engineering proceedings

MDPI

Proceeding Paper

Remote Control of ADAS Features: A Teleoperation Approach to Mitigate Autonomous Driving Challenges †

İsa Karaböcek *, Batıkan Kavak and Ege Özdemir

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

* Correspondence: isa.karabocek@avl.com

† Presented at the 11th International Electronic Conference on Sensors and Applications (ECSA-11), 26–28 November 2024; Available online: https://sciforum.net/event/ecsa-11.

Abstract: This paper presents a novel approach to enhancing the safety of Advanced Driver Assistance Systems (ADAS) by integrating teleoperation for the remote control of ADAS features in a vehicle. The primary contribution of this research is the development and implementation of a teleoperation system that allows human operators to take control of the vehicle's ADAS features, enabling timely intervention in critical situations where autonomous functions may be insufficient. While the concept of teleoperation has been explored in the literature, with several implementations focused on the direct control of vehicles, there are relatively few examples of teleoperation systems designed specifically to utilize ADAS features. This research addresses this gap by exploring teleoperation as a supplementary mechanism that allows human intervention in critical driving situations, particularly where autonomous systems may encounter limitations. The teleoperation system was tested under two critical ADAS scenarios, cruise control and lane change assist, chosen for their importance in real-world driving conditions. These scenarios demonstrate how teleoperation can complement and enhance the performance of ADAS features. The experiments reveal the effectiveness of remote control in providing precise control, allowing for swift and accurate responses in scenarios where the autonomous system might face challenges. The novelty of this work lies in its application of teleoperation to ADAS features, offering a new perspective on how human intervention can enhance vehicle safety. The findings provide valuable insights into optimizing teleoperation for real-world driving scenarios. As a result of the experiments, it was demonstrated that integrating teleoperation with ADAS features offers a more reliable solution compared to standalone ADAS driving.

Keywords: teleoperation; remote assistance; ADAS

check for updates

Citation: 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/ecsa-11-20449

Academic Editor: Francisco Falcone

Published: 25 November 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In recent years, significant improvements have been witnessed in the ADAS field of the automotive industry. Designed to increase vehicle safety and improve driving efficiency, these systems include functions such as adaptive cruise control, lane keeping assistance, and lane change assistance. However, despite these advances, the need for human intervention continues in complex or unpredictable situations where such systems may struggle. This need has increased interest in teleoperation systems that allow a remote human operator to control the vehicle when necessary. Teleoperated driving can be leveraged as an enabling technology to smoothen the transition towards fully self-driving vehicles [1]. Although teleoperation shifts the concept of autonomy back to humans, it contributes to the development of automated systems [2].

Teleoperation is used as an umbrella term that comprises all functionalities needed to remotely support the operation of systems, including but not limited to autonomous vehicles, as discussed in the literature [3]. Many different categories of teleoperation exist; however, what is common among all is that, in each case, they require three characteristic

elements: a remote device, an operator interface (which can also be seen as the teleoperation workspace) and a communication link between these two. In this case, the remote device is the autonomous vehicle, the operator interface is a workspace from which an operator commands the vehicle using a remote controller and the communication is the transmission of sensor and control signals between the controller and the autonomous vehicle [4]. System latency is a crucial aspect of the communication element of teleoperation. Although there has been a drastic evolution in network technologies over time, with each iteration improving performance and reducing latency [5], the need to send and receive vehicle states and control signals in real time means that teleoperation will always be challenged by the issue of latency [6]. Many studies exist in the literature that try to measure the impact and minimize latency or that try to decouple it from the system.

In this study, a teleoperation system is introduced that allows human operators to manage a vehicle's ADAS features, enabling timely intervention in critical scenarios where autonomous systems may fall short. Since low-level vehicle commands are executed by the ADAS software stack, the impacts of issues that might arise from latency such as variable or low data transmission rates are minimized. The results demonstrate that the system can be utilized as a supplementary approach to enhance vehicle safety by providing a reliable fallback strategy. The remainder of this study is organized as follows: First, a literature review provides an overview of related work in teleoperation systems. Next, the details of the developed teleoperation system and its communication structure are presented, highlighting its architecture and functionality. This is followed by a discussion of the tests conducted and the results obtained, demonstrating the system's effectiveness. Finally, this study concludes with a summary of key findings and suggestions for future research directions.

