

# Enhanced Teleoperation for Manual Remote Driving: Extending ADAS Remote Control Towards Full Vehicle Operation <sup>†</sup>

İsa Karaböcek \*, Ege Özdemir and Batıkan Kavak

AVL Türkiye Research and Engineering, İstanbul 34885, Türkiye; batikan.kavak@avl.com (E.Ö.); ege.ozdemir@avl.com (B.K.)

\* Correspondence: isa.karabocek@avl.com

<sup>†</sup> Presented at the 12th International Electronic Conference on Sensors and Applications (ECSA-12), 12–14 November 2025; Available online: <https://sciforum.net/event/ECSA-12>.

## Abstract

This study advances prior work on the remote control of Advanced Driver Assistance Systems (ADAS) by introducing a full manual teleoperation mode that enables remote control over both longitudinal and lateral vehicle dynamics via accelerator, brake, and steering inputs. The core contribution is a flexible, dual-mode teleoperation architecture that allows seamless switching between assisted ADAS control and full manual operation, depending on driving context or system limitations. While teleoperation has been explored primarily for autonomous fallback or direct remote driving, few existing systems integrate dynamic mode-switching in a unified, real-time control framework. Our system leverages a wireless game controller and a Robot Operating System (ROS)-based vehicle software stack to translate remote human inputs into low-latency vehicle actions, supporting robust and adaptable remote driving. This design maintains a human-in-the-loop approach, offering improved responsiveness in complex environments, edge-case scenarios, or during autonomous system fallback. The proposed solution extends the applicability of teleoperation to a broader range of use cases, including remote assistance, fleet management, and emergency response. Its novelty lies in the integration of dual-mode teleoperation within a modular architecture, bridging the gap between ADAS-enhanced autonomy and full remote manual control.

Academic Editor(s): Name

Published: date

**Citation:** Karaböcek, İ.; Özdemir, E.; Kavak, B. Enhanced Teleoperation for Manual Remote Driving: Extending ADAS Remote Control Towards Full Vehicle Operation. *Eng. Proc.* **2025**, *volume number*, *x*. <https://doi.org/10.3390/xxxxx>

**Copyright:** © 2025 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

**Keywords:** teleoperation; ADAS; connectivity; remote control; cruise control; manual control

## 1. Introduction

In recent years, ADAS have become integral to the automotive industry's pursuit of enhanced safety and driving efficiency. Features such as adaptive cruise control (ACC), lane keeping (LKA), and lane change assistance (LCA) are widely adopted, offering partial automation to support drivers in routine tasks. However, ADAS functions often exhibit limitations in handling complex or unpredictable scenarios, especially in edge cases where contextual understanding and rapid adaptability are essential. In such situations, human intervention remains necessary to ensure safety and operational continuity.

Teleoperation, the practice of remotely controlling vehicles or systems via human input, has emerged as a promising solution to bridge these autonomy gaps. In our previous work, titled “Remote Control of ADAS Features: A Teleoperation Approach to Mitigate Autonomous Driving Challenges”, we introduced a teleoperation system that enabled remote control of selected ADAS features, specifically targeting ACC and LCA. That system demonstrated how human-in-the-loop intervention can effectively supplement autonomous behavior in critical scenarios, mitigating risks associated with full autonomy by allowing real-time operator decisions [1].

Building on this foundation, the current study significantly expands the scope of teleoperation by introducing a full manual remote driving mode. Unlike traditional ADAS-level intervention, this new approach enables remote control over both longitudinal (acceleration, braking) and lateral (steering) dynamics of the vehicle, offering the operator complete manual control when required. The key contribution of this work lies in the design and implementation of a dual-mode teleoperation architecture that allows seamless transitions between assisted ADAS-level control and full manual remote operation, depending on situational demands or system limitations.

While prior studies have explored teleoperation in contexts such as autonomous fallback and fleet support, few have addressed the integration of dynamic mode-switching within a unified and modular framework. Our system addresses this gap by incorporating a ROS-based control stack and a wireless game controller interface, enabling intuitive and low-latency remote operation. This flexible architecture is designed not only to enhance safety during autonomous fallback events but also to support broader use cases, such as remote fleet management, remote assistance in constrained environments, and emergency response scenarios.

By extending the capabilities of teleoperation beyond ADAS feature control and into full remote manual driving, this work contributes to the development of robust, adaptable, and human-centered remote driving systems. The remainder of this paper is structured as follows: Section 2 reviews related work in teleoperation systems and manual remote driving. Section 3 details the architecture and components of the developed dual-mode system. Section 4 presents experimental setup and evaluation results. Section 5 concludes the paper and outlines potential directions for future research.

