Project Proposal
Environmentally Adaptive Cruise Control in Autonomous Vehicles that Imitate Human Behavior
Anthony Peters
December 3, 2023

1. Introduction

Autonomous Vehicles (AV) are at the forefront of the future and will soon play a significant role in the daily lives of Motorists. AV must be developed alongside strict requirements that encourage safety and improve efficiency for drivers on the roadway. Driving a car can be extremely dangerous and challenging as many variables are present on the road as a driver goes to their destination. For most, driving a car can be the most dangerous activity they ever do as they are constantly avoiding other drivers' mistakes, roadway hazards, mechanical failure of their car, or even fighting the urge to become distracted from irrelevant events of the roadway. Road conditions can also drastically change throughout the day depending on the season, time of day, or day of the week. With deductive reasoning, a human can assume that driving on a curvy road in the middle of the night might be more dangerous than in the middle of the day for many reasons, and they will adjust their driving behavior accordingly. All these factors play into the lethal possibility of driving a car on the road. With the popularity of autonomous vehicles rising in the past decade, they have the potential to effectively mitigate the dangers that present themselves from driving a car on the road. Using Sensor Fusion and Deep learning algorithms, appropriate control strategies will allow an AV to alter its driving characteristics based on the surrounding environment and road characteristics to adhere to passenger comfort and trust in the AV.

2. Literature Review

AVs do not alter their driving behavior to adhere to passenger comfort. Today, AVs are jerky and robotic and require implementing an elegant solution where they drive for passenger safety and convenience. Sensor Fusion is a necessary topic as AVs use multiple onboard sensors to sense their environment and dynamics while driving on the road. Because of the nature of Sensor Fusion, deep learning has an opportunity to be implemented alongside the data generated to refine the large amount of data coming in from the multiple sensors, in turn helping the efficiency of the Sensor Fusion Framework. Deep learning also has its place in Dynamically tracking the AV's surrounding environment as other hazards present themselves on the roadways, such as road conditions, pedestrians, motorcyclists/cyclists, and even wildlife. Dynamically modeling the environment is crucial for a decision-making process in the AV. For this project, the vehicle must know the status of its ever-changing environment relative to its position to deploy meaningful dynamic control for the passenger's standards. A Passengers Standard for a comfortable threshold of an AV is another realm of study to be pursued as this topic can be widely opinionated and quantifiably difficult. Once the correct data from the car and the environment is obtained, these standards must be reelected through the vehicle's control strategies to drive itself.

2.1 Sensor Fusion for Localization of an AV

A wide variety of Sensors equipped inside an AV set up the vehicle for success to achieve a wide variety of tasks, given defined performance criteria. It's necessary to deploy Sensor Fusion methodology inside an AV to achieve a desirable and elegant solution. When choosing a Sensor Fusion Framework, essential trade-offs to consider are between data resolution and cost of sensors as well as fusion model and real-world real-time applicability [1]. One way to lower the cost and processing power from onboard sensors is to take advantage of the previously set up infrastructure the road already has, such as the traffic cameras to locally map the car and assist with its localization algorithms [2]. Another approach to extracting large amounts of data is to use Fast Fourier Transform algorithms that obtain critical features from the frequency domain instead of the time domain and then place the filtered data inside a deep learning model to leverage the model's inherent robustness [3]. Aside from elegant approaches to locally characterize an AV, it is popular to approach a Sensor Fusion framework of an AV with a combination of Visual and inertial sensors (Camera, IMU, and Accelerometer) and then to give a specific weight of the sensors within control principles based on the limitations of the sensors *[4].

