Adaptive Lane Keeping Assist for an Autonomous Vehicle based on Steering Fuzzy-PID Control in ROS

Hoang Tran Ngoc, Luyl-Da Quach
Software Engineering Department
FPT University Can Tho city, Vietnam

Abstract—**An autonomous vehicle is a vehicle that can run autonomously using a control. There are two modern autonomous assistant systems that are proposed in this research. First, we introduce a real-time approach to detect lanes of the streets. Based on a series of multi-step image processing through input data from the camera, the vehicle's steering angle is estimated for lane keeping. Second, the steering control system ensures that autonomous vehicles can operate stably, and smoothly, and adapt to various road conditions. The steering controller consists of a PID controller and fuzzy logic control strategy to adjust the controller parameters. The simulation experiments by Gazebo simulator of the Robot Operating System (ROS) not only indicate that the vehicle can keep the lane safely, but also demonstrate that the proposed steering angle controller is more stable and adaptive than the conventional PID controller.**

***Keywords*—Autonomous vehicles; automated steering; lane detection; fuzzy PID control; ROS, Gazebo**

I. Introduction

Every year nearly 1.25 million people die in traffic accidents. Therefore, safety is the most important criterion when driving on the road. Traffic accidents are caused by human errors up to 90% according to National Highway Traffic Safety Administration (NHTSA) statistics [1]. In recent five years, autonomous vehicles, and many advanced driver assistance systems (ADAS) have been developed to help drivers to drive safely. It attracts great attention not only in academia but also in automotive industries such as Google, Tesla, BMW, and Hyundai. In there, lane boundary detection and steering control systems play a key role in autonomous vehicles (AVs).

To detect the lanes as well as the steering angle, common sensors are used such as radar, light imaging detection and ranging (LIDAR), laser sensors, and even global positioning systems (GPS) [2-5]. However, sensors with high accuracy of distance measurement are very expensive. Therefore, numerous studies have been developing vision-based systems using camera sensors for lane detection [6-7] in recent years. These studies indicate that the lane detection process has major challenges such as lighting and background clutter.

Along with the studies of determining steering angle based on lane detection via vision, approaches to improve the stability and accuracy of the steering angle controller are also interesting in several studies around the world. For example, a basic steering control algorithm based on a PID controller was proposed in [8]. Moreover, to address a repeated problem of steering control, a robust PID controller is designed in [9]. Another approach was nested conventional PI and PID controllers to improve the accuracy of the steering angle controllers [10-15]. However, the damping effect of the conventional PID controller is not good enough to keep the vehicle running smoothly for lane tracking. Therefore, to develop the autonomous system without mentioned problems, we have two main goals: estimating the steering angle and designing the steering controller with high stability.

In this research, the steering angle is estimated through a sequence of image processing steps and computer vision approaches. The image will be filtered, determined the region of interest (ROI), and finally extracted the line segments using the simplified Hough Transform technique. After obtaining the estimated steering angle, this paper proposed a method to optimize the performance of the steering control system. The idea is to use the fuzzy logic control strategy to tune the parameters of the PID controller. The fuzzy controller adjusts the parameters of the PID based on the steering angle error and previous information. The effectiveness of the two proposed systems is verified by the robot simulation results in the Gazebo environment.

The structure of this research paper is organized as follows. First, in Section 2, the vehicle steering model is introduced. In Section 3, lane detection and steering angle estimation are presented. The proposed steering control system using a Fuzzy-PID controller is analyzed and designed in Section 4. Then, the simulation results and analysis of the corresponding system are compared in Section 5. Finally, the conclusions in Section 6 will summarize the content of this research.

