Coordinated Monitoring of the Timeliness of Commands and Video Data for Remote Vehicle Control

Yurii Yarmola[1], Maksym Oleksiv[2]

1,2 Department of Computer Engineering, Lviv Polytechnic National University,
UKRAINE, Lviv, 12 S. Bandera St., 79013

E-mail: 1 yurii.yarmola.mkisp.2025@lpnu.ua; 2 maksym.v.oleksiv@lpnu.ua

Abstract. A frame-linked method coordinates command validity, primary-frame timestamps and explicit recovery in remote vehicle control. Discrete-event experiments assess protective response, recovery processing and nominal rejection. Simulated video freezes and repeated frame transmission trigger blocking; threshold sensitivity reveals the trade-off between video-data timeliness and control availability.

Keywords: teleoperation, digital video processing, command admission, frame provenance, watchdog, discrete-event simulation, explicit recovery.

Introduction and Related Work

Remote driving requires both timely commands and usable visual feedback. A reception-only watchdog can remain satisfied while the camera stream freezes, because steering or traction messages continue to arrive. The relevant question is therefore not only whether a command was delivered, but whether its execution still refers to sufficiently recent information. This paper develops a frame-linked command admission method and evaluates its response to video freezes, repeated frame delivery and variable communication delays using discrete-event simulation.

Existing approaches address complementary parts of this problem. TUM Teleoperation is an open-source remote-driving stack with monitoring and a safety module that can restrict commands or initiate a stop [1]. Agenet is a Python package for analytical and simulation-based assessment of Age of Information (AoI), the elapsed time since generation of the latest received update [2]. The Age of Loop (AoL) metric extends the assessment of information timeliness to a closed-loop control system by linking the most recently applied action or state to the information that caused it [3]. Recent research also uses AoL in resource allocation for queued remote control of Markov decision processes [4].

Avatar is an immersive teleoperation interface focused on panoramic visual feedback and low display latency [5]. Motion-to-Motion measurements instead concern physical operator and actuator events, which cannot be replaced by software output timestamps [6]. These studies justify admitting commands based on the timeliness of information, but they do not establish the proposed rule as a new age metric or a new general concept of a protective stop. The narrower contribution is an explicit combination of immutable primary-frame provenance, a joint command deadline and versioned recovery, with reproducible tests of their interaction. The comparison concerns published mechanisms, not empirical benchmarking of those complete platforms.

Frame-Linked Admission Method

A primary frame is a new image buffer received by the source-side software from the camera. It receives an identifier k and a monotonic receipt timestamp t_k. Reusing that buffer retains both values even if transmission assigns a different presentation timestamp. Identical pixels do not imply reuse: two newly acquired frames of a static scene remain distinct. The method does not detect a camera that internally produces frozen content as apparently new buffers.

The sender maintains a registry mapping a video-session identifier and a transport token to (k, t_k). A command carries that token, a monotonically increasing sequence number n, a permission version g and an explicit rearm flag. Version g separates commands issued under the permissions

before and after a protective block. Unknown or ambiguous frame mappings do not permit motion. The functional structure is shown in Fig. 1.