## 2. Related Work

Teleoperation in vehicle systems ranges from ADAS-level intervention, where operators issue high-level commands for limited features, to full direct remote control, enabling steering and throttle inputs. A human-in-the-loop ADAS control framework allowed remote activation of ACC, LKA and LCA features, demonstrating effective intervention in edge-case scenarios using standard control hardware [1]. Other systems explore remote assistance for ADAS failures but often face challenges in operator situational awareness in remote settings [2,3]. More recent work extends teleoperation toward hands-on remote control with real vehicles, integrating latency detection and mode-switch mechanisms [4,5].

Broad surveys categorize teleoperation into ADAS support, full control, and hybrid fallback paradigms [6]. Deep reinforcement learning-based intent recognition methods have been introduced to dynamically switch modes and reduce operator burden [7]. Field-deployed mining and construction systems also emphasize full remote control in safety-critical operations [8], while studies on teleoperated excavators show that assisted (trajectory-guided) modes significantly improve control accuracy and operator trust [9].

Operator interface design remains critical: comparative studies show that command separation (path vs. velocity) yields better usability, and context-aware Graphical User

Interface (GUI) layouts significantly improve task efficiency and operator satisfaction in remote driving environments [10,11].

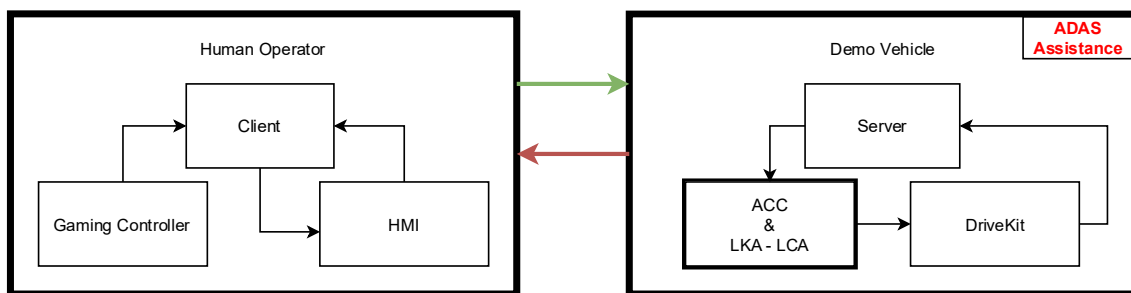
Table 1 summarizes key differences between the proposed controller and representative teleoperation systems in literature. While prior works focus on either ADAS-level intervention or full remote control, our approach integrates both modes in a modular architecture with predictive latency handling and automatic mode switching.

**Table 1.** Comparison of Proposed Controller with Existing Teleoperation Solutions.

Reference	Control Type	Mode Switching	ADAS Integration	Novelty
[2] Fallah et al.	ADAS fallback	Manual	GPS + LiDAR	Human-in-the-loop fallback
[4] Georg et al.	Full control	None	No	Sensor-actuator latency study
[6] Majstorovic et al.	Trajectory-guided	Manual	Yes	Enhanced trajectory planning
[7] Kizilkaya et al.	Hybrid	DRL-based	No	DRL-based mode switching
This work	Modular dual-mode	Automatic + Manual	Full ADAS support	Unified controller with seamless mode transition

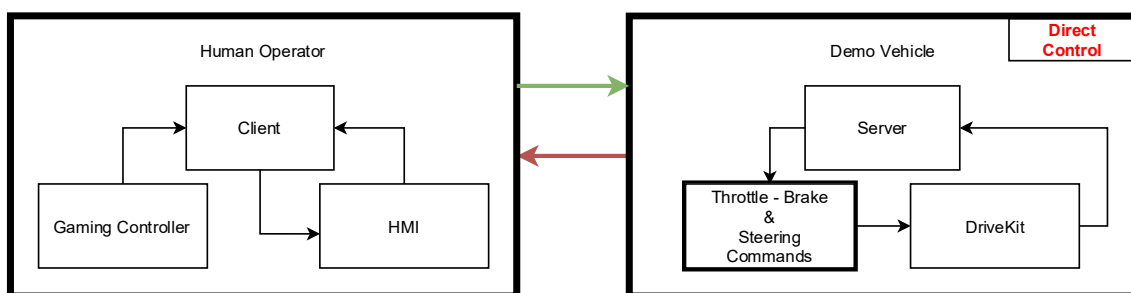
### 3. System Architecture and Direct Control Implementation

Teleoperation systems span a continuum of operator involvement, from high-level assistance to full manual control. In our prior work, we adopted a remote assistance paradigm, leveraging built-in ADAS functionalities such as ACC, LKA, and LCA to intervene in highway scenarios requiring human oversight [1]. These features allowed an operator to supervise and selectively control longitudinal and lateral behavior by remotely triggering the vehicle’s ADAS components (see Figure 1).