2.2 Dynamic Environment Tracking in Autonomous Vehicles

Along the lines of Sensor Fusion comes the ability to dynamically track the activities happening in the environment around the car. Dynamically tracking the environment will influence the Sensor Fusion framework that maps the localization of the vehicle; however, it will run in parallel to those control principles, generating its own. OpenCV is an open-source library that leverages pre-existing object detection models to help track various objects in an environment. The use of OpenCV and previously trained models bypasses the meticulous data collection and allows the control engineer to focus on control laws to better process and control the desired performance of the AV. Object detection and deep learning model YOLOv3, alongside OpenCV, is used to obtain the lateral and longitudinal distances relative to the camera frame on the car [5]. A classical approach is used to get these parameters, which uses the camera on board coupled with an observer controller to generate estimations for the lateral and longitudinal distances of objects in the AV's Environment [5]. Because learning models and vision detection have inaccuracies and demanding computational loads, alternative approaches, such as an adaptive approximation-based tracking control made via fuzzy systems, have been implemented to ensure robust tracking in the face of functional uncertainties from outside disturbances [6]. Because of the uncertainties in the vehicle lateral dynamics due to the road surface or the pitch and roll movements of the car, robust fuzzy and sliding mode control schemes are implemented to track objects in the environment [7]. The environmental uncertainties have also been characterized using an Adaptive Robust Controller alongside the Lyapunov minimax analysis, validating the approach through Car-Sim in Math Works Simulink [8]. Far more complex and elegant methods have taken place to better account for the uncertainties in the environment using homogeneity-based finite-time control nested with a finite-time disturbance observer aiming to deal with complicated unmodeled unknowns [9].

2.3 Autonomous Vehicle and Human Behavior

AVs need to imbed a driving algorithm that emulates safe driving qualities. It is essential to quantify the skills of a human driver. Driving skills can be heavily opinionated and vary from person to person, and quantifying this within a computer presents a problem as it operates on a binary basis. Through a simulation, a corridor-based Nonlinear Model Predictive Control (NMPC) controls the vehicle's state to achieve a smooth and comfortable trajectory while applying trajectory constraints using the safety corridor [10]. This model became highly influential in adapting the cruise control speed as it traveled through a turn whose radius was continuously changing [10]. A comfort function was also developed and defined as jerk divided by negative 60 all to the power of 2 [11]. This defined comfort function was an action boundary for the AV to operate under [11]. Even though comfort is a significant value to consider in driving skills, the comfort boundary is a trade-off to the stability of the tracking problem, as the low amount of acceleration and jerk will lead a car with a slow tracking algorithm [11]. Defining the correct amount of trade-off is challenging, and some work has been done to explain a comfortable driving setting. Relevant factors that affect passenger comfort were determined via binary passenger comfort and the differences between passengers' acceptance of the same vehicle operating parameters [12]. This real-time study presents valuable, quantifiable data. It suggests establishing a multi-objective optimization function in local trajectory planning of intelligent vehicles, including vehicle acceleration, passenger gender, and carickness susceptibility. This data can be used with algorithms such as fuzzy logic controllers to allocate the weight function in the multi-objective function to improve the comfort of passengers on the premise of their own personal comfort and safety standards [12].

2.4 Status and Outlook

A large amount of work has been done toward optimizing the driving behavior of an AV. Many control strategy approaches use fuzzy control logic to quantify the environmental disturbances, feeding the scaled data into a fuzzy controller and using it within a chosen control scheme. Various Sensor Fusion Frameworks have been experimented with in the research to find an optimal configuration. However, to balance the project's cost, an IMU, accelerator, radar, and a Camera are popular to test control laws within a controlled environment [4]. Most of the presented work has been made through computer simulation, which offers an exciting challenge when implementing the principles in the literature within a real-world environment and test scheme.

3. Objective

The objective of this proposed project is to implement cruise control within a robotic autonomous vehicle kit that adapts driving behavior to road conditions and dynamics. The autonomous vehicle will be able to recognize obscurities on the road and the road's turn radius to adapt its driving behavior safely.

3.1 Significance

Current autonomous vehicles drive with a robotic behavior, and driving quality and passenger comfort are not considered. For example, level 2 and level 3 autonomous cars can drive themselves for the most part, and they do a great job driving when a vehicle is in front of them. However, when the cars reach a sharp turn, a pothole, gravel on the road, etc, the car does not adjust its driving behavior. As a result, this leaves the driving trajectory of the vehicle unoptimized and not trusted by the passenger.