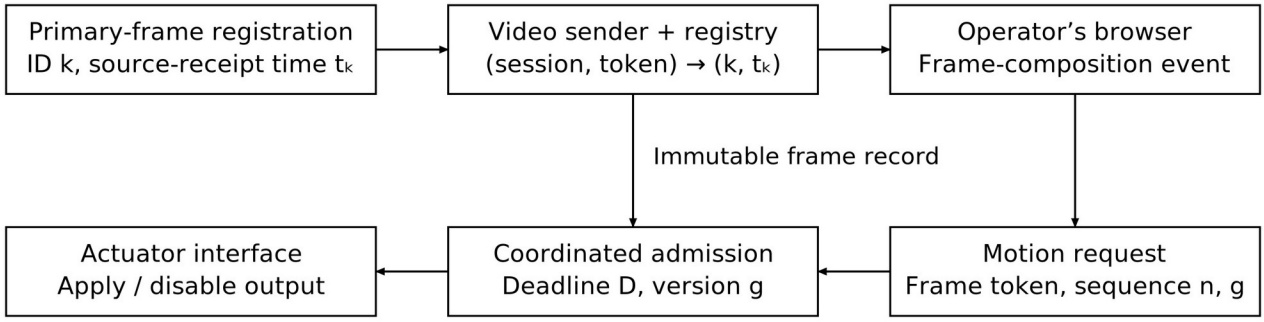

Fig. 1. Frame provenance and command admission in the proposed system

Let r_j be command reception time at the onboard node, T_c its allowed lifetime after reception, and T_v the maximum permitted age of its referenced primary frame. At application time t, the deadline D_j and remaining budget $B_j(t)$ are:

$$D_j = \min(r_j + T_c, t_k + T_v); \quad B_j(t) = \max(0, D_j - t). \quad (1)$$

A motion command is admitted only if $B_j(t) > 0$, the session and permission version are current, its sequence number exceeds those already processed in that version, and its frame is known. Reception and final output application are separate events; time must be checked again at the latter. T_c is not the age of the command since creation and is not a one-way network-delay estimate.

Before a new command is processed, expiry of the previous time limit causes blocking and a single increment of g. Periodic checks also disable the active output after expiry. Invalid requests do not extend the current time limit. A stop command takes precedence regardless of video-data timeliness. A connection break or a change in the video session revokes permission. Recovery requires a primary frame received strictly after blocking and a new explicit command with the current g. Neither restored connectivity nor a continuously held previous input should resume motion.

Eq. (1) gives $D_j \leq t_k + T_v$. Repeated commands referring to one primary frame therefore cannot extend that frame's deadline. This is a property of the rule, not a proof of physical safety. The output can remain active until the next check executes; scheduler, interface and mechanical delays are outside this algebraic statement.

Technical Implementation and Measurement Events

The sender must preserve an unambiguous association between each transmitted frame and its primary-frame identifier and receipt time. Within each video session, the association must identify the source stream and the frame. Changes or reuse of transport identifiers must not overwrite the original receipt time or create ambiguity. Obsolete records are removed, and a new session is established before identifiers can become ambiguous. Repeated transmissions retain the original pair (k, t_k).

The operator interface confirms the software event at which a received frame is submitted for display and associates this event with the frame identifier returned in a command [7]. Missing confirmation or an unresolved frame mapping keeps motion blocked. This event confirms software processing, not perception by the operator. The sender-to-interface correspondence must be verified when integrating a specific video system; the simulation tests the admission rules rather than a browser implementation.

Admission decisions, periodic checks, stop requests and session resets must update one shared control state in a defined order. The final validity check and application of the output form one protected operation, so another handler cannot change the permission between them. Concurrently received camera data must be transferred consistently to the component managing this state. The synchronization mechanism must match the execution model: protection of asynchronous tasks alone does not ensure protection between operating-system threads [8]. The simulation evaluates ordered state transitions and a logical output, not concurrent hardware execution.

The proposed onboard log uses one monotonic clock. It records t_k immediately after obtaining the image buffer and before processing; r_j after receiving the command and before parsing; t_{off} after completing the output-disable write; and t_{app} after applying the first accepted nonzero rearm command. The video-age boundary $t_{exp} = t_k + T_v$ is computed, not taken from a later timer callback. The timestamp unit does not establish measurement accuracy, and t_k is not sensor exposure time.

The method is evaluated using three quantities. Protective response $R = t_{off} - t_{exp}$ is evaluated during a frame-freeze episode with continuing commands. Recovery processing $Q = t_{app} - r_j$ concerns the first accepted recovery command, excluding waiting for network recovery or operator action. Nominal rejection $P = 100\ N_{rejected} / N_{offered}$ measures denial of well-formed motion requests in a specified fault-free workload. It takes into account blocking caused by outdated frames and waiting for explicit recovery, not just implementation errors. Correctness against expected decisions is tested separately; an intentional protective rejection is not considered an implementation error.

