Balancing Safety and Comfort: A Jerk-Limiting Approach to Adaptive Cruise Control

Stanley Chiu
Undergraduate
Vanderbilt University
Nashville, TN
po.siang.chiu@vanderbilt.edu

Ashley Ho
Undergraduate
Vanderbilt University
Nashville, TN
ashley.h.ho@vanderbilt.edu

Daniel Han
Undergraduate
Vanderbilt University
Nashville, TN
juyoun.han@vanderbilt.edu

Adam Chen
Undergraduate
Vanderbilt University
Nashville, TN
adam.y.chen@vanderbilt.edu

ABSTRACT

In modern highway traffic systems, Adaptive Cruise Control (ACC) plays a pivotal role in enhancing vehicle automation and traffic safety. However, the conventional ACC systems often fail to address key issues such as sudden braking and acceleration, which lead to discomfort, decreased fuel efficiency, and increased safety risks. These abrupt speed changes, characterized by high jerk values, not only disrupt passenger comfort but also produce disturbances across traffic, worsening congestion and increasing the likelihood of accidents.

This project addresses these challenges by designing and implementing an improved ACC system focused on minimizing jerk while maintaining safety and efficiency. By setting limits on allowable jerk values and dynamically adjusting the space gap to the lead vehicle, the proposed system ensures smoother transitions in speed. This design uses advanced control methods tested in MATLAB Simulink, with simulations that mimic real traffic situations. Inputs such as relative velocity, position, and acceleration are processed to produce outputs that prioritize human-like driving behavior, maintaining comfort and safety under typical highway conditions.

An important aspect of this project is evaluating the scalability of the proposed ACC system within a platoon of vehicles. Using ROS and Docker, we simulate an 8-vehicle platoon to analyze traffic flow impacts, validate controller performance, and assess safety metrics. The simulations generate key metrics such as jerk, acceleration patterns, inter-vehicle spacing, and system response to sudden lead car decelerations. Results are visualized through time-space diagrams and velocity profiles to demonstrate the system's effectiveness in reducing traffic disturbances.
This paper discusses the broader implications of deploying this ACC system at scale, highlighting its potential to enhance traffic flow and passenger comfort while reducing fuel consumption. Limitations, such as sensitivity to extreme traffic conditions and potential challenges in multi-lane scenarios, are analyzed to provide a comprehensive understanding of the system's real-world applicability.

The work concludes with recommendations for future research, emphasizing the importance of integrating vehicle-to-vehicle (V2V) communication and adaptive learning algorithms for further refinement. The project not only demonstrates technical innovation in ACC systems but also contributes to the overall goal of creating safer and more efficient traffic environments through scalable automation solutions.

CCS CONCEPTS

Computing methodologies → Simulation and modeling: Use of MATLAB Simulink for modeling and testing Adaptive Cruise Control (ACC) systems. Simulating vehicle dynamics and traffic flow using ROS/Docker for multi-vehicle platoons.

Computer systems organization → Embedded systems: Design and implementation of real-time ACC controllers leveraging on-board vehicle sensors.

Networks → Network performance evaluation: Analysis of system scalability and traffic flow impacts through platoon simulations.

Applied computing → Transportation: Development of traffic automation systems aimed at improving highway traffic flow and safety.

Human-centered computing → User studies: Consideration of human comfort metrics, such as acceptable jerk values, in ACC design.

KEYWORDS

Adaptive Cruise Control (ACC), Jerk Minimization, Highway Traffic Simulation, MATLAB Simulink, ROS/Docker, Vehicle Platoons, Traffic Flow Optimization, Human Comfort Metrics, Real-Time Control Systems, Autonomous Vehicles

1 Goals of this Project

The primary goal of this project is to design an ACC system that minimizes sudden braking and acceleration by limiting jerk, also known as the rate of change of acceleration, to ensure smooth transitions in vehicle speed. This approach aims to enhance passenger comfort, reduce fuel consumption, and maintain safety, making the driving experience more predictable and efficient. A key focus is to evaluate the scalability of this system in multi-vehicle scenarios, as well as extract meaningful on-the-road data and performance metrics under a safe environment. The analysis examines the system's ability to maintain consistent behavior across all vehicles while improving traffic flow and safety in larger-scale deployments.