3.*2 Deliverables*

In addition to obligatory reports and presentations, the most critical deliverables are as follows:

1. A small-scale (RC) autonomous vehicle and an indoor test track environment
2. Written protocol for the autonomous car to complete a turn aligned with a defined safe human driving behavior.
3. Computer simulations and control algorithms for the safe handling and driving of the vehicle.

3.3 Evaluation Metrics

The following evaluation metrics will measure success in this endeavor:

1. Achieve 98% speed tracking accuracy on an obstacle-free straight road.
2. Maintain at most 15% departure from the road center during turns.
3. Achieve above 90% success rate of obscurities within the road with sensor detection.

4. Methodology

This project will utilize the knowledge and mathematical principles from Control Theory, Mechatronics, and Mechanical design. First, an RC-Car kit and a test track will be acquired and built to operate indoors. Second, the vehicle's kinematic bicycle model will be derived, and the electrical system parameters will be identified. Third, model derivation and the acquired parameters will be used for MATLAB simulations. Fourth, real-time sensor data will be collected and mapped. Lastly, a control strategy will be developed and executed for the vehicle in the test environment. Thonny will be the IDE, and Python will be the logic used for the hardware. High-quality data will be obtained from the system and cross-validated at each stage to reduce model validation errors.

4.1 Work Breakdown Structure

	Jan 2024	Feb 2024	Mar 2024	Apr 2024	May 2024	Jun 2024	Jul 2024	Aug 2024	Sep 2024	Oct 2024	Nov 2024	Dec 2024
Deliverable 1: Small-scale (RC) autonomous vehicle and an indoor test track environment												
Task 1.1 Select and procure robotic car kit components												
Task 1.2 Assemble Kit and develop simple drive algorithms through Python.												
Task 1.3 Design and manufacture track components.												
Deliverable 2: Computer simulations and control algorithms for the safe handling and driving of the vehicle												
Task 2.1 Characterize vehicle and motor dynamics												
Task 2.2 Develop safe and comfortable control laws												
Task 2.3 Generate desirable simulation results												
Deliverable 3: Written protocol for autonomous vehicles to complete a turn aligned with a defined safe human driving behavior.												
Task 3.1 Acquire vehicle velocity in terms of voltage input												
Task 3.2 Acquire appropriate data from Sensor Fusion												
Task 3.3 Tune driving algorithms												
Reporting-related deliverables			1	2	3		4		5		6	7

Table 1: Expected Project Gantt Chart

4.1.1 Small-scale (RC) autonomous vehicle and indoor test track environment

The first deliverable will be the complete test environment for the autonomous vehicle. This includes a completely assembled autonomous vehicle and the test track it will operate within. The autonomous vehicle will be bought as a kit online with the appropriate hardware, and the test track will be constructed with cardboard with 3D-printed stands to prop the cardboard upright. After the vehicle's assembly, simple drive algorithms will be used to test functionality. The cardboard will be used as the boundary for the autonomous vehicle to operate within, and the 3D printed stand will allow the border to be changed throughout the project for different levels of difficulty for the car to work within.

4.1.2 Computer simulations and control algorithms for the safe handling and driving of the vehicle.

The second deliverable is the simulation results of the vehicle's kinematic model's safe driving and handling control algorithms. The simulation will provide plots that depict a 15% departure from the road's center line and speed tracking plots that maintain 98% accuracy from input and the vehicle. The simulation will require the motor and kinematic vehicle models to select and tune a control strategy accurately. The DC motor model can be modeled with an electrical system coupled to a mechanical system where the relationship is seen between the applied voltage and the spinning shaft of the motor. DC motor parameters can be determined from steady-state voltage and current measurements. The kinematic vehicle model will be modeled as a ridged link connected by two wheels, i.e., using the notorious bicycle model. These two models can be merged to formulate a complete model of the actual car and used for simulation to test various adaptive control techniques.