**Figure 1.** Remote Assistance Teleoperation Structure.

Building on that foundation, the current implementation transitions toward a direct control architecture (see Figure 2), providing the operator with low-level command authority over throttle, brake, and steering actions. This shift was motivated by the limitations observed in ADAS-based assistance, particularly in dynamic or edge-case highway conditions where more granular control is needed. To support this, new input mappings were developed on a standard Xbox game controller, allowing the operator to issue real-time commands via analog and digital triggers.



**Figure 2.** Direct Control Teleoperation Structure.

The system architecture continues to follow a client-server model. The client system, operated by the human driver, communicates with the vehicle over a local Wi-Fi network using the User Datagram Protocol (UDP). As in the previous implementation, the Xbox game controller is connected to the client machine via Bluetooth, allowing the operator to issue real-time throttle, brake, and steering commands. A web-based GUI remains integrated into the client interface, providing visual feedback such as vehicle telemetry, ADAS states, and camera streams to maintain situational awareness. A new mode-switch button was introduced on the controller, enabling seamless transitions between remote assistance mode and full direct control. This allows the teleoperator to utilize ADAS-assisted features such as ACC or LKA in high-speed highway conditions and switch to direct manual control in lower-speed or complex scenarios. To reduce latency and improve communication reliability, the onboard wireless module was upgraded from Wi-Fi 4 to Wi-Fi 5.

### 3.1. Operator Control Modes and Input Integration

The teleoperation system allows the human operator to control the vehicle through a Bluetooth-connected Xbox game controller interfaced with a host application running on the client machine. This client is responsible for interpreting input commands and transmitting them to the vehicle server over a local Wi-Fi network using the UDP protocol. The communication link between the game controller and the client remains based on Bluetooth, as in the previous implementation, maintaining a typical latency of around 10 ms. The system is event-driven, meaning inputs are processed and sent to the server as soon as a button or joystick movement is detected.

A central contribution of this work is the addition of a **mode-switching mechanism**, enabling seamless transitions between **remote assistance** and **direct control**. In remote assistance mode, the operator interacts with the vehicle's existing ADAS stack, remotely triggering features such as ACC or LCA. In contrast, direct control mode allows the operator to manually issue throttle, brake, and steering commands, bypassing ADAS logic and engaging with low-level actuators via the DriveKit interface. This dual-mode control is essential in complex or low-speed environments (e.g., tight turns, construction zones) where direct manipulation is more effective, while high-speed driving on highways remains more suitable for ADAS-guided remote assistance.

To facilitate this transition, a **dedicated mode-switch button** was integrated into the controller interface. When activated, the system transitions between control paradigms, while maintaining feedback and status visualization through the accompanying web-based GUI. This interface allows the operator to monitor real-time vehicle telemetry, ADAS status, and live camera streams.

All controller inputs are handled via a custom input manager built on Simple Direct-Media Layer (SDL), which captures the real-time state of buttons, triggers, and joystick axes. These inputs are then serialized into structured messages and transmitted to the server over the network using the UDP protocol. SDL provides low-latency, cross-platform access to joystick devices and ensures robust compatibility with the Xbox game controller.

The system operates in two distinct teleoperation modes—**ADAS Mode (Remote Assistance)** and **Manual Mode (Direct Control)**—each mapped to specific controller input configurations. Mode transitions are managed via dedicated switch buttons, enabling the operator to fluidly shift between high-level ADAS-guided control and low-level manual command without interrupting the session. The complete input mapping for both modes is presented in Table 2, and the corresponding button layout of the Xbox controller is illustrated in Figure 3.

**Table 2.** Controller Input Mapping for ADAS and Manual Teleoperation Modes.

<b>ADAS Mode (Remote Assistance):</b>	<b>Manual Mode (Direct Control)</b>
Switch to Manual Mode: <b>START</b>	Switch to ADAS Mode: <b>BACK</b>
Activate ACC: <b>X</b>	
Set ACC Speed: <b>A</b>	Acceleration Command: <b>Right Trigger (RT)</b>
Increase ACC Speed: <b>Y</b>	Brake Command: <b>Left Trigger (LT)</b>
Decrease ACC Speed: <b>A</b>	Steering Control: <b>Left Stick Horizontal Axis</b>
Lane Change Left: <b>LB</b>	
Lane Change Right: <b>RB</b>	