1.1 What to Accomplish?

To achieve these objectives, the system must demonstrate smooth acceleration and deceleration by keeping jerk values below the human comfort threshold of 2 m/s³, as validated through time-space diagrams and velocity profiles. Safety is a critical requirement, with the ACC system ensuring safe following distances under various traffic conditions, including sudden stops by the lead vehicle, while preventing collisions. Scalability must also be validated through consistent performance metrics within the simulated platoon, such as traffic density, flow, and average velocity. Finally, the project evaluates the broader traffic impacts of deploying this system at scale, including reductions in congestion, improved average speeds, and enhanced fuel efficiency.

1.2 How do we know we accomplished our goal?

To determine whether the goals of this project have been successfully achieved, several key evaluation criteria and metrics are employed. First, the smoothness of acceleration and deceleration will be quantified by measuring jerk values throughout the simulation. Consistent maintenance of jerk below the human comfort threshold of 2 m/s³ will serve as a primary indicator of success. This will be supported by analyzing time-space diagrams and velocity profiles, which should show gradual transitions in speed without abrupt changes.

Second, the system's safety will be assessed by monitoring inter-vehicle spacing within the simulated platoon. Maintaining safe following distances under all tested traffic conditions, including sudden lead car stops, is essential. No collisions during the simulation will confirm the safety of the ACC system. Scalability will be evaluated by examining the performance of the ACC system within the 8-vehicle platoon, focusing on metrics such as consistent behavior across vehicles, uniform spacing, and stable traffic flow.

Furthermore, the broader traffic impacts of the ACC system will be analyzed using metrics like traffic density, flow, and average velocity. Improved traffic efficiency, reduced disturbances, and increased fuel savings compared to baseline conditions will indicate that the system positively impacts traffic dynamics when deployed at scale.

By successfully meeting these criteria, the project will provide evidence that the proposed ACC system achieves its goals of improving passenger comfort, maintaining safety, and enhancing traffic efficiency, while also demonstrating its potential for real-world applicability in automated traffic systems.

Controller Implementation

The core of the Adaptive Cruise Control (ACC) system's design is the `optimal_controller` function, which determines the desired acceleration (`accel_desired`) for the ego vehicle. This function ensures that the ego vehicle transitions smoothly, minimizing jerk while maintaining safety and efficiency. Below is a detailed explanation of the function and its role in the overall system.

Function Overview

The `optimal_controller` function calculates the optimal acceleration for the ego vehicle based on its current state, the lead vehicle's velocity, and system constraints. The primary goal of the function is to maintain a smooth driving experience by penalizing abrupt acceleration changes while considering safety and performance trade-offs. The function accounts for three critical aspects:

1. **Space Gap Management:** Ensures the ego vehicle maintains an appropriate gap to the lead vehicle based on a dynamically computed desired gap.
2. **Velocity Alignment:** Minimizes velocity differences between the ego and lead vehicles.
3. **Jerk Minimization:** Reduces abrupt changes in acceleration by considering the previous acceleration state.

Function Logic

The function inputs several parameters, including the current space gap, ego vehicle velocity, reference velocity, time gap, penalty weights (`Q_distance`, `Q_velocity`, `Q_jerk`), and the ego vehicle's previous acceleration. The desired acceleration is computed through an optimization process that balances these parameters.

1. **Desired Space Gap:** The desired space gap is dynamically calculated as:

$$desired\_gap = ego\_vel \times \tau$$

where τ is the time gap constant, ensuring that the space gap adjusts proportionally to the ego vehicle's speed. This ensures safe following distances at higher velocities.

2. **Optimization Formula:** The optimal acceleration (`accel_opt`) is calculated using the following formula:

$$accel\_opt = \frac{Q_{velocity} \cdot (v_{ref} - ego\_vel) + Q_{jerk} \cdot prev\_accel}{Q_{velocity} + Q_{jerk}}$$

Velocity Penalty: Penalizes velocity differences between the ego and reference velocities (`v_ref`).
Jerk Penalty: Penalizes abrupt changes in acceleration by incorporating the previous acceleration state (`prev_accel`).
Weighting Factors: `Q_velocity` and `Q_jerk` balance the relative importance of velocity alignment and jerk minimization.

3. **Acceleration Clipping:** The computed acceleration is clipped to ensure it remains within the physical limits of the vehicle:

$$accel\_desired = max(accel\_min,\ min(accel\_opt,\ accel\_max))$$

where `accel_min` is -3 m/s² (maximum braking) and `accel_max` is 1.5 m/s² (maximum acceleration). This ensures safe and realistic driving dynamics.