4.1.3 Written protocol for the autonomous vehicle to complete a turn aligned with a defined safe human driving behavior.

The third deliverable is to present a written protocol for the autonomous vehicle that resembles safe driving behavior from a human. The data from the vehicle's IMU will be acquired and calibrated from the ADAFruits CircuitPython MPU6050 library. Image recognition will be processed through OpenCV, using a variety of open-source object detection and classification libraries. OpenCV will be used for image processing for lane detection and position. The motor encoder values are used alongside the derived motor model for vehicle speed and positioning. The protocol will use Sensor Fusion from the vehicle's motor encoders, IMU, and camera to feed back into the developed control laws derived during the simulation phase and present smooth and cautious driving within the test track. Separate tests can then be performed to validate a 90% detection rate of road obscurities, 98% speed tracking on an obstacle-free road, and a 15% deviation of the car position from the road center line.

4.2 Resources Needed

No	Resource	Time Frame	Source
1	MATLAB License	January 2024 – December 2024	SJSU CoE Licenses
2	SolidWorks License	January 2024 – March 2024	SJSU CoE Licenses
3	3D-Printer	January 2024 – March 2024	Acquired
4	SunFounder Smart Video Robot Car Kit	January 2024 – December 2024	Amazon
5	Raspberry Pi 4 B	January 2024 – December 2024	Amazon
6	Monitor, Keyboard, Mouse, MicroSD Card	January 2024 – December 2024	Acquired
7	Voltmeter, Bread Board Kit, Sensors, Solder, Magnets	January 2024 – July 2024	Amazon
8	Fasteners	January 2024 – December 2024	Acquired

Table 2: Resources Needed for the Success of the Project

This project is designed to be completed using MATLAB as the simulation software and the Python platform for hardware implementation. SolidWorks will be used to ensure proper fixture design, and MATLAB will be used for system modeling and generating 2D plots. San Jose State University College of Engineering will provide MATLAB and SolidWorks licenses. Alongside the Robot Car Kit, the Raspberry Pi 4 B will be purchased from Amazon to run the

Python on the hardware. To operate the Raspberry Pi 4 B, a monitor, keyboard, mouse, and a MicroSD card will be used. To support software and fixture implementation, essential prototyping tools such as a 3D printer, fasteners, voltmeter, soldering iron, breadboard kit, and sensors will be needed to assemble and troubleshoot the SunFounder car kit.

4.3 Risks and Contingency Plans

No	Risk	Contingency Plan
1	Camera resolution limitation for image processing	Altering the environment light with an LED can help filter out camera noise.
2	Hardware Failure (Motors, Sensors, Mechanical Parts)	Buy new with saved funds or use a 3D printer to create custom replacement parts.
3	Image processing algorithm to perceive characteristics and objects in the car's path.	Using Opensource algorithms along with consistent peer reviews will assist in the desired data acquisition of the camera.

Table 3: Risks and Contingency Plans

5. Preliminary Work

5.1 Relevance to the Objectives

The preliminary work established a foundation for implementing and observing control principles within the autonomous vehicle and initializing the simulation model for the RC Car in MATLAB. Initially, a car kit containing the required electrical components and mechanical mechanisms was identified and bought. Subsequently, the vehicle was assembled, and the car's characteristics were acquired. Lastly, the derivation of the bicycle model was conducted as the initial step towards simulating the car in MATLAB. The primary output of this preliminary work was a simulation report of the RC-Car, including a position tracking plot. Minor accomplishments included the procurement of necessary project resources. For future endeavors, various control strategies will be incorporated and expanded upon in the simulation utilizing the available sensor data from the car.

5.2 Component Selection and Assembly

"SunFounders Raspberry Pi Smart Video Robot Car Kit" was bought on Amazon and shipped to be assembled at home. The vehicle, in essence, is a scaled-down version of a real car on public roads and can be modeled and simulated as such. The kit comes with the necessary sensors to use for correct control strategies to take place.