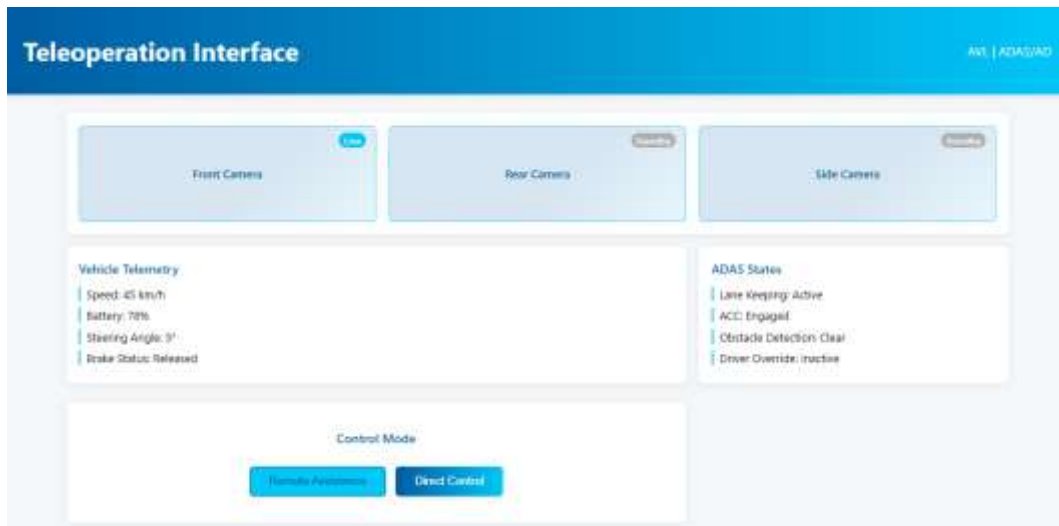
**Figure 3.** Button layout of the gaming controller.

This modular control interface allows for **dynamic and seamless switching** between assisted and direct control modes, adapting to contextual driving needs. For instance, the operator can utilize ADAS in structured highway segments and instantly switch to manual control in urban or constrained environments, maintaining full situational authority. This approach improves system flexibility, reduces operational delays during transitions, and enhances overall remote driving responsiveness.

### 3.2. HMI Interface

The teleoperation system features an enhanced web-based Human-Machine Interface (HMI) that provides comprehensive real-time feedback and control capabilities for the teleoperator. Building upon our previous work, this interface introduces a dual-mode operation system with seamless switching between Remote Assistance and Direct Control modes, significantly improving operator situational awareness and control flexibility.

The web-based GUI integrates multiple camera streams, vehicle telemetry data, and ADAS system states into a unified dashboard Figure 4. This professional interface ensures optimal readability and usability during critical teleoperation scenarios.



**Figure 4.** Human Machine Interface.

Key features of the enhanced HMI include:

- **Multi-Camera Video Streams:** Displays live feeds from front, rear, and side cameras with real-time status indicators, providing 360-degree environmental awareness for the teleoperator.
- **Vehicle Telemetry Panel:** Real-time monitoring of critical vehicle parameters including speed, battery level, steering angle, and brake status, enabling informed decision-making during teleoperation.
- **ADAS State Monitoring:** Comprehensive display of Advanced Driver Assistance Systems status, including LKA, ACC, obstacle detection, and driver override states.
- **Dual-Mode Control Switch:** Interactive toggle between Remote Assistance and Direct Control modes, allowing operators to adapt their intervention level based on the driving scenario complexity and safety requirements.
- **Status Indicators:** Visual feedback system distinguishing between active (“Live”) and standby camera feeds, providing clear operational context.

The interface’s responsive design ensures compatibility across different screen sizes and devices, while the intuitive layout minimizes cognitive load on the teleoperator. This enhanced HMI significantly improves the operator’s ability to maintain situational awareness and execute precise control commands in both assistance and direct control scenarios.

### 3.3. Communication Structure

In our previous study, the teleoperation system was implemented using Wi-Fi 4 (IEEE802.11n) as the wireless communication technology. While it provided acceptable performance under certain conditions, its limitations in bandwidth and latency posed challenges for real-time remote control—especially in environments with dense traffic or signal interference. To address these issues and enhance system stability, the current study adopts Wi-Fi 5 (IEEE 802.11ac), which offers significantly higher throughput and better latency handling in medium-range industrial scenarios.

Recent research confirms that Wi-Fi 5 outperforms Wi-Fi 4 in industrial communication environments, demonstrating improved performance in terms of latency, reliability, and spectral efficiency under real-world conditions [12]. These advancements make Wi-Fi 5 a viable and effective solution for low-latency teleoperation systems.