2. Related Work

Teleoperation in autonomous systems has evolved significantly, focusing on human intervention for safety and reliability. Early systems used basic remote control, but advancements include real-time data integration. Fallah et al. [2] improved safety with robust systems using GPS and LiDAR data, enhancing vehicle algorithms with human decision-making. Zhang [7] proposed cloud-based teleoperation, leveraging 5G and AI for scalability despite latency and network issues. Georg et al. [8] investigated latency in sensor and actuator responses, offering insights into optimizing system performance. Lu et al. [9] explored teleoperation's role in complementing ADAS, with a framework for remote monitoring, assistance, and control, though HMI integration remains complex. Kettwich et al. [10] developed a user-centered HMI for automated service vehicles, achieving high satisfaction with camera views and interaction design. Their work aligns with that of Georg and Diermeyer [11], who enhanced HMI with adaptive camera systems. These studies highlight the ongoing evolution of teleoperation, addressing latency, HMI design, and cloud-based solutions, as well as signaling new opportunities and challenges for autonomous systems.

3. Teleoperation System Structure

Teleoperation systems vary based on the level of human control and intervention. The main types are remote control, supervised manipulation, and remote assistance. In remote control, the human operator directly commands the vehicle, handling all driving tasks such as steering and braking. This mode is crucial in challenging scenarios where full human control is needed, like complex or hazardous environments. Supervised manipulation involves a vehicle operating autonomously while a human supervises and can intervene if necessary. This approach is used when autonomy is generally reliable but occasional human oversight is required for exceptional cases or emergencies. Remote assistance provides high-level guidance rather than direct control. The operator offers instructions and makes decisions to assist the vehicle's automated systems, useful in uncertain situations like

construction zones or complex intersections, where a balance of human oversight and automation is beneficial.

In this paper's implementation, remote assistance is utilized to enhance ADAS functionality. The teleoperation system enables the operator to remotely manage ADAS features while benefiting from the safety measures inherent in these automated systems. The operator can intervene when ADAS encounters limitations or requires human judgment, such as in scenarios involving unpredictable traffic patterns or complex road conditions. This hybrid approach ensures that the vehicle operates autonomously under normal conditions, with ADAS handling routine tasks. Simultaneously, the teleoperation system serves as a supplementary mechanism, allowing the operator to take over for critical decision-making. The integration of teleoperation with ADAS reduces the operator's workload, improves response times, and ensures smooth transitions between manual and autonomous control.

The teleoperation system is based on a client–server architecture to enable remote vehicle control. The client, controlled by a human operator, uses a game controller and a web-based human–machine interface (HMI) to send control inputs (shown on the left side of Figure 1). These inputs are transmitted over the network using the UDP protocol to the server. The server, which is the test vehicle, processes these commands and converts them into actionable controls for the vehicle's ADAS. The server manages both longitudinal and lateral control, ensuring that throttle, brake, and steering inputs are accurately executed via the DriveKit (shown on the right side of Figure 1). ADAS features such as adaptive cruise control (ACC) and lane keeping assist (LKA) are continuously monitored, with the operator able to intervene when necessary. Additional details about the operator, server, and communication structure are provided in subsequent sections.

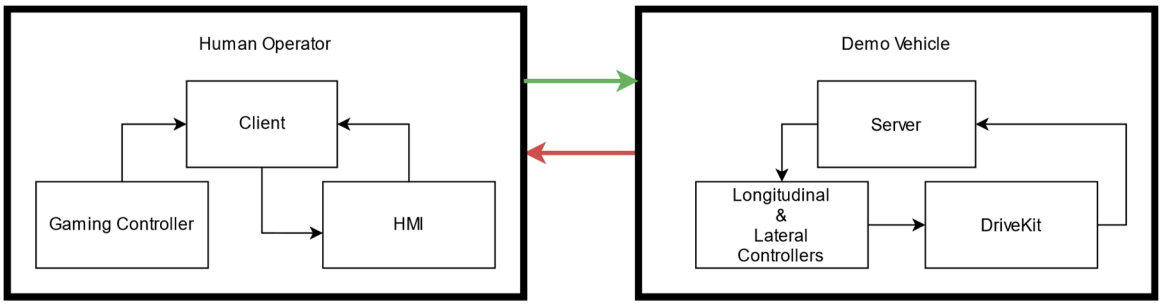


Figure 1. Remote assistance teleoperation structure.