II. Vehicle Dynamic Model

The vehicle dynamics and vehicle steering behavior are considered in the 2D bicycle model [16]. The linear vehicle steering model is described as follows:

$$\begin{bmatrix} \dot{y} \\ \dot{\beta} \\ \dot{\psi} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} 0 & V & V & 0 \\ 0 & -\frac{D_r+D_f}{MV} & 0 & \frac{D_rL_r-D_fL_f}{MV^2} \\ 0 & 0 & 0 & 1 \\ 0 & \frac{D_rL_r-D_fL_f}{L_z} & 0 & \frac{D_rL_r^2-D_fL_f^2}{L_zV} \end{bmatrix} \begin{bmatrix} y \\ \beta \\ \psi \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{D_f}{MV} \\ 0 \\ \frac{D_fL_f}{L_z} \end{bmatrix} \delta \quad (1)$$

where y, β, ψ, $\dot{\psi}$, V, L_z, M, L_f and L_r are vehicle position, side slip angle, yaw angle, yaw rate, vehicle speed, vehicle inertia, vehicle mass, and the center of gravity distance from front tires, and rear tires. The lateral forces with

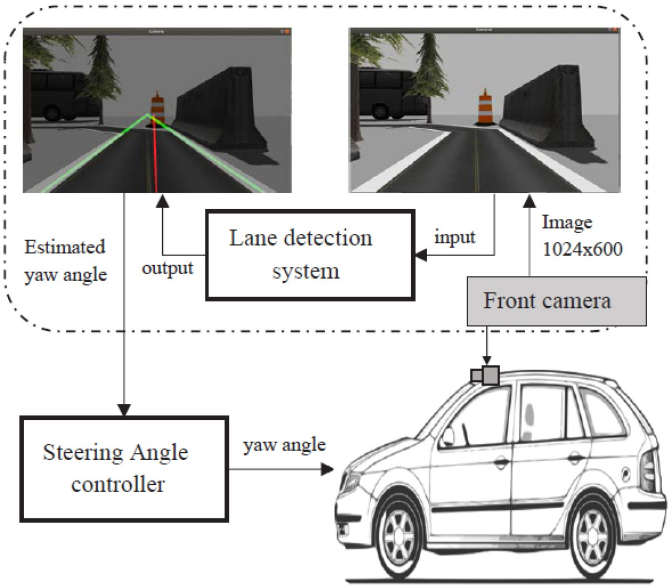


Fig. 1. Block Diagram of the Proposed System.

linearized cornering stiffness of the front, and rear wheel (D_f and D_r) are shown in (2) below:

$$\begin{cases} F_{yf} = D_f \alpha_f = D_f \left(\delta - \beta - \frac{L_f \dot{\psi}}{V} \right) \\ F_{yr} = D_r \alpha_r = D_r \left(-\beta + \frac{L_r \dot{\psi}}{V} \right) \end{cases} \quad (2)$$

where α_f, α_r, and δ are front, rear tire slip angle, and steering angle from the forward direction of the vehicle.

The proposed system and variables of the vehicle model are presented in Fig. 1 and Fig. 2. In this research, the yaw angle is estimated from the lane detection system and used as the steering angle to control the robot model in the Gazebo simulator environment. In the real system, the front wheel steering angle (δ) will be converted from a yaw angle (ψ).

A. Lane Detection Algorithm

To control the proposed autonomous vehicle to keep the lane accurately, we used algorithms [17] and adapted them to match our system by using the following steps: First, to detect the white lanes, we isolate all the white areas on the image. To do this, we converted the color space of the image from RGB (Red/Green/Blue) to the HSV (Hue/Saturation/Value) and created the mask image for a range of white colors. Second step, we extracted edges in the white mask by using the Canny edge detector [18-19]. This algorithm is developed by John F. Canny that can detect edges and reduce the number of erroneous edges detected in mask images. In the third step, we need to detect a few white edge areas that are no lane lines. The extraction of an Isolate Region of Interest (IROI) is performed in this step. We cropped the detected edges in the top half of the image. Fourth step, line segment detection is applied. We extracted the coordinates of the lane lines from white pixels by using the Hough Transform technique [20]. The Hough Transform is a common algorithm, used in image processing to find features such as lines, circles, and ellipses.