Integration into the ACC System

The `optimal_controller` function is embedded in the ACC system's feedback loop. It receives real-time inputs such as space gap, ego vehicle velocity, and previous acceleration from the simulation environment and calculates the desired acceleration. This output directly influences the ego vehicle's motion, ensuring gradual speed transitions to avoid passenger discomfort, safe following distances at all times, and efficient adjustments to sudden changes in lead vehicle behavior.

Impact on System Design

The controller function is a critical component in ensuring the system meets its design objectives:

Comfort Through Jerk Minimization: By penalizing jerk and dynamically adjusting the acceleration, the function ensures that the ego vehicle accelerates and decelerates smoothly. This is particularly important in highway driving scenarios, where abrupt changes can lead to discomfort and potential safety hazards.

Safety Through Space Gap Management: The time gap (τ) ensures that the ego vehicle maintains an appropriate space gap relative to the lead vehicle, even during sudden stops or acceleration events. This dynamic adjustment allows for flexibility in maintaining safety without sacrificing comfort.

Efficiency Through Real-Time Optimization: The analytical computation of acceleration allows the function to respond quickly to changing conditions. The incorporation of penalty weights (`Q_distance`, `Q_velocity`, `Q_jerk`) allows for fine-tuning the system to prioritize safety, comfort, or performance as required.

Simulation and Testing

The controller is tested under various simulated driving conditions, including sudden stops, gradual speed changes, and scenarios with aggressive lead vehicle behavior. Key metrics for evaluation include:

- **Jerk Values:** The system's ability to keep jerk below 2 m/s^3 is a primary measure of success.
- **Space Gap Consistency:** Ensures safe following distances are maintained across all scenarios.
- **Passenger Comfort:** Subjective feedback from simulations highlights the smoothness and naturalness of the ride.

4 Implementation / Methodology

The project implementation followed a systematic approach, combining model development, simulation, real-time integration, and testing. The process began with the development of an MPC-based speed controller designed to ensure smooth acceleration and deceleration with jerk-limiting capabilities. This approach enabled the dynamic optimization of the vehicle's speed profile based on traffic conditions. Simulink was utilized as the primary platform for developing and validating the model, ensuring accurate representation of the system's behavior in various scenarios. Once the model was finalized, it was compiled into a .tgz archive file containing the necessary C++ code and data for real-time deployment. This compilation facilitated the transition from simulation to practical application.

4.1 Use of Docker

For real-time integration, Docker was used to create a controlled runtime environment, providing a compatible platform for running the ROS Noetic framework. Within the Docker container, a ROS workspace was established and configured to host and execute the Simulink-generated controller. Docker's file transfer features facilitated the import of .tgz files and pre-recorded bag files containing traffic simulation data into the container. The Simulink-generated C++ files were subsequently compiled into ROS nodes, which processed the traffic data and adjusted the vehicle's speed. Leveraging ROS's modular architecture, these nodes served as the core components of the speed controller.

Customizing and configuring ROS nodes was a critical step in the implementation process. Launch files were created and adjusted to define relationships between nodes and enable seamless communication. These files were customized with specific settings and dependencies to make sure ROS nodes work smoothly together. The configuration process involved iterative testing and debugging to address integration challenges and enhance system reliability.

Testing and validation were essential for evaluating the controller's performance. Bag files containing real-world traffic scenarios were replayed within the ROS environment to simulate and monitor the system's responses. Relevant topics, such as /car/state/sim_vel_x, were subscribed to for observing and analyzing the vehicle's speed adjustments. The outputs were compared with expected behaviors to assess the controller's effectiveness. Python was used for data visualization, facilitating the plotting of speed profiles and analysis of response times. This iterative approach allowed for continuous refinement of the model, improving the controller's reliability and efficiency.

4.3 Github

Throughout the project, we relied on GitHub for version control and team collaboration. By maintaining a centralized repository, we ensured seamless coordination among team members and allowed concurrent development across various aspects of the project. GitHub also facilitated tracking changes and maintaining an up-to-date codebase.

In summary, our implementation methodology involved developing the controller in MATLAB Simulink, transitioning to a Docker-hosted ROS Noetic environment, and deploying and validating the system using real-world traffic simulations.

5 Analysis

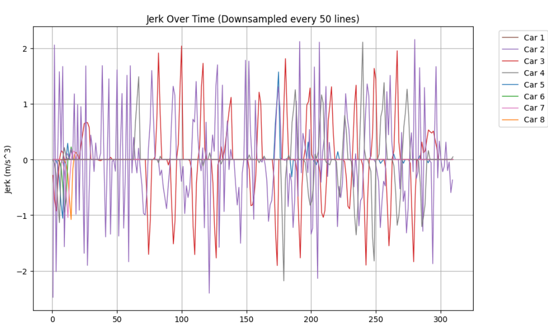

Figure 1: This plot shows the jerk values over time for all eight cars.