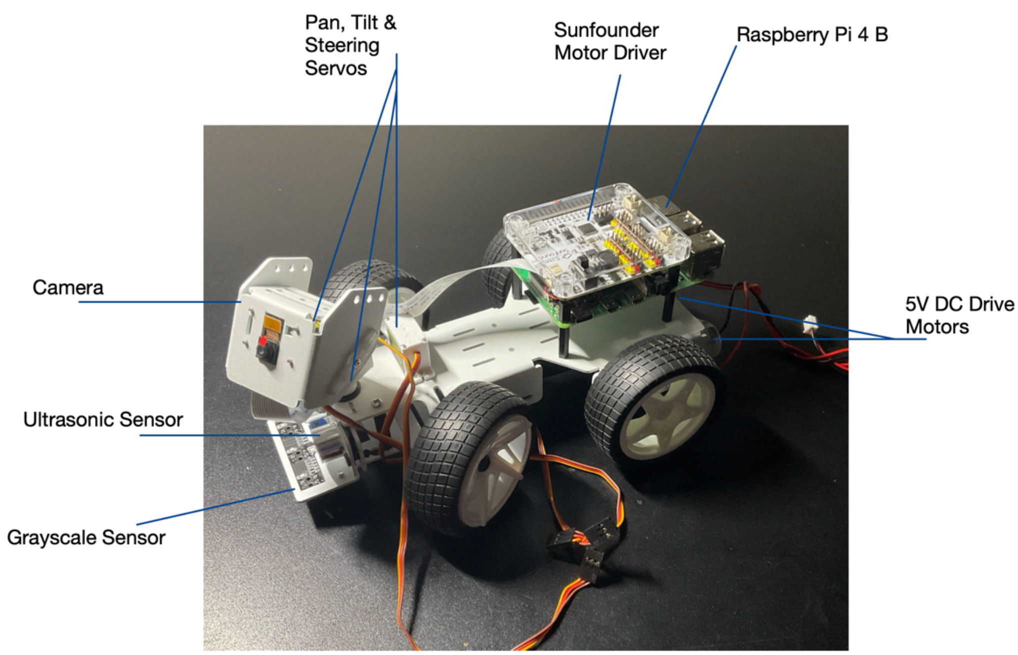

Figure 1: Completed Assembly

5.3 Kinematic Bicycle Model Simulation

5.3.1 Model Derivation and Car Data Acquisition

Previous research conducted by Kumbhani [13] formulates foundational work for the simulation and its governing equations to be expanded upon. The nonlinear kinematic derivation for the bicycle model in an inertial frame is as follows:

Lateral Vehicle Control

$$\dot{X} = v\cos(\varphi + \beta) \qquad (1)$$

$$\dot{Y} = v\sin(\varphi + \beta) \qquad (2)$$

$$\dot{\varphi} = \frac{v}{Lf}\sin(\beta) \qquad (3)$$

$$\beta = tan^{-1}\left(\frac{Lf}{L}\tan(\delta)\right) \qquad (4)$$

$$\dot{y} = v\sin(\beta) \qquad (5)$$

The lateral tracking error is $e_y = y_d - y$ where y_d is the desired lateral trajectory.

$$\delta = tan^{-1}\left(\frac{L}{Lf}\tan\left(sin^{-1}\left(\frac{1}{v}\left(\dot{y_d} + K_d\dot{e_y} + K_y e_y + \int K_i e_y dt\right)\right)\right)\right) \qquad (6)$$

Sigma is the control variable updated with a PID Control law.