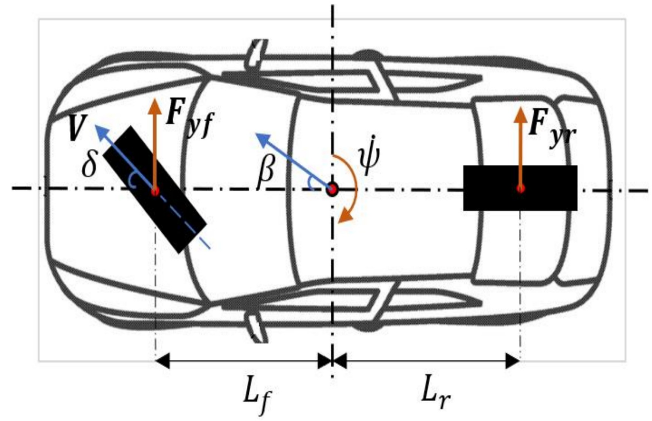


Fig. 2. Block Diagram of the Proposed System.

We applied it to detect straight lines from pixels that seem to line up.

The Hough Transform is the transformation from points to curves, which converts the Cartesian coordinate system of the image to the polar coordinate Hough space as (3):

$$\rho = x cos\left(\theta\right) + y sin\left(\theta\right), \quad \theta \in \left[-\pi \quad \pi\right] \quad (3)$$

where (x, y) is the pixel coordinates; (θ, ρ) is the polar coordinates; ρ the distance of the straight line from the coordinate origin; θ is the minimum angle of the straight line in the normal direction with the positive direction of the x-axis. Points on the same line satisfy the (3) with a set of (θ, ρ) constants. Fig. 3 shows the basic principle of the Hough Transform. From this basic principle, line segments are determined.

III. Lane Detection and Steering Angle Estimation

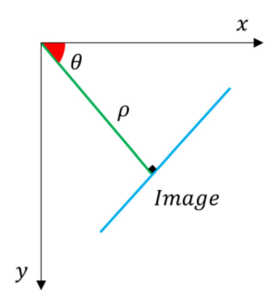


Fig. 3. Basic Principle of Hough Transform.

The final step is to combine line segments into two-lane lines. We classified small line segments into two groups by their slope. All the line segments of the left lane line have an upward slope (θ range from 15^o to 85^o), whereas all line segments of the right lane line have a downward slope (θ range from -15^o to -85^o). We then averaged the slope and the intersection point to detect the left and right lanes. Fig. 4 shows the lane detection stages and results of our system in the Gazebo simulation environment.