3.1. Operator Control and Integration

In the developed teleoperation system, the operator controls the target vehicle via the host computer. The host computer runs an application that initializes the game controller and processes inputs from the operator. Bluetooth technology is used for communication between the controller and the host application. Bluetooth enables short-range wireless communication using low-power radio waves, providing a balance between convenience and energy efficiency. The latency between the controller and the host application over Bluetooth is approximately 10 ms. The controller functions on an event-driven basis: each time the operator presses a button, the current input status is immediately transmitted to the host application.

Controller Input Handling

The inputs sent by the operator via the controller are captured by a custom input handler. These inputs are then converted into a format suitable for transmission to the server via UDP. SDL (Simple DirectMedia Layer) 3.1.6 is used to facilitate communication between the controller and the host application. SDL is an open-source software library that offers cross-platform support and low-level access to hardware such as graphics, audio, keyboard, mouse, and joysticks, ensuring both portability and performance.

The game controller's inputs are shown in Figure 2 and have the following functionalities:

- ACC mode: Activation is carried out by the X button. We set the speed to increase and decrease using the Y and A buttons, respectively;
- Motion Planning mode: Lane change decisions are given via the LB and RB buttons.

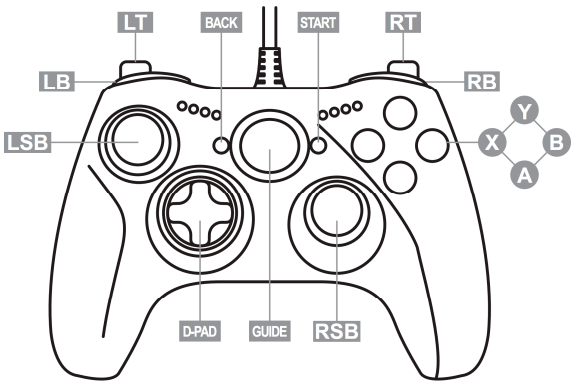


Figure 2. The button layout of the gaming controller.

3.2. HMI Interface

In addition to the game controller, the operator uses a human–machine interface (HMI), which provides real-time feedback on the vehicle's status, such as speed, active commands, and a live video feed from the vehicle's front camera (Figure 3). This interface is designed to give the operator a comprehensive view of the vehicle's current state, enhancing situational awareness and facilitating better decision-making.

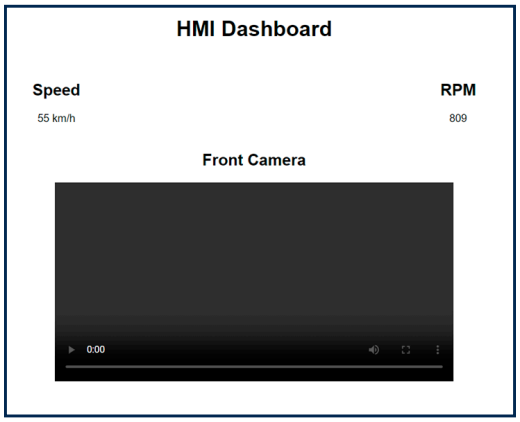


Figure 3. Human–machine interface.

The key features of the HMI include the following:

- Speed Monitoring: Displays the vehicle's real-time speed;
- Command Screen: Shows the currently active commands sent to the vehicle;
- Camera Feed: Provides a live video stream from the vehicle's front camera, enabling the operator to monitor the environment directly.

3.3. Communication Structure

Client–server communication is managed using the User Datagram Protocol (UDP). The UDP is ideal for teleoperation systems because of its low-latency transmission, making it faster than TCP for time-sensitive tasks. UDP's simplicity comes from its lack of extensive error-checking mechanisms or connection handshakes, which lowers the overhead but sacrifices reliability in favor of speed. This trade-off is acceptable in teleoperation, where real-time responsiveness is prioritized over guaranteed delivery.

The system handles continuous streams of data, such as controller inputs, sensor updates, and video feeds. UDP's stateless nature makes it especially suited for this type of real-time data transfer without needing session management.