The proposed system architecture consists of two onboard cameras—one mounted at the front and one at the rear of the vehicle—each capable of capturing 3.4 MP video at 60 frames per second. The video streams, along with telemetry data, are transmitted to the

HMI using Wi-Fi 5. The data is encapsulated as raw UDP packets and transmitted over three separate channels with distinct UDP ports: one for the front camera, one for the rear camera, and one for telemetry.

On the operator side, the HMI displays both camera feeds and telemetry in real-time. Commands from the operator are sent back to the vehicle over a dedicated UDP control channel. The operator controls the vehicle using a game controller: the left joystick is used for steering (range:  $-1.0$  to  $+1.0$ ), while the left and right trigger buttons correspond to throttle and brake inputs, respectively (range:  $0.0$  to  $2.0$ ). These control values are serialized and transmitted to the vehicle in a structured format.

By leveraging Wi-Fi 5, this communication structure ensures high-quality video streaming, reduced control latency, and improved responsiveness, which are all critical for effective and safe teleoperation. A detailed overview of this communication structure is illustrated in Figure 5.



**Figure 5.** Communication structure for the teleoperation system.

### 3.4. Server Implementation

Building upon our previous ADAS-assisted teleoperation system, the current server implementation extends the architecture to support **dual-mode teleoperation**, enabling operators to switch between **Remote Assistance Mode (ADAS-based)** and **Manual Mode (Direct Control)** in real-time. This enhancement was driven by the need for more granular control in complex or constrained driving scenarios, where high-level ADAS abstractions may fall short.

As in our prior work, the server facilitates bidirectional communication between the operator and the vehicle using ROS and UDP protocol. Incoming control signals from the Xbox controller are captured via SDL, serialized, and dispatched over UDP. A dedicated `udpListen()` thread continuously receives these signals, parses them, and queues them for mode-specific handling.

In **ADAS Mode**, the system functions identically to the previous implementation. The operator issues high-level commands (e.g., lane keeping, lane change, or speed adjustments), which are mapped to the appropriate ADAS features: LKA, LCA, and ACC. These features execute complex trajectory planning and speed regulation locally, freeing the operator from fine-grained stabilization and ensuring minimal control delay.

The key advancement lies in **Manual Mode**, which bypasses ADAS and gives the operator direct low-level authority over **steering, throttle, and braking**. Inputs from the Xbox controller are mapped to continuous ROS control messages using a dynamic scaling mechanism:

- **Throttle** values are scaled from the analog trigger range  $[-1.0, 1.0]$  to a capped acceleration of  $3.2 \text{ m/s}^2$ .
- **Brake** inputs are likewise mapped to a deceleration of  $-2.1 \text{ m/s}^2$ , with **brake commands prioritized** over throttle to ensure safety.
- **Steering** inputs from the joystick are translated into angular wheel commands within the range  $[-45^\circ, +45^\circ]$ , converted to radians and constrained to vehicle-specific limits.

This mapping ensures safe and bounded control, even in direct operation. Before any command is transmitted to the vehicle's actuators, a safety layer filters and limits the

values to predefined thresholds, preventing signal spikes or conflicting inputs. These sanitized commands are then published via ROS to topics responsible for **steering control** (e.g., *front wheel angle*) and **longitudinal control** (e.g., *acceleration request*).

The mapping and mode-switching procedure is summarized in *Algorithm 1: Dual-Mode Teleoperation Logic*, which outlines the sequential checks, safety validations, and ROS message publications for both ADAS and Manual Modes.

---

**Algorithm 1** Dual-Mode Teleoperation Logic

---

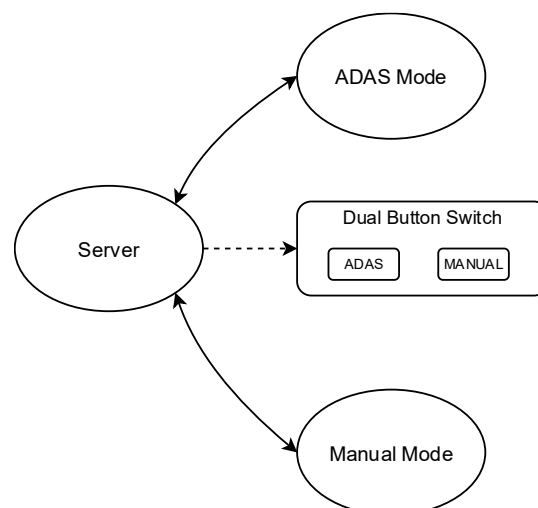
```

1: Class TeleoperationServer:
2:   Initialize ROS publishers for throttle, brake, steering, and ADAS commands
3:   Initialize mode state (ADAS or Manual)
4:
5:   Function udpListener():
6:     while True do
7:       Receive UDP message from client
8:       Update latest controller input array
9:     end while
10:
11:  Function processInputs():
12:    if mode switch button pressed then
13:      Toggle between ADAS Mode and Manual Mode
14:    end if
15:
16:    if current mode is ADAS then
17:      Publish ADAS command messages (e.g., ACC, LCA)
18:    else if current mode is Manual then
19:      Publish low-level throttle, brake, and steering commands
20:    end if
21:
22: Main Function:
23:   Initialize ROS node
24:   Start thread to run udpListener()
25:   while ROS is running do
26:     Call processInputs()
27:     Sleep for short duration
28:   end while