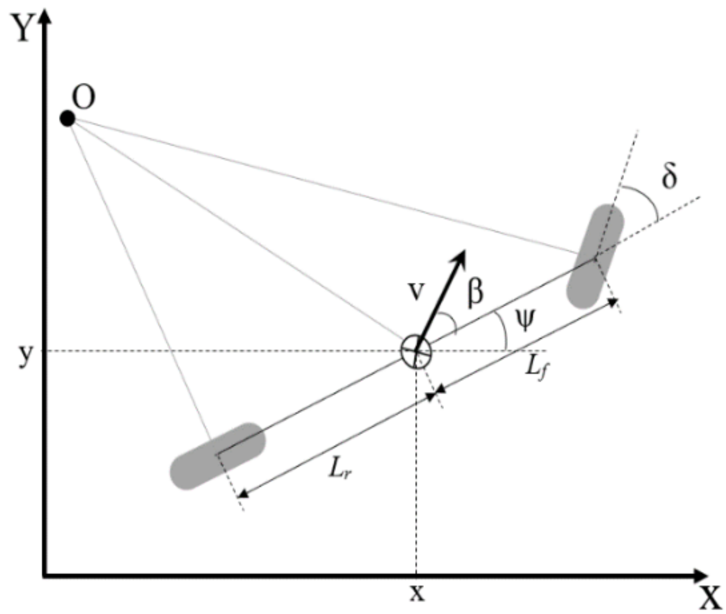

Figure 2: Kinematic bicycle vehicle model

$\dot{X}$ and $\dot{Y}$ is the vehicle velocity in the coordinate system, φ is the vehicle heading, β represents the vehicle velocity angle, and δ is the steering angle. For lateral control simulation of the vehicle, Lr, Lf, and L were data gathered from the car.

L_f	0.05 [m]
L_r	0.05 [m]
L	0.1 [m]

Table 4: Vehicle Parameters

5.3.3 Simulation Results

To simulate lateral control dynamics, Matalab applied a sinusoidal road to the governing equations of the vehicles (1)-(4) and (6), with a constant velocity used, as shown in Figure 3. The gains of the PID controller were tuned for a desired path trajectory for the vehicle.

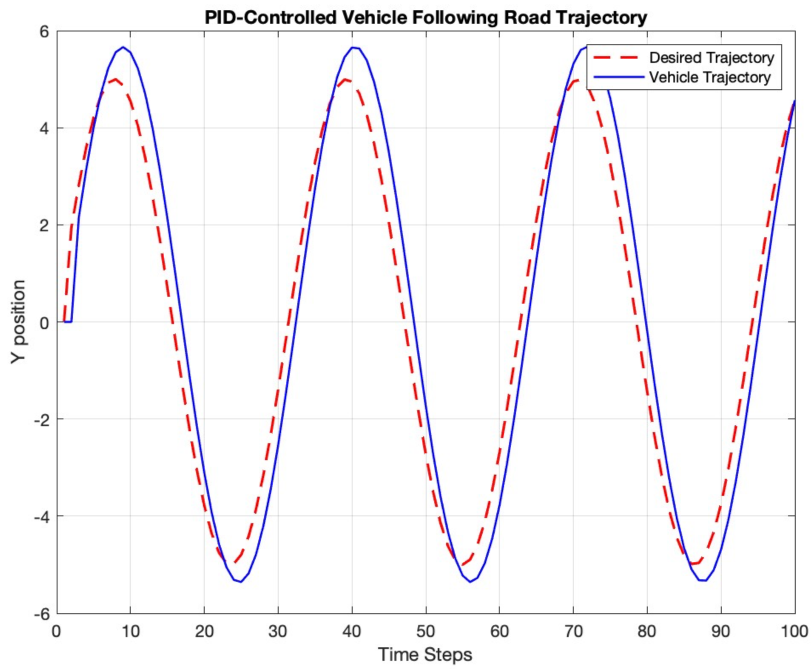

Figure 3: Drive Cycle Tracking

5.4 Preliminary Work Conclusion

In this preliminary work report, an autonomous vehicle is selected and assembled. The kinematic bicycle model is derived alongside a PID controller to update the steering for the vehicle's lateral control. Length measurements were taken from the car to implement into the simulation. The governing equations were then programmed into Matlab and simulated with a sinusoidal road and constant input velocity. In future work, this simulation will be used as a framework to implement a longitudinal control for the vehicle and various types of control laws.