Wi-Fi is the primary network protocol used for communication between the client and the server (Figure 4). It provides the necessary speed, low latency, and wireless flexibility for teleoperation. Wi-Fi's high bandwidth is especially useful for real-time video streaming and sensor data transmission, enabling smooth vehicle control. The wireless nature of Wi-Fi allows the system to operate without the need for physical cables, making it more versatile and mobile. However, Wi-Fi's wireless capabilities come with challenges like interference and network outages, which need to be considered. The built-in security protocols of Wi-Fi ensure the safety of transmitted data, making it reliable for teleoperation in various fields, such as mobile robots and the remote operation of hazardous environments.

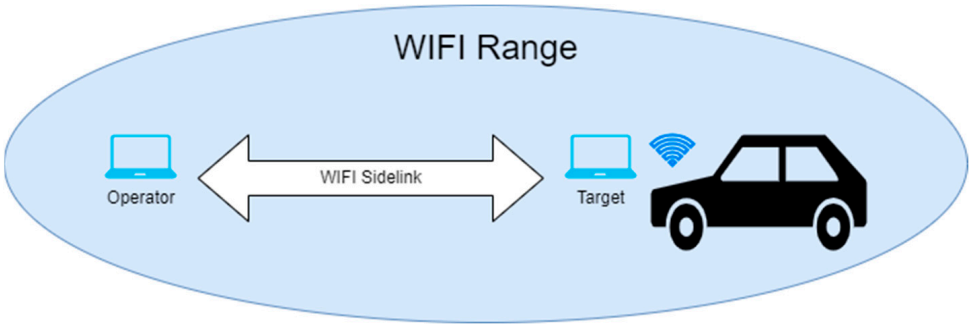


Figure 4. Communication structure.

3.4. Server Implementation

The demo vehicle is equipped with ADAS, including lane keeping assist (LKA) which helps to maintain the vehicle within the current lane; lane change assist (LCA), which assists with safe lane changes; and adaptive cruise control (ACC), which maintains a safe distance from the vehicle ahead by automatically adjusting the speed.

Once the operator takes over the task of decision-making, the high-level command regarding the action to take is sent to the vehicle. If a lateral movement is required, a lane change command or a lane following command is sent to the vehicle. This command is processed by the server structure, which acts as an intermediary, relaying the operator's input to the vehicle's ADAS and vice versa. When no specific command is issued, the vehicle remains in the LKA state, ensuring that it stays centered within its lane. However, when the teleoperator sends a lane change command, the server forwards this to the LCA system, which then initiates the lane change. The trajectory planning module generates an optimal trajectory towards the direction specified. This path is then further sent to the low-level lateral controller where a reference point among the trajectory is chosen and the required steering angle to track the chosen point is calculated, and the vehicle starts following this path. A schematic representation of this structure is also demonstrated in Figure 5. In this structure, the server's role is crucial in managing the flow of information and commands, allowing the teleoperation system to balance human input with automated assistance.

Similarly, when the vehicle is required to move longitudinally, a set speed value is sent to the vehicle. Based on the operator's decision, the vehicle can either keep a safe distance from other vehicles moving in front of it in traffic, or the perception module is over-ridden and the speed of the vehicle is set to a fixed value. Reaching the set speed point is the responsibility of the ACC module. Once the set speed value is received, this is sent to the low-level longitudinal controller of the vehicle where the necessary throttle and brake pedal inputs are calculated and consequently realized. In both cases, the operator is decoupled from the task of vehicle stabilization. This has the advantage of causing little or no significant amount of delay in the local control loop during task execution and, thus, no

danger of delay-caused instabilities [12]. Figure 6 shows the teleoperation structure with regard to longitudinal motion schematically.