```

---

The server dynamically toggles between the two modes based on operator input, using a dual-button switch system. This allows seamless transitions between ADAS-guided navigation (ideal for structured highway driving) and manual control (required for dynamic, unstructured, or failure-prone environments), without interrupting the session. A high-level schematic of this dual-mode teleoperation structure is illustrated in Figure 6.





**Figure 6.** Dual-Mode Teleoperation Structure.

## 4. Tests and Results

The extended teleoperation system was evaluated on a 2014 Kia Niro hybrid equipped with a DriveKit interface, enabling direct control over the vehicle's steering, throttle, and braking systems. Tests were conducted in a controlled environment with no external traffic to ensure safety during manual actuation. The primary objective was to verify the effectiveness, responsiveness, and safety of the newly introduced Direct Control Mode, as well as to evaluate the reliability of dual-mode switching between Manual and ADAS assisted operations (see Figure 7 for the test vehicle and onboard equipment).



**Figure 7.** Test Vehicle and Onboard Equipment.

### 4.1. Setup and Methodology

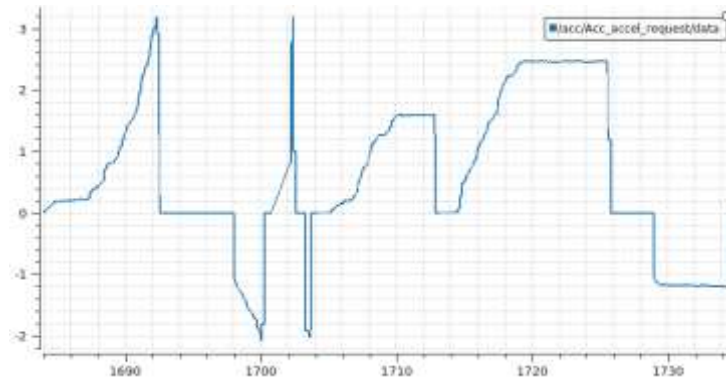
The Xbox game controller was connected via Bluetooth to the client machine, which transmitted control inputs to the vehicle-side server using UDP over a local Wi-Fi 5 network. The server application, running on a separate onboard machine, parsed incoming UDP packets and published control commands to ROS topics interfaced with the DriveKit. A web-based GUI was used throughout the testing to provide real-time feedback on vehicle telemetry and ADAS states.

Controller latency was measured from the moment a button or joystick was actuated to the time a corresponding response was visually detected on the vehicle (e.g., throttle application, brake light activation, steering motion). Several driving scenarios were tested at low speeds (<20 km/h) to validate manual input mapping and system responsiveness.