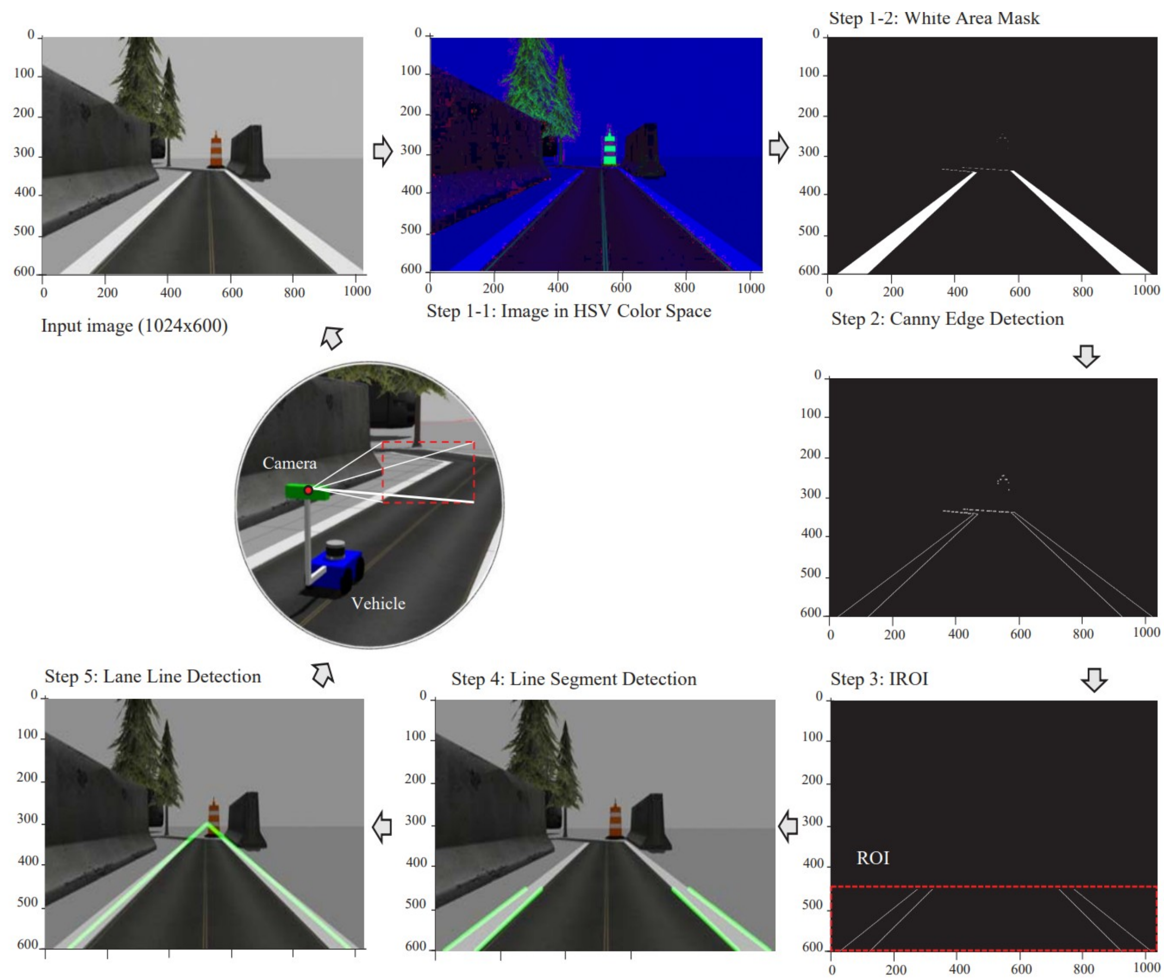


Fig. 4. Lane Detection Steps and Results of our System.

A. Steering Angle Estimation

After using the lane detection algorithm, we have the coordinates of two lanes. We need to estimate the steering angle to steer the car so that it keeps in the middle of the two lanes. We have two cases where the vehicle drifts to the left side or the right side of the road, and the vehicle is already in the middle of the lane. From the coordinates of the two lanes, the steering angle will be determined according to the following (4):

$$\widehat{\psi} = tan\left(\frac{x_{offset}}{\frac{H_i}{2}}\right) \tag{4}$$

$$\widehat{\psi} = \begin{cases} tan\left(\frac{\left(\frac{x_{1(R)}-x_{1(L)}}{2}+x_{1(L)}\right)-\frac{W_i}{2}}{H_i/2}\right) & , if \begin{cases} < 0, \ right\,site. \\ > 0, \ left\,side. \end{cases} \\ 0 & , vehicle\,on\,center, \end{cases}$$

where $\widehat{\psi}$ is the yaw angle estimated for use as the steering control angle in the Gazebo simulator; x_{offset} is the distance from the vehicle's center to the line between the two detected lanes; $(x_{1,2(L)}, y_{1,2(L)})$ and $(x_{1,2(R)}, y_{1,2(R)})$ are the start points and endpoints of the left lane and the right lane; W_i and H_i are width and height of the image input (unit: pixel). Fig. 5 shows the sample coordinates of the system on the right side (a) and the left side (b).