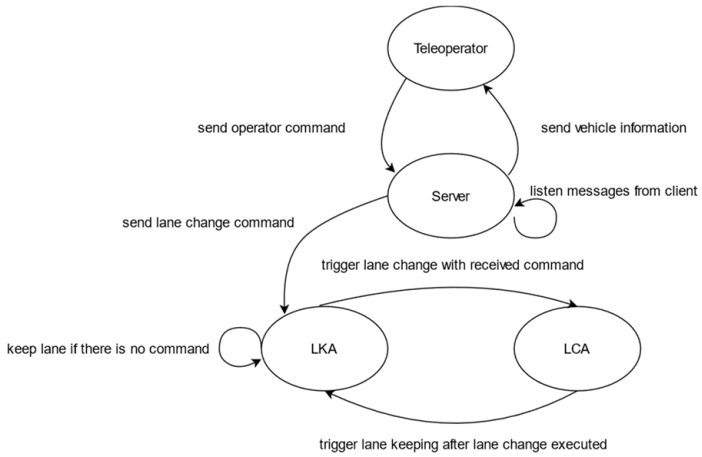


Figure 5. Lateral control state machine.

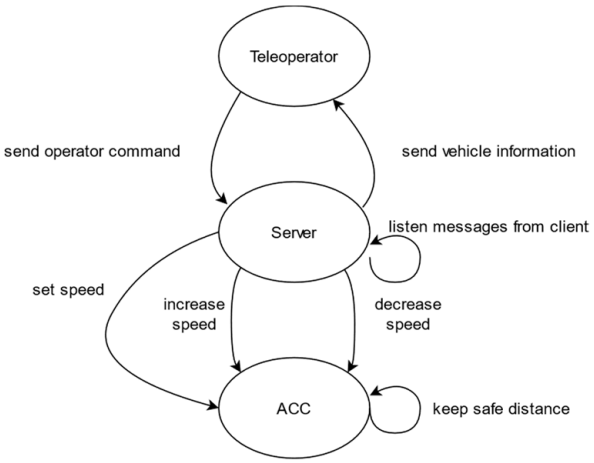


Figure 6. Longitudinal control state machine.

The implementation of the server side, which enables communication to occur between the controller and vehicle, has been carried out using the Robot Operating System (ROS) and the UDP. The pseudo-code of this algorithm is presented in Algorithm 1 below. The algorithm is structured around a central teleoperation class, which is responsible for handling both message reception from the teleoperation client and publishing relevant control commands to the vehicle. In the initialization phase, the algorithm sets up ROS publishers for key control topics such as lane change, speed settings, and control buttons (set and res). The udpListen() function is then launched in a separate thread to continuously listen for incoming UDP messages. This function creates and binds a UDP socket to a specified port, receiving control data from the client in the form of a float array. Each incoming message is parsed and stored as the latest control input, and a confirmation message is sent back to the client to acknowledge reception. The getLatestArray() function allows access to the most recent control data, while the processUdpMessage() function takes the received UDP message, verifies its size, and parses it into the appropriate ROS message fields (lane change, speed control, etc.). If the message contains sufficient control data, corresponding ROS messages are published to update the vehicle's control system.

Algorithm 1 Teleoperation Algorithm

```
Class Teleoperation:
Initialize ROS publishers for throttle, brake, lane change, set speed, set button, and res button topics
Function udpListen ():
Create UDP socket
Set socket options
Bind socket to specified port
while True do
        Receive UDP message
        Parse message into float array
        Update latest received array with new data
        Print received float array
        Send confirmation message to client
end while
    Function getLatestArray ():
Return latest received array
     Function processUdpMessage(message):
if message size is at least length of control inputs then
         Parse message into relevant ROS message fields
         Publish ROS messages for lane change, set button, set/increase/decrease speed
end if
Main Function:
Initialize ROS node" teleoperation node"
Create instance of Teleoperation class
Start new thread to run udpListen function
while True do
         Retrieve latest received array
         Process received array to publish relevant ROS messages
         Sleep for specified duration
end while
Call ros::spin ()
```

4. Tests and Results

The test vehicle used for this study was a 2014 Kia Niro hybrid, which was modified with a DriveKit to enable the reading and writing of CAN messages. It was equipped with custom lateral and longitudinal controllers that provide ACC, LKA, and LCA capabilities. The vehicle communicated with the control systems via the ROS framework, which interfaced with the DriveKit for efficient message handling. All tests were conducted in a controlled area closed to traffic to ensure safety and accuracy. The teleoperation tests between the client and server were conducted within the range of the vehicle's Wi-Fi router, with a maximum tested range of 90 m. The testing focused on two primary scenarios, longitudinal control and lateral control, which will be elaborated in the subsections of this section. The goal of these tests was to evaluate the performance and reliability of the teleoperation system, with expected results including smooth vehicle control during teleoperation and stable communication within the tested range. The test equipment and onboard hardware are shown in Figure 7.