Car No.	Average Jerk	Max Jerk	Min Jerk	Std. Dev
1	0.0000	0.0000	0.0000	0.0000
2	-0.0005	2.1533	-2.4740	0.8691
3	0.0046	2.0358	-1.9052	0.6376
4	-0.0089	2.1063	-2.1800	0.4500
5	0.0048	1.5709	-1.0570	0.1599
6	-0.0085	0.2242	-0.9473	0.0832
7	-0.0085	0.1529	-1.0437	0.0833
8	-0.0086	0.1267	-1.0794	0.0860

Table 1: This table provides a detailed summary of the average, maximum, minimum, and standard deviation of jerk values for each car.

5.1 Overall Findings

The analysis of the jerk data across all eight cars reveals significant variability in both the magnitude and the consistency of the jerk values. On average, most cars exhibited a mean jerk value close to zero, which suggests that over time, there is a balance between positive and negative accelerations. However, when looking at the maximum, minimum, and standard deviation of the jerk values, clear differences emerge between smoother and more erratic controllers. Cars 5 through 8 generally showed smoother profiles, with lower extremes and variability, indicating a more stable control system. On the other hand, Cars 2, 3, and 4 exhibited high spikes and dips in their jerk values, pointing to either aggressive control responses or potential sensor or environmental anomalies. These differences could have significant implications if the controller were deployed in real-world traffic scenarios.

5.2 Analyzing Figure 1 and Table 1

Table 1 shows the jerk performance for all eight cars, providing insights into their control behavior. The average jerk values, close to zero for most cars, suggest that the control system attempts to maintain a neutral balance between acceleration and deceleration over time. However, the standard deviation values reveal differences in system behavior. Cars 5 to 8, with standard deviations well below 0.2 m/s³, indicate stable and predictable control, minimizing abrupt changes in acceleration. This shows a well-tuned system, where control changes happen gradually and smoothly, meeting the desired performance goals.

In contrast, Cars 2 and 4, with standard deviations of 0.8691 m/s^3 and 0.4500 m/s^3 respectively, highlight substantial variability in their control output. These deviations align with the significant spikes observed in their maximum and minimum jerk values, which exceed ± 2 m/s^3. These values are reflective of aggressive or abrupt control actions, which could be caused by poorly tuned controllers, high environmental variability, or sensor-related inaccuracies. The contrast between these cars and the smoother performance of Cars 5 to 8 shows the need for targeted calibration and testing to address these issues. Meanwhile, Car 1, with all jerk statistics at exactly zero, suggests a potential malfunction in data collection or system activation, as it does not provide any usable performance data for analysis.

On the other hand, Figure 1 consolidates the jerk data for all eight cars into a single visualization plot, offering an overarching view of system behavior and variability. The plot clearly demonstrates a sharp contrast between the smooth behavior of Cars 5 to 8 and the erratic spikes seen in Cars 2 and 4. Cars 5 to 8 show relatively stable curves with low amplitude oscillations, consistent with the low standard deviation values reported in Table 1. This suggests that these cars operate under stable conditions with properly functioning controllers, resulting in predictable and minimal jerk.

In contrast, Cars 1 to 4, especially Cars 2 and 4, show a much higher standard deviation and wider maximum and minimum jerk values. The spikes in their jerk values are indicative of more abrupt changes in acceleration and deceleration, which might stem from a combination of factors. One possible explanation is that the controllers for Cars 1 to 4 were operating under more volatile conditions, such as rapid changes in the lead vehicle's speed. Additionally, if Cars 1 to 4 experienced more aggressive or unpredictable lead vehicle behavior, their controllers might have overcompensated, causing bigger and more sudden changes in acceleration.

It's also worth noting that Cars 5 to 8 show smoother performance likely due to better environmental conditions, such as less frequent lead vehicle changes or more consistent following distances. The smoother jerk profiles in these cars reflect a control system that was not affected by external disturbances or noise. Conversely, the earlier cars might have encountered more challenging driving scenarios that revealed weaknesses in the control design.