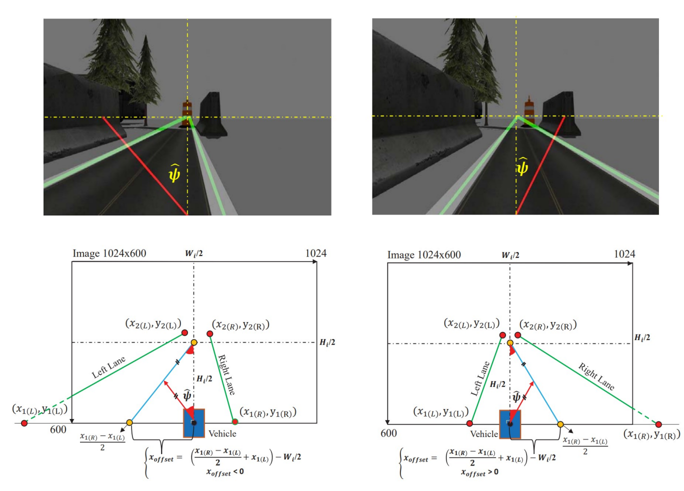


Fig. 5. Sample Coordinates of the System on the Right Side (a) and the Left Side (b).

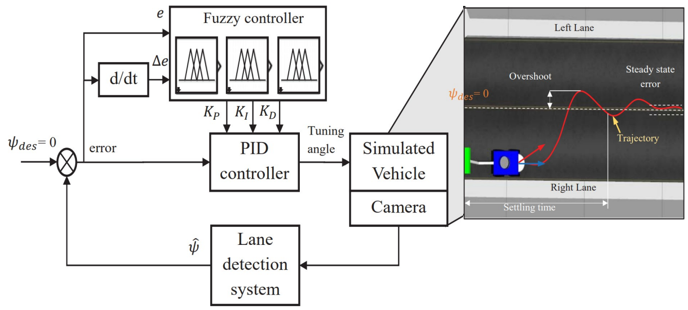


Fig. 6. Fuzzy PID Control Algorithm Structure Diagram for our Vehicle Autonomous System.

TABLE I. Fuzzy Control Rules

K_p		Δe				
		N	NM	Z	PM	P
e	N	V_h	H	H	M_e	M_e
	NM	V_h	H	H	M_e	L_o
	Z	H	H	M_e	M_e	L_o
	PM	H	H	M_e	M_e	L_o
	P	H	H	M_e	M_e	L_o
K_i		Δe				
		N	NM	Z	PM	P
e	N	Z_e	Z_e	M_e	M_e	M_e
	NM	Z_e	M_e	H	M_e	M_e
	Z	M_e	H	H	M_e	M_e
	PM	H	H	H	M_e	Z_e
	P	H	H	M_e	Z_e	Z_e
K_d		Δe				
		N	NM	Z	PM	P
e	N	L_oL	M_e	M_e	Z_e	Z_e
	NM	M_e	M_e	Z_e	Z_e	H
	Z	M_e	Z_e	Z_e	M_e	H
	PM	Z_e	Z_e	M_e	M_e	V_h
	P	Z_e	H	H	V_h	V_h

IV. Steering Control System

The steering control system controls the vehicle to keep the center of the lane. To achieve this, the desired steering angle (ψ_{des}) must be controlled at 0. However, the damping effect of ordinary PI and PID controllers is not good enough to keep the vehicle running smoothly and stably, so a fuzzy PID controller is proposed to tune the PID parameters for further improvement. Fig. 6 shows the fuzzy PID control algorithm structure diagram for our vehicle autonomous system. A mathematical equation for controlling the steering angle of a typical PID controller is expressed in (5).

$$\psi_{Tuning}(t) = K_p e(t) + K_i \int_0^t e(t) + K_d \frac{de}{dt} \quad (5)$$

where $K_{p,I,d}$ are the proportional gain, integral gain, and derivative gain; ψ_{Tuning} is the tuning steering angle; e is the angle error between the estimated steering angle ($\hat{\psi}$) and the desired steering angle (ψ_{des} = 0).