While conducting the tests, it has been observed that since the reception of signals on the server side follows a queue structure, when any request is sent too rapidly from the controller, the latency of processing this request also increases. The average amount of time delays caused by this issue is illustrated in Figure 8a below. Since the low-level actuation is performed by the ADAS system, the only latency of significance is the latency associated with sending the high-level decision command from the controller to the vehicle. Additionally, to evaluate the operational limits of the system, tests were conducted by sending teleoperation signals from various distances. Figure 8b illustrates the correlation between the packet losses and the increasing distances from the vehicle, highlighting the

critical thresholds for effective communication. Here, the success rate corresponds to the number of packets successfully received by the server relative to the total number of packets sent by the client, and as can be seen, its value decreases drastically at about 90 m away from the vehicle.



Figure 7. Test vehicle and onboard equipment.

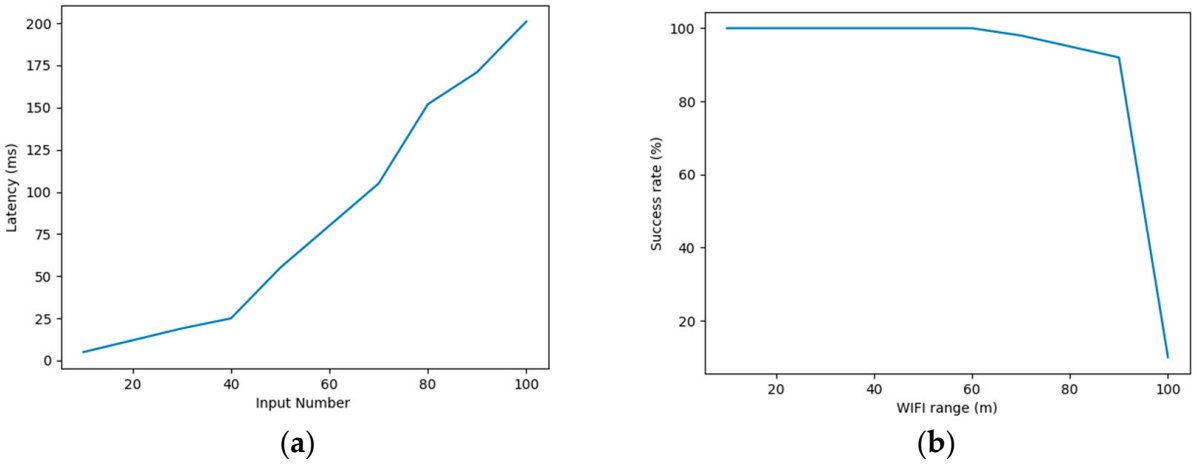


Figure 8. Latency as a function of number of inputs sent to the client (**a**); the success rate of teleoperation signals relative to the distance from the vehicle (**b**).

4.1. Longitudinal Test Scenario

In this scenario, the vehicle speed has manually been increased to 30 km/h. At this point, the command to activate the ACC is remotely sent to the vehicle via the controller. Once this signal is received, the vehicle continues to drive at its current speed. After this stage, signals to increase or decrease the current set speed are also sent via the controller and the ACC adjusts the required acceleration of the vehicle based on this input. Changes in the state of the set speed increase button of the ACC are shown in Figure 9 below. Since this input has a button logic, it returns to the default state right after a signal is received from the controller. The ADAS system response is shown in Figure 10.