Simulation Design

The method was evaluated with a discrete-event simulation implemented in Python using its standard library. The model uses integer milliseconds and a Boolean control output; vehicle-motion dynamics are outside its scope. Frames are created every 40 ms and new motion requests are generated every 50 ms. Primary-frame-to-browser delay is sampled uniformly from the integer ranges in Table 1; command return delay is uniform from 10 to 30 ms. These distributions are synthetic study conditions, not measured network traces. Loss, codec processing details and human reaction time are not modeled.

Table 1

Synthetic delay profiles and frame age at command reception

Profile	Video delay, ms	Median A_j, ms	95th percentile A_j, ms	Maximum A_j, ms
L	60–100	120	147	160
M	120–200	198	239	260
H	200–320	292	348	380

Here $A_j = r_j - t_k$ includes the video path, the wait until the next scripted request and its return path. Table 1 reports medians of the five run-level medians and 95th percentiles, and the maximum over all five runs. Out-of-order display events are discarded by increasing primary-frame identifier. Permission-version notifications and recovery acknowledgments take 20 ms in the model. After a block notice, the scripted operator model makes a distinct explicit recovery attempt when a post-block frame is available. This test action is not an automatic recovery policy for the practical system.

The command lifetime T_c is set to 400 ms and the periodic-check interval to 100 ms. Request processing takes a fixed, assumed 1 ms. Runs last 75 simulated seconds; the first 15 seconds are excluded. Seven frame-age thresholds, 100, 150, 200, 250, 300, 400 and 600 ms, are tested across three profiles and seeds 101–105, giving 105 fault-free runs. Random-number streams for video,

commands and checker phase are separate; a seed is reused across thresholds. Each profile–threshold cell contains 6 000 offered requests.

Two faults are injected from 30 to 32 s in profile L at T_v = 300 ms: suspension of new primary frames, and repeat delivery of the last primary frame under new transport identifiers. Both retain the last source receipt time, 29 960 ms. Five seeds per fault give ten episodes. For comparison, a reception-only watchdog is modeled on the same command-arrival trace, ignoring frame age and permission versions. At simultaneous events, timer checks precede command application.

Results and Threshold Interpretation

All 30 functional tests passed, including expiry boundaries, duplicate sequence numbers, unknown frames, immutable timestamp enforcement, old permission versions, explicit stop, session reset and recovery prerequisites. One test also executes 10 000 seeded deadline cases against an independent Boolean expression for both time constraints. These finite checks support implementation consistency, not exhaustive verification of every possible event sequence.

In all ten injected-fault episodes, coordinated admission disabled the model output; the reception-only comparator remained active throughout the two-second fault interval. Both source-freeze and repeat-transmission faults produced the same per-seed response values: 4, 9, 3, 4 and 2 simulated milliseconds after t_{exp}. Continuing command processing detects expiry in addition to the periodic checker. These values are phase-dependent model outputs, not measured reaction times on the board. A separate 100-phase checker-only sweep yields 0–99 ms, with a median of 49.5 ms, under ideal scheduling.

Accepted recovery requests took 1 simulated millisecond from reception to application because that processing delay was explicitly assigned. This confirms recovery sequencing but is not evidence of fast real implementation. Hardware recovery-processing latency remains unmeasured. The comparator’s response is recorded as “not detected”, never as zero; recovery without an earlier block is “not applicable”.