### 4.2. Manual Mode: Control Responsiveness

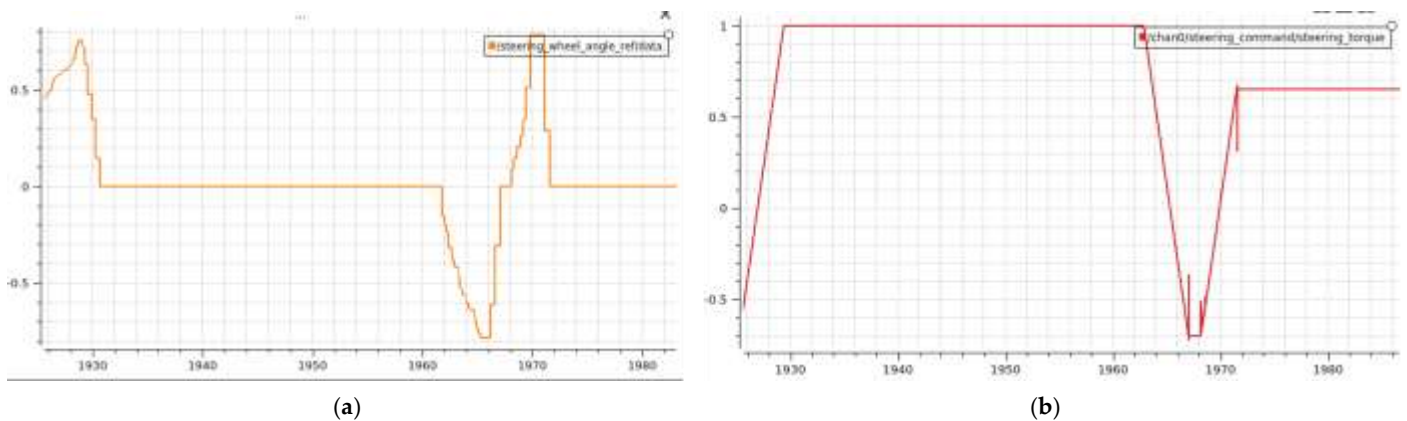
The vehicle responded to steering, throttle, and brake commands with minimal delay. In all tests, the total latency between controller input and vehicle actuation remained under 1 s, which is acceptable for low-speed remote driving. Control responsiveness was consistent across multiple sessions, confirming the system's stability.

For longitudinal control, acceleration and deceleration commands were scaled according to predefined safety limits. Throttle application resulted in smooth vehicle acceleration, while brake commands immediately overrode throttle input to bring the vehicle to a controlled stop—demonstrating the prioritization of safety in the input mapping. As illustrated in Figure 8, the acceleration request signal transmitted to the vehicle ECU followed the intended profile without sudden spikes, confirming proper filtering and scaling in the command pipeline



**Figure 8.** Acceleration request signal sent to the vehicle ECU.

For lateral control, joystick steering commands were scaled to fit the predefined  $[-45^\circ, +45^\circ]$  wheel angle range. As shown in Figure 9a, the steering wheel angle reference closely followed the intended trajectory without clipping or oscillations—demonstrating the stability and accuracy of the input mapping. Figure 9b presents the corresponding steering torque output from the vehicle ECU, which mirrored directional changes and confirmed precise command tracking with effective actuation.



**Figure 9.** Lateral control signal mapping and response: (a) steering wheel angle reference after scaling and safety filtering, showing smooth tracking of the intended profile; (b) steering torque output from the vehicle ECU, confirming accurate execution of directional commands without overshoot or oscillations.

#### 4.3. Mode Switching Performance

The system's **dual-mode operation** was tested by repeatedly switching between Remote Assistance Mode (ADAS-enabled) and Manual Mode during operation. The transition occurred smoothly and consistently, without requiring a restart or session reset. ADAS functionalities such as ACC and LKA were able to resume normal operation after returning from manual control.

- The mode switch logic prevented any unintended overlap of ADAS and manual commands.
- The transition latency was negligible, typically occurring within **100–200 ms** after the dedicated mode-switch buttons were activated on the controller.

#### 4.4. Safety Observations

- The brake command consistently and immediately overrode throttle input, providing a fail-safe mechanism.
- All direct control commands passed through a **sanitization layer**, ensuring they were within safe, predefined boundaries before being forwarded to the vehicle's actuators.
- No signal spikes, conflicting inputs, or unexpected vehicle behavior were recorded during the tests.

#### 4.5. Communication Reliability

Leveraging **Wi-Fi 5** significantly improved the communication link's stability compared to the previous Wi-Fi 4 setup. The UDP-based control messages experienced negligible packet loss during testing, and the average round-trip time for control packets remained under **80 ms** within a 90-m range.

## 5. Conclusions

This study presents an extended teleoperation architecture that builds upon prior ADAS-based remote assistance to incorporate a fully functional **Direct Control Mode**, enabling low-level actuation of throttle, brake, and steering via a game controller. The dual-mode design allows seamless transitions between high-level autonomous assistance and manual remote control, providing flexibility for a wide range of driving scenarios—from structured highway cruising to dynamic or failure-prone environments. A robust client-server model was developed using **UDP-based communication**, **SDL-based input capture**, and **ROS-integrated message handling**, ensuring low-latency and reliable command delivery. Direct control inputs are scaled, filtered, and safety-bounded before reaching the vehicle's actuators, preserving operational integrity while granting the operator precise authority over the vehicle. Extensive testing on a real vehicle platform validated the system's responsiveness, safety, and communication performance. Mode transitions occurred reliably with minimal delay, and direct commands produced predictable and stable behavior. The integration of **Wi-Fi 5** and **safety prioritization mechanisms** (such as brake override and input sanitization) further enhanced reliability and robustness. This work demonstrates the feasibility and advantages of **dual-mode teleoperation**, offering a hybrid solution that combines the stability of ADAS with the adaptability of manual intervention. It lays the groundwork for future deployments in remote support, testing, and recovery operations where human oversight remains essential alongside automated systems.