The fuzzy logic controller is designed with two input signals the error and the derivative of the error, while the three outputs are K_p,K_i, and K_d. The range of e, and Δe are limited as -45,45 and -10,10, respectively. The fuzzy rule base is shown in Table I. The membership functions are N (Negative); NM (Negative medium); Z (Zero); PM (Positive medium); P (Positive) for error input, and Z_e(Zero); L_o(Low); M_e(Medium); H(High); V_h(Very high) for parameters output. The Segeno model is applied to the fuzzy logic structure to obtain the best value for PID parameters.

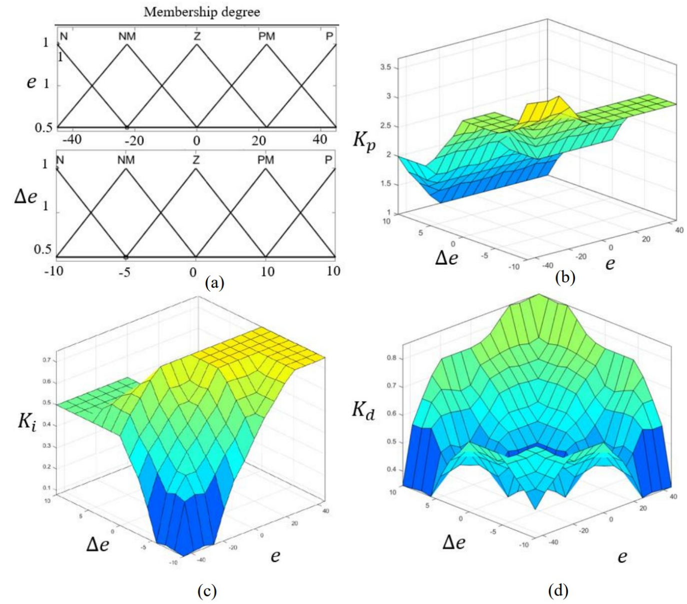


Fig. 7. Fuzzy Logic. (a) Membership Function of e and Δe. (b), (c) and (d) View of the Fuzzy Rule-Base of K_p,K_i, and K_d.

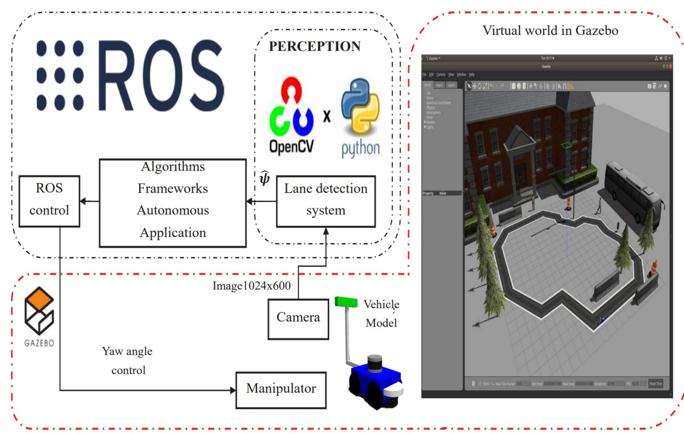


Fig. 8. Architecture of Simulation System.

By using the Scikit-Fuzzy library for the Python computing language, the steering angle controller was tuned based on the designed rule, so that the best dynamic response of the vehicle is achieved with the smallest overshoot and steady-state error when comparing centerline lane keeping. Fig. 7(a) shows the membership function and Fig. 7(b)-(d) the rule base of the fuzzy logic controller.

The simulation results of vehicle operation in the Gazebo environment are provided later to demonstrate the effectiveness of the two proposed systems.