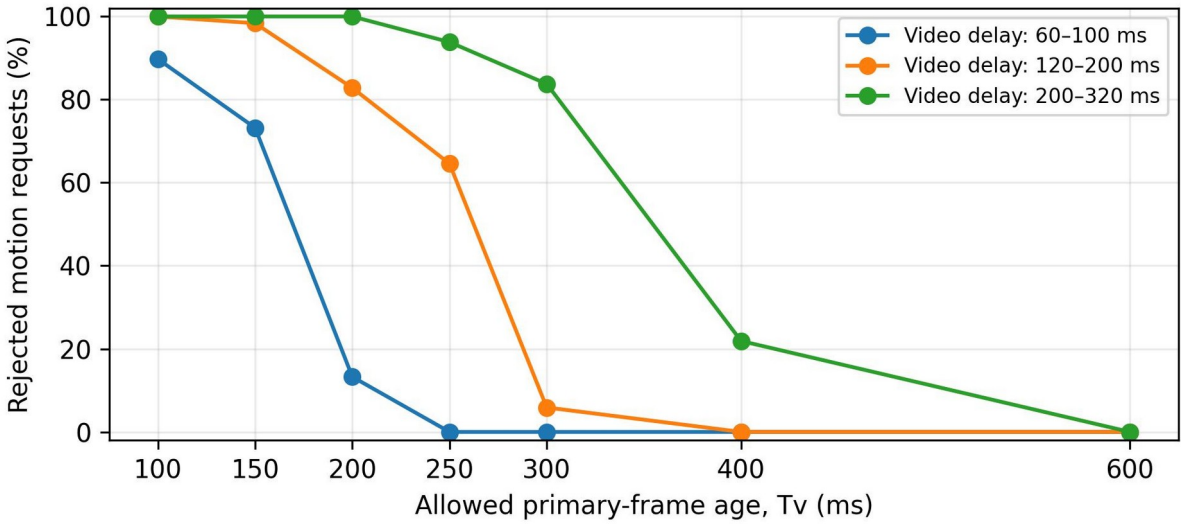

Fig. 2. Nominal command rejection under synthetic delay profiles (five seeds)

At T_v = 300 ms, nominal rejection was 0 of 6 000 requests in L, 353 of 6 000 (5.88%) in M and 5 026 of 6 000 (83.77%) in H. The run-level ranges were 0%, 4.17–8.67% and 83.00–84.25%, respectively. No threshold in the sweep is claimed universally optimal. The first tested thresholds with zero rejection in these finite traces were 250, 400 and 600 ms for L, M and H. The reception-only comparator accepted all nominal requests, but did not detect either video fault.

Choosing T_v from the age distribution alone is insufficient. If p_j is local processing time and h_j is the interval from this application to the next valid application attempt, uninterrupted permission additionally requires $T_v > A_j + p_j + h_j$ and $T_c > p_j + h_j$. These are conditional design inequalities, not physical safety limits. Profile M illustrates the distinction: its observed maximum reception age is 260 ms, below 300 ms, yet permission can expire before the next application, followed by a recovery lockout. Increasing T_v improves availability but permits older visual evidence. The 300 ms threshold is therefore an illustrative operating point, not a hardware-calibrated choice.

Practical Application and Scope of the Results

The practical application considered is a browser-controlled mobile platform built around Radxa Zero 3W. The onboard software receives commands over WebSocket, provides video through WebRTC and controls the actuators through a digital interface. In the proposed architecture, the admission layer sits between command reception and output application, using a primary-frame registry and browser feedback.

The relevant operating scenario is continued steering or traction input while the displayed image stops updating. The frame-linked deadline expires even if requests continue to arrive or the stored image is retransmitted under new transport tokens. Blocking replaces the permission version; recovery requires a new post-block frame and a distinct operator command. Both injected video-fault scenarios demonstrate this behavior in the simulation.

For deployment, sender timestamps must be matched to browser feedback. The threshold T_v must be selected from measured A_j and inter-application gaps, subject to the driving task, speed and stopping behavior. The sensitivity analysis helps assess the availability cost of a candidate threshold before hardware validation; it does not establish an acceptable stopping distance.