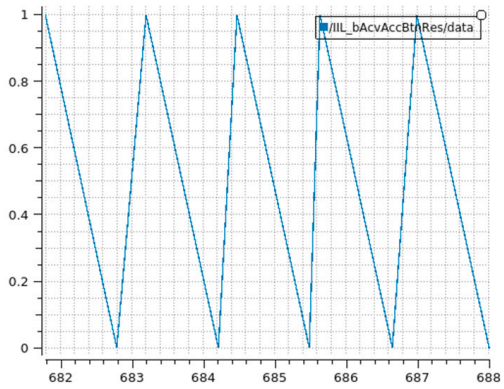
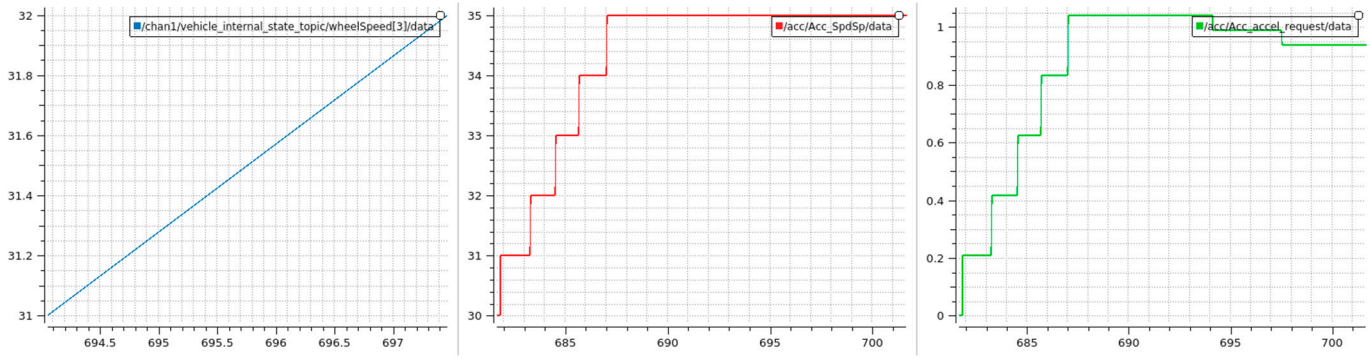


Figure 9. The response of the ACC set speed increase button to the controller.

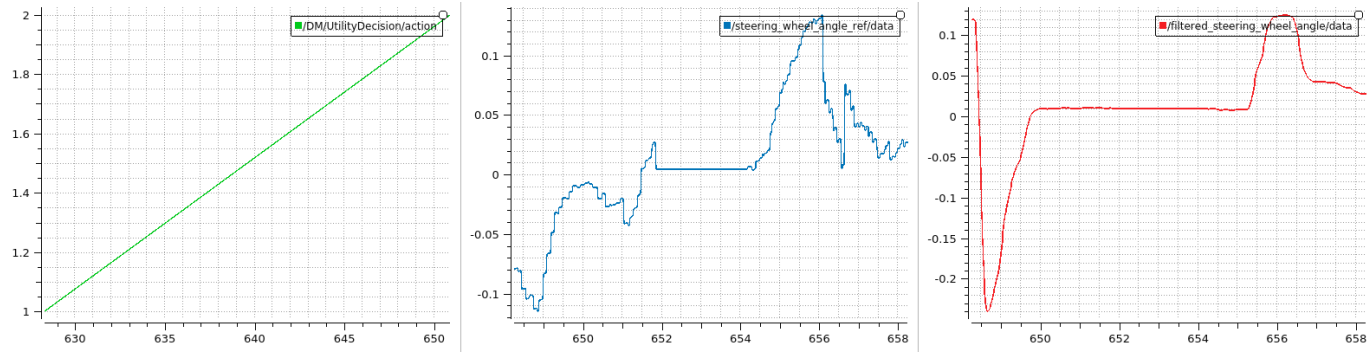


(**a**) (**b**) (**c**)

Figure 10. Change in vehicle speed (**a**), increase in ACC set speed value (**b**) and change in ACC acceleration request as a command to increase the set speed are sent via the controller (**c**).

4.2. Lateral Test Scenario

In the lateral test scenario, consecutive left and right lane change decisions are transmitted to the vehicle via the controller in combination with the longitudinal commands illustrated in the longitudinal test scenario. In each case, the vehicle successfully generates the lane change trajectory and manages to follow the reference path. The changes in the decision maker module states when a right lane change command is issued by the controller are shown in Figure 11a. This module is the software component of the vehicle responsible for deciding in which direction the lane change should occur. The values of 0, 1 and 2 correspond to left lane change, lane keep and right lane change decisions, respectfully. The generated steering wheel angle reference and response are also shown in Figure 11b,c.



(**a**) (**b**) (**c**)

Figure 11. Right lane change request received by controller (**a**), steering angle reference (**b**), and actual steering response (**c**).