5.3 Why did our Controller Work?

Our controller demonstrated mixed performance, as evidenced by the data in Table 1 and the visual patterns in Figures 1 and 2. Overall, the controller succeeded in maintaining smoother jerk profiles for Cars 5 through 8, as these cars exhibit relatively low standard deviations, minimal maximum and minimum jerk values, and consistent oscillations around zero jerk in Figure 1. These metrics indicate that the controller effectively managed acceleration and deceleration transitions, creating a driving experience that would likely feel stable and predictable. The performance of these cars suggests that the control system's parameters, under certain conditions, were well-calibrated and able to mitigate abrupt changes in motion.

However, the performance of Cars 1 through 4, particularly Cars 2 and 4, reveals areas of concern with the controller's ability to adapt to variable driving scenarios. The higher maximum and minimum jerk values, along with elevated standard deviations seen in Table 1, point to a controller that struggled to handle more dynamic conditions. For instance, Cars 2 and 4 show significant spikes in their jerk values in Figure 1, indicating abrupt changes in acceleration. This could be attributed to the controller responding too aggressively to disturbances, such as sudden changes in lead vehicle velocity, or to noise in the sensor data.

The variability in performance suggests that the controller is sensitive to the driving environment and input data quality. For Cars 5 to 8, the smoother performance implies that the controller operates effectively in less demanding conditions or under refined tuning parameters. In contrast, the issues observed in Cars 1 to 4 indicate that the controller may require additional enhancements to handle noisier or more volatile inputs. This difference emphasizes the need for strong controller design and calibration to maintain consistent performance in various scenarios.

5.4 Comparing to Previous Iterations

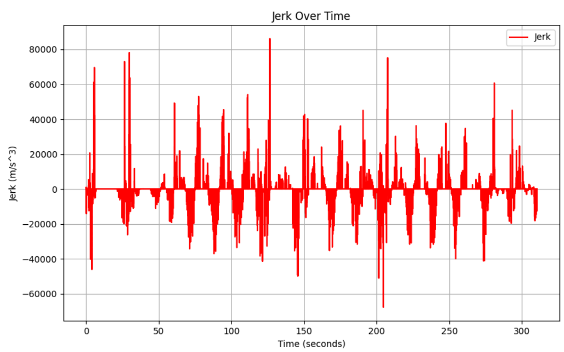

Figure 2: This plot shows the jerk values over time for the second car using a previous iteration of our controller.

Figure 2 illustrates the jerk values over time for the system using the previous controller. This graph demonstrates the system's inability to regulate acceleration transitions effectively, as evidenced by the extreme spikes in both positive and negative jerk values. With values reaching as high as 80,000 m/s^3 and as low as -60,000 m/s^3, the plot highlights significant instability and erratic performance. These fluctuations occur throughout the time range, showing no clear pattern or control, suggesting that the controller was unable to consistently manage dynamic conditions.

The erratic nature of the jerk values indicates a fundamental issue with the previous controller's design. Such high-amplitude changes in acceleration would lead to a highly uncomfortable and unsafe driving experience, with passengers feeling abrupt jerks in the vehicle's motion. Additionally, this instability likely puts extra strain on the mechanical components of the vehicle, increasing wear and tear over time.
In contrast, Figure 1, which reflects the jerk values of the improved controller, shows much more stable behavior. Across all cars, the maximum and minimum jerk values are considerably lower, with none exceeding ± 2.5 m/s^3. This reduction in amplitude highlights the controller's improved ability to regulate acceleration transitions smoothly. Furthermore, the standard deviations for all cars (as seen in Table 1) are now significantly smaller, particularly for Cars 5 to 8, which exhibit highly consistent and minimal jerk values.

The difference in performance is especially evident when examining Cars 2 and 4. In the previous controller's plot (Figure 2), Car 2 displayed frequent, large spikes indicative of abrupt, poorly managed transitions. However, in Figure 1, the improved controller mitigates such behavior, with Cars 2 and 4 still showing some variability but at a much more controlled and manageable level.

This stark improvement can be attributed to changes in the controller's design, including better calibration of its response to input signals, improved filtering of sensor noise, and more refined handling of dynamic conditions. These adjustments have allowed the system to maintain smoother transitions, avoiding the dangerous and uncomfortable spikes observed in the previous controller's performance.

5.5 Deployment at Scale

If our improved controller were deployed at scale, the overall impact on traffic dynamics would likely be positive, with a noticeable improvement in driving comfort, vehicle stability, and overall traffic flow. The analysis of our controller, as demonstrated in Figures 1 and Table 1, provides several pieces of evidence supporting these outcomes. However, there are still areas of concern and challenges that need to be addressed for large-scale implementation.