**Author Contributions:**

**Funding:**

**Institutional Review Board Statement:**

**Informed Consent Statement:**

**Data Availability Statement:**

**Conflicts of Interest:**

## References

1. Karaböcek, İ.; Kavak, B.; Özdemir, E. Remote Control of ADAS Features: A Teleoperation Approach to Mitigate Autonomous Driving Challenges. *Eng. Proc.* **2024**, *82*, 36. <https://doi.org/10.3390/ecs-a11-20449>.
2. Fallah, Z.; Shukla, V.K.; Khalid, M.N. Redefining Safety in Autonomous Vehicle Through Remote Teleoperation. In *Computational Intelligence in Pattern Recognition*; Springer: Singapore, 2022; pp. 211–222. [https://doi.org/10.1007/978-981-16-2543-5\\_19](https://doi.org/10.1007/978-981-16-2543-5_19).

3. Bouzón, I.; Pascual, J.; Costales, C.; Crespo, A.; Cima, C.; Melendi, D. Design, Implementation and Evaluation of an Immersive Teleoperation Interface for Human-Centered Autonomous Driving. *Sensors* **2025**, *25*, 4679. <https://doi.org/10.3390/s25154679>.
4. Georg, J.-M.; Feiler, J.; Hoffmann, S.; Diermeyer, F. Sensor and Actuator Latency during Teleoperation of Automated Vehicles. In Proceedings of the IEEE Intelligent Vehicles Symposium (IV), Paris, France, 9–12 June 2020; pp. 1–6. <https://doi.org/10.1109/IV47402.2020.9304802>.
5. Kettwich, C.; Schrank, A.; Oehl, M. Teleoperation of Highly Automated Vehicles in Public Transport: User-Centered Design of a Human-Machine Interface for Remote-Operation and Its Expert Usability Evaluation. *Multimodal Technol. Interact.* **2021**, *5*, 26. <https://doi.org/10.3390/mti5050026>.
6. Majstorovic, D.; Hoffmann, S.; Diermeyer, F. Trajectory Guidance: Enhanced Remote Driving of Highly-Automated Vehicles. In Proceedings of the 2024 IEEE Intelligent Vehicles Symposium (IV), Jeju Island, Republic of Korea, 2024; pp. 29–36. <https://doi.org/10.1109/IV55156.2024.10588536>.
7. Kizilkaya, B.; She, C.; Zhao, G.; Imran, M.A. Intelligent Mode-switching Framework for Teleoperation. In Proceedings of the 2024 IEEE International Conference on Robotics and Automation (ICRA), Yokohama, Japan, 2024; pp. 15692–15698. <https://doi.org/10.1109/ICRA57147.2024.10611333>.
8. 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* **2022**, *22*, 9119. <https://doi.org/10.3390/s22239119>.
9. Tanimoto, T.; Shinohara, K.; Yoshinada, H. Research on Effective Teleoperation of Construction Machinery Fusing Manual and Automatic Operation. *ROBOMECH J.* **2017**, *4*, 14. <https://doi.org/10.1186/s40648-017-0083-5>.
10. Wolf, M.-M.; Taupitz, R.; Diermeyer, F. Should Teleoperation Be Like Driving in a Car? Comparison of Teleoperation HMIs. In Proceedings of the 2024 IEEE Intelligent Vehicles Symposium (IV), Jeju Island, Republic of Korea, 2024; pp. 3304–3311. <https://doi.org/10.1109/IV55156.2024.10588688>.
11. Wolf, M.-M.; Schmidt, H.; Christl, M.; Fank, J.; Diermeyer, F. A User-Centered Teleoperation GUI for Automated Vehicles: Identifying and Evaluating Information Requirements for Remote Driving and Assistance. *Multimodal Technol. Interact.* **2025**, *9*, 78. <https://doi.org/10.3390/mti9080078>.
12. Bartolín-Arnau, L.M.; Orozco-Santos, F.; Sempere-Payá, V.; Silvestre-Blanes, J.; Albero-Albero, T.; Llacer-Garcia, D. Exploring the Potential of Wi-Fi in Industrial Environments: A Comparative Performance Analysis of IEEE 802.11 Standards. *Telecom* **2025**, *6*, 40. <https://doi.org/10.3390/telecom6020040>.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.