: Human exploration using real-time robotic operations (herro): A space exploration strategy for the 21st century. Acta Astronautica **80**, 105–113 (2012). https://doi.org/10.1016/j.actaastro.2012.05.036, https://www.sciencedirect.com/science/article/pii/S0094576512002366
58. Schwarz, M., Stückler, J., Behnke, S.: Mobile teleoperation interfaces with adjustable autonomy for personal service robots. In: Proceedings of the 2014 ACM/IEEE International Conference on Human-Robot Interaction. p. 288–289. HRI '14, Association for Computing Machinery, New York, NY, USA (2014). https://doi.org/10.1145/2559636.2563716, https://doi.org/10.1145/2559636.2563716
59. Sheridan, T.B.: Telerobotics, automation, and human supervisory control. MIT Press, Cambridge, MA, USA (1992)
60. Sierotowicz, M., Weber, B., Belder, R., Bussmann, K., Singh, H., Panzirsch, M.: Investigating the influence of haptic feedback in rover navigation with communication delay. In: Nisky, I., Hartcher-O'Brien, J., Wiertlewski, M., Smeets, J. (eds.) Haptics: Science, Technology, Applications. pp. 527–535. Springer International Publishing, Cham (2020)
61. Slawinski, E., Mut, V.A., Postigo, J.F.: Teleoperation of mobile robots with time-varying delay. IEEE Transactions on Robotics **23**(5), 1071–1082 (2007). https://doi.org/10.1109/TRO.2007.906249
62. Steinfeld, A., Fong, T., Kaber, D., Lewis, M., Scholtz, J., Schultz, A., Goodrich, M.: Common metrics for human-robot interaction. In: Proceedings of the 1st ACM

SIGCHI/SIGART Conference on Human-Robot Interaction. p. 33–40. HRI '06, Association for Computing Machinery, New York, NY, USA (2006). https://doi.org/10.1145/1121241.1121249, https://doi.org/10.1145/1121241.1121249
63. Storms, J., Chen, K., Tilbury, D.: A shared control method for obstacle avoidance with mobile robots and its interaction with communication delay. The International Journal of Robotics Research **36**(5-7), 820–839 (2017). https://doi.org/10.1177/0278364917693690
64. Storms, J., Tilbury, D.: Equating user performance among communication latency distributions and simulation fidelities for a teleoperated mobile robot. In: 2015 IEEE International Conference on Robotics and Automation (ICRA). pp. 4440–4445 (2015). https://doi.org/10.1109/ICRA.2015.7139813
65. Trivedi, S., Warg, F.: Evaluating the safety impact of network disturbances for remote driving with simulation-based human-in-the-loop testing. In: 2023 53rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W). pp. 215–222 (2023). https://doi.org/10.1109/DSN-W58399.2023.00059
66. Vozar, S., Storms, J., Tilbury, D.M.: Development and analysis of an operator steering model for teleoperated mobile robots under constant and variable latencies. Robotica **36**(2), 167–186 (2018). https://doi.org/10.1017/S0263574716000679
67. Wick, J., Callas, J., Norris, J., Powell, M., Vona, M.: Distributed operations for the mars exploration rover mission with the science activity planner. In: 2005 IEEE Aerospace Conference. pp. 4162–4173 (2005). https://doi.org/10.1109/AERO.2005.1559721
68. Wormnes, K., Carey, W., Krueger, T., Cencetti, L., den Exter, E., Ennis, S., Ferreira, E., Fortunato, A., Gerdes, L., Hann, L., Lombardi, C., Luzzi, E., Martin, S., Massironi, M., Payler, S., Pereira, A., Rossi, A.P., Pozzobon, R., Sauro, F., Schoonejans, P., van der Hulst, F., Grenouilleau, J.: Analog-1 iss – the first part of an analogue mission to guide esa's robotic moon exploration efforts. Open Astronomy **31**(1), 5–14 (2022). https://doi.org/10.1515/astro-2022-0002, https://doi.org/10.1515/astro-2022-0002
69. Yoshida, A., Dossa, R.F.J., Di Vincenzo, M., Sujit, S., Douglas, H., Arulkumaran, K.: A multi-user multi-robot multi-goal multi-device human-robot interaction manipulation benchmark. Frontiers in Robotics and AI **Volume 12 - 2025** (2025). https://doi.org/10.3389/frobt.2025.1528754, https://www.frontiersin.org/journals/robotics-and-ai/articles/10.3389/frobt.2025.1528754
70. Zhang, Q., Xu, Z., Wang, Y., Yang, L., Song, X., Huang, Z.: Predicted trajectory guidance control framework of teleoperated ground vehicles compensating for delays. IEEE Transactions on Vehicular Technology **72**(9), 11264–11274 (2023). https://doi.org/10.1109/TVT.2023.3269517
71. Zheng, Y., Brudnak, M.J., Jayakumar, P., Stein, J.L., Ersal, T.: An experimental evaluation of a model-free predictor framework in teleoperated vehicles. IFAC-PapersOnLine **49**(10), 157–164 (2016). https://doi.org/10.1016/j.ifacol.2016.07.513, https://www.sciencedirect.com/science/article/pii/S2405896316307273, 13th IFAC Workshop on Time Delay Systems TDS 2016
72. Zimmerman, M., Bagchi, S., Marvel, J., Nguyen, V.: An analysis of metrics and methods in research from human-robot interaction conferences, 2015–2021. In: 2022 17th ACM/IEEE International Conference on Human-Robot Interaction (HRI). pp. 644–648 (2022). https://doi.org/10.1109/HRI53351.2022.9889595