5. Conclusions

This research demonstrates the successful integration of teleoperation into ADAS to enhance vehicle safety and provide a means for human intervention in critical driving situations. By utilizing a remote assistance model, the teleoperation system allows a human operator to seamlessly control ADAS features such as ACC and LKA, complementing autonomous vehicle capabilities when necessary. The client–server architecture, where commands from the human operator are transmitted via a game controller and web interface, ensures reliable control over both longitudinal and lateral vehicle systems, while the ADAS features maintain safe and efficient driving. The results from testing this system, conducted on an autonomous test vehicle, confirm that the teleoperation system can maintain smooth vehicle control and stable communication within a 90 m range. This study highlights the potential of teleoperation to enhance the reliability and flexibility of ADAS, offering a more robust safety solution compared to standalone autonomous driving systems. The findings emphasize the importance of human intervention in complex driving scenarios, providing valuable insights for future developments in vehicle teleoperation and ADAS integration.

Author Contributions: Conceptualization, İ.K., B.K. and E.Ö.; methodology, İ.K. and B.K.; software, İ.K., B.K. and E.Ö.; validation, İ.K., B.K. and E.Ö.; investigation, İ.K., B.K. and E.Ö.; supervision, İ.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing does not apply to this paper.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Schimpe, A.; Diermeyer, F. Steer with me: A predictive, potential field-based control approach for semi-autonomous, teleoperated road vehicles. In Proceedings of the 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), Rhodes, Greece, 20–23 September 2020; pp. 1–6. [CrossRef]
2. Fallah, Z.; Shukla, V.K.; Khalid, M.N. Redefining safety in autonomous vehicle through remote teleoperation. In *Computational Intelligence in Pattern Recognition*; Springer: Cham, Switzerland, 2022; pp. 219–230.
3. Majstorovic, D.; Hoffmann, S.; Pfab, F.; Schimpe, A.; Wolf, M.M.; Diermeyer, F. Survey on teleoperation concepts for Automated Vehicles. In Proceedings of the 2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Prague, Czech Republic, 9–12 October 2022. [CrossRef]
4. Winfield, A.F.T. Future Directions in Tele-operated Robotics. In *Telerobotic Applications*; Professional Engineering Publishing: London, UK, 2000; pp. 147–163.
5. Kamtam, S.B.; Lu, Q.; Bouali, F.; Haas, O.C.; Birrell, S. Network latency in teleoperation of connected and autonomous vehicles: A review of trends, challenges, and mitigation strategies. *Sensors* **2024**, *24*, 3957. [CrossRef]
6. Neumeier, S.; Wintersberger, P.; Frison, A.K.; Becher, A.; Facchi, C.; Riener, A. Teleoperation: The holy grail to solve problems of automated driving? Sure, but latency matters. In Proceedings of the 11th International Conference on Automotive User Interfaces and Interactive Vehicular Applications, Utrecht, The Netherlands, 21–25 September 2019. [CrossRef]
7. Zhang, T. Toward automated vehicle teleoperation: Vision, opportunities, and challenges. *IEEE Internet Things J.* **2020**, *7*, 11347–11354. [CrossRef]
8. Georg, J.-M.; Feiler, J.; Hoffmann, S.; Diermeyer, F. Sensor and actuator latency during teleoperation of automated vehicles. In Proceedings of the 2020 IEEE Intelligent Vehicles Symposium (IV), Las Vegas, NV, USA, 19 October–13 November 2020. [CrossRef]
9. Lu, S.; Zhong, R.; Shi, W. Teleoperation technologies for enhancing connected and autonomous vehicles. In Proceedings of the 2022 IEEE 19th International Conference on Mobile Ad Hoc and Smart Systems (MASS), Denver, CO, USA, 20–22 October 2022; Volume 13, pp. 435–443. [CrossRef]
10. 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. [CrossRef]

11. Georg, J.-M.; Diermeyer, F. An adaptable and immersive real-time interface for resolving system limitations of automated vehicles with teleoperation. In Proceedings of the 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC), Bari, Italy, 6–9 October 2019; Volume 12, pp. 2659–2664. [CrossRef]
12. Sheridan, T. Space teleoperation through time delay: Review and prognosis. *IEEE Trans. Robot. Autom.* **1993**, *9*, 592–606. [CrossRef]

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.