The model assumes honest client reports, successful serial state transitions and no output-write failures. It does not cover malicious acknowledgments, board-process failure, sensor-internal freezes, operating-system scheduling stalls or mechanical dynamics. The nominal rejection fraction depends on the scripted recovery workload and is not a direct measure of human driving usability. Physical operator-to-actuator measurement requires separate instrumentation [6]. A practical safety design would need independent fault handling beyond this software rule.

Conclusion

A coordinated command admission method was specified and evaluated through discrete-event simulation for remote vehicle operation. Immutable primary-frame timestamps restrict command validity, while permission versions prevent pre-block requests from restoring motion. The method disabled the model output in all ten injected video-fault episodes, whereas the reception-only comparator remained active. At T_v = 300 ms, nominal rejection ranged from 0% to 83.77% across the three delay profiles. Practical parameter selection must therefore consider both frame age and the interval until the next command update. The results characterize the admission logic under the stated model assumptions; integration and physical timing require further validation.

Academic Integrity, Author Contributions and AI Use

Yurii Yarmola provided the vehicle prototype and technical materials, selected the research direction and defined the study requirements and revisions. Maksym Oleksiv contributed scientific supervision and methodological review of thresholds, evaluation criteria, technical mechanisms and the presentation of results. External ideas are cited, and the assumptions, simulation outputs and conditions of practical application are distinguished.

ChatGPT (OpenAI, accessed through its web interface) was used as an auxiliary tool to assist with analyzing the supplied materials, refining the method description, literature discovery, drafting and language editing, and developing simulation and test code. Computational checks were run with its code-execution tools. The assistance followed author-defined tasks and several rounds of substantive revisions based on author decisions and supervisor feedback. The authors retain responsibility for verifying sources, interpreting the results and approving the final manuscript.

References

[1] Kerbl, T., Brecht, D., Gehrke, N., Karunainayagam, N., Krauss, N., Pfab, F., Taupitz, R., Trautmannsheimer, I., Su, X., Wolf, M.-M., & Diermeyer, F. (2025). TUM Teleoperation: Open source software for remote driving and assistance of automated vehicles. In *2025 IEEE Intelligent Vehicles Symposium (IV)* (pp. 2593–2600). IEEE. https://doi.org/10.1109/IV64158.2025.11097531

[2] Basnayaka, C. M. W., & Fachada, N. (2025). Agenet: Age of Information evaluation in wireless networks. *SoftwareX, 30*, Article 102086. https://doi.org/10.1016/j.softx.2025.102086

[3] de Sant Ana, P. M., Marchenko, N., Popovski, P., & Soret, B. (2021). Age of Loop for wireless networked control systems optimization. In *2021 IEEE 32nd Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*. IEEE. https://doi.org/10.1109/PIMRC50174.2021.9569366

[4] Arun, P. R., Mubarak, M., & Sukumaran, V. B. (2025). On resource allocation for remote control of MDPs over queues using Age-of-Loop. In *2025 National Conference on Communications (NCC)* (pp. 1–6). IEEE. https://doi.org/10.1109/NCC63735.2025.10982939

[5] Li, J., Han, D., Xu, J., Li, K., & Ma, Z. (2026). Design and evaluation of Avatar: An ultra-low-latency immersive human–machine interface for teleoperation. *Displays, 92*, Article 103292. https://doi.org/10.1016/j.displa.2025.103292

[6] Provost, F., Hawlader, F., Testouri, M., & Frank, R. (2026). Motion-to-motion latency measurement framework for connected and autonomous vehicle teleoperation. In *2026 21st Wireless On-Demand Network Systems and Services Conference (WONS)* (pp. 57–60). https://doi.org/10.23919/WONS68803.2026.11501834

[7] Guilbert, T. (Ed.). (2024, August 2). *HTMLVideoElement.requestVideoFrameCallback()* [Draft Community Group Report]. Web Platform Incubator Community Group. https://wicg.github.io/video-rvfc/

[8] Python Software Foundation. (n.d.). *Synchronization primitives*. Python documentation. Retrieved November 1, 2026, from https://docs.python.org/3/library/asyncio-sync.html