References Cited

*[1] B. Shahian Jahromi, T. Tulabandhula, and S. Cetin, "Real-Time Hybrid Multi-Sensor Fusion Framework for Perception in Autonomous Vehicles," Sensors, vol. 19, no. 20, Art. no. 20, Jan. 2019, doi: 10.3390/s19204357.

*[2] M. Hasanujjaman, M. Z. Chowdhury, and Y. M. Jang, "Sensor Fusion in Autonomous Vehicle with Traffic Surveillance Camera System: Detection, Localization, and AI Networking," Sensors, vol. 23, no. 6, Art. no. 6, Jan. 2023, doi: 10.3390/s23063335.

*[3] P. de Souza et al., "A Study on the Influence of Sensors in Frequency and Time Domains on Context Recognition," Sensors, vol. 23, no. 12, Art. no. 12, Jan. 2023, doi: 10.3390/s23125756.

*[4] J. Kelly and G. S. Sukhatme, "Visual-Inertial Sensor Fusion: Localization, Mapping and Sensor-to-Sensor Self-calibration," The International Journal of Robotics Research, vol. 30, no. 1, pp. 56–79, Jan. 2011, doi: 10.1177/0278364910382802.

*[5] F. Lotfi and H. D. Taghirad, "A framework for 3D tracking of frontal dynamic objects in autonomous cars," Expert Systems with Applications, vol. 183, p. 115343, Nov. 2021, doi: 10.1016/j.eswa.2021.115343.

*[6] N. Wang, M. J. Er, J.-C. Sun, and Y.-C. Liu, "Adaptive Robust Online Constructive Fuzzy Control of a Complex Surface Vehicle System," IEEE Transactions on Cybernetics, vol. 46, no. 7, pp. 1511–1523, Jul. 2016, doi: 10.1109/TCYB.2015.2451116.

*[7] C.-L. Hwang and W.-L. Fang, "Global Fuzzy Adaptive Hierarchical Path Tracking Control of a Mobile Robot With Experimental Validation," IEEE Transactions on Fuzzy Systems, vol. 24, no. 3, pp. 724–740, Jun. 2016, doi: 10.1109/TFUZZ.2015.2476519.

*[8] Z. Hu, Z. Yu, Z. Yang, Z. Hu, and Y. Bian, "Rendering bounded error in adaptive robust path tracking control for autonomous vehicles," IET Control Theory & Applications, vol. 16, no. 12, pp. 1259–1270, 2022, doi: 10.1049/cth2.12303.

*[9] N. Wang, S. Lv, M. J. Er, and W.-H. Chen, "Fast and Accurate Trajectory Tracking Control of an Autonomous Surface Vehicle With Unmodeled Dynamics and Disturbances," IEEE Transactions on Intelligent Vehicles, vol. 1, no. 3, pp. 230–243, Sep. 2016, doi:10.1109/TIV.2017.2657379

*[10] C. Wei et al., "Risk-based autonomous vehicle motion control with considering human driver's behavior," Transportation Research Part C: Emerging Technologies, vol. 107, pp. 1–14, Oct. 2019, doi: 10.1016/j.trc.2019.08.003.

*[11] W. Tang, L. Xu, and J. Ma, "Modeling Autonomous Vehicles' Altruistic Behavior to Human-Driven Vehicles in the Car following Events and Impact Analysis," Journal of Advanced Transportation, vol. 2023, p. e4060451, Apr. 2023, doi: 10.1155/2023/4060451.

*[12] P. A. M. Ruijten, J. M. B. Terken, and S. N. Chandramouli, "Enhancing Trust in Autonomous Vehicles through Intelligent User Interfaces That Mimic Human Behavior," Multimodal Technologies and Interaction, vol. 2, no. 4, Art. no. 4, Dec. 2018, doi: 10.3390/mti2040062.

*[13] N. Kumbhani and S. Bashash, "A Supervisory Lateral Slip Prevention Controller for Autonomous Vehicles," presented at the ASME 2019 Dynamic Systems and Control Conference, American Society of Mechanical Engineers Digital Collection, Nov. 2019. doi: 10.1115/DSCC2019-9166.