Positive Impacts:

1. Smoother Driving Experience:
   The controller effectively reduced jerk values to acceptable levels across most vehicles. For Cars 5 through 8, in particular, the standard deviation of jerk was very low (e.g., 0.0832 m/s^3 for Car 6, as shown in Table 1), and the jerk values were well-contained within a small range (Figure 3). This means the controller minimized

abrupt changes in acceleration, providing a smoother driving experience for passengers. At scale, this would lead to better comfort and fewer instances of motion sickness or driver fatigue during prolonged commutes.

2. Reduced Vehicle Wear and Tear:
   By limiting extreme spikes in jerk values, as shown in Figure 1, the improved controller reduces stress on vehicle components such as brakes, tires, and suspension systems. Over time, this would lower maintenance costs for vehicle owners and enhance the reliability of vehicles in traffic.

3. Improved Traffic Flow:
   With the controller ensuring smoother acceleration and deceleration transitions, vehicles are less likely to abruptly brake or accelerate, reducing the occurrence of shockwave traffic jams. This can lead to more consistent traffic flow, especially in urban areas with stop-and-go conditions. The reduced jerk values for Cars 5 to 8 suggest that the system performs particularly well at lower speeds and with vehicles that are less influenced by external disturbances.

4. Safety Enhancements:
   The controller mitigates sudden acceleration or deceleration, reducing the likelihood of rear-end collisions in dense traffic. By maintaining consistent behavior, the system creates predictable driving patterns that can improve the safety of automated vehicles and their interactions with human-driven cars.

Challenges and Concerns:

1. Inconsistent Performance Across Vehicles:
   While Cars 5 through 8 showed significant improvements, the performance for Cars 1 through 4 was less consistent, as highlighted by their higher standard deviation and maximum jerk values (Table 1). Table 1 also clearly demonstrates that Cars 1 through 4 experienced greater variability and higher peaks in jerk, suggesting that the controller’s effectiveness depends on factors such as the vehicle’s hardware or environmental conditions. At scale, this inconsistency could lead to uneven experiences for drivers and passengers, with some vehicles performing far better than others.

2. Environmental Sensitivity:
   The analysis showed that external factors like traffic density or road conditions could still impact controller performance. Cars 2 and 4, in particular, showed erratic behavior, likely due to poor initial parameters or weak noise filtering in the feedback loop. On a larger scale, this sensitivity might cause performance to vary depending on the driving environment.

3. Scalability and Infrastructure:
   Deploying this controller system at scale would need wide adoption of the required technology, including high-quality sensors and reliable vehicle communication. Differences in hardware between manufacturers or in older vehicles could reduce the system's effectiveness and cause inconsistencies in traffic behavior.

4. Edge Cases and Failures:
   While the system works well under typical conditions, edge cases involving sudden environmental changes (e.g., icy roads, sudden braking by a lead car) may still challenge the controller. Extreme jerk spikes need to be entirely eliminated for large-scale deployment to ensure safety in all scenarios.

5.4 Impacts on Traffic

If deployed at scale, the improved controller would likely lead to smoother, more efficient traffic flow, particularly in urban environments. However, to achieve full-scale deployment, further development is required to address the inconsistencies observed in specific vehicles. By refining the system to handle variability in hardware, environmental conditions, and edge cases, the controller has the potential to significantly enhance traffic dynamics, reduce accidents, and improve the overall driving experience.

6 Limitation / Implication

Our controller faced several limitations and implications that may impact its performance and applicability in various driving conditions. First, the system assumes that the driver will stay alert and step in when needed, especially in situations like merging lanes or unexpected obstacles. This dependence on the driver reduces the system's independence and may lower the driver's trust in its ability to handle every situation on its own, which could increase safety risks in emergencies. The system's performance also assumes normal road and weather conditions, which may not hold in extreme weather, such as snow, ice, or heavy rain, where road traction and visibility are reduced. In such conditions, the system may not function optimally, requiring either manual deactivation or more frequent driver intervention. Another limitation is in the system's goal to keep jerk (rate of acceleration change) below 2 m/s^3 to ensure comfort, which can slow the system's responses in critical situations. While this reduces discomfort for passengers, it may compromise the system's ability to react quickly enough during high-stress driving conditions. Overall, while the system offers a promising solution for driving, these limitations highlight the challenges in balancing comfort, safety, and performance across different driving environments. To address these issues, further refinement and extensive real-world testing would be necessary to improve the system's capabilities and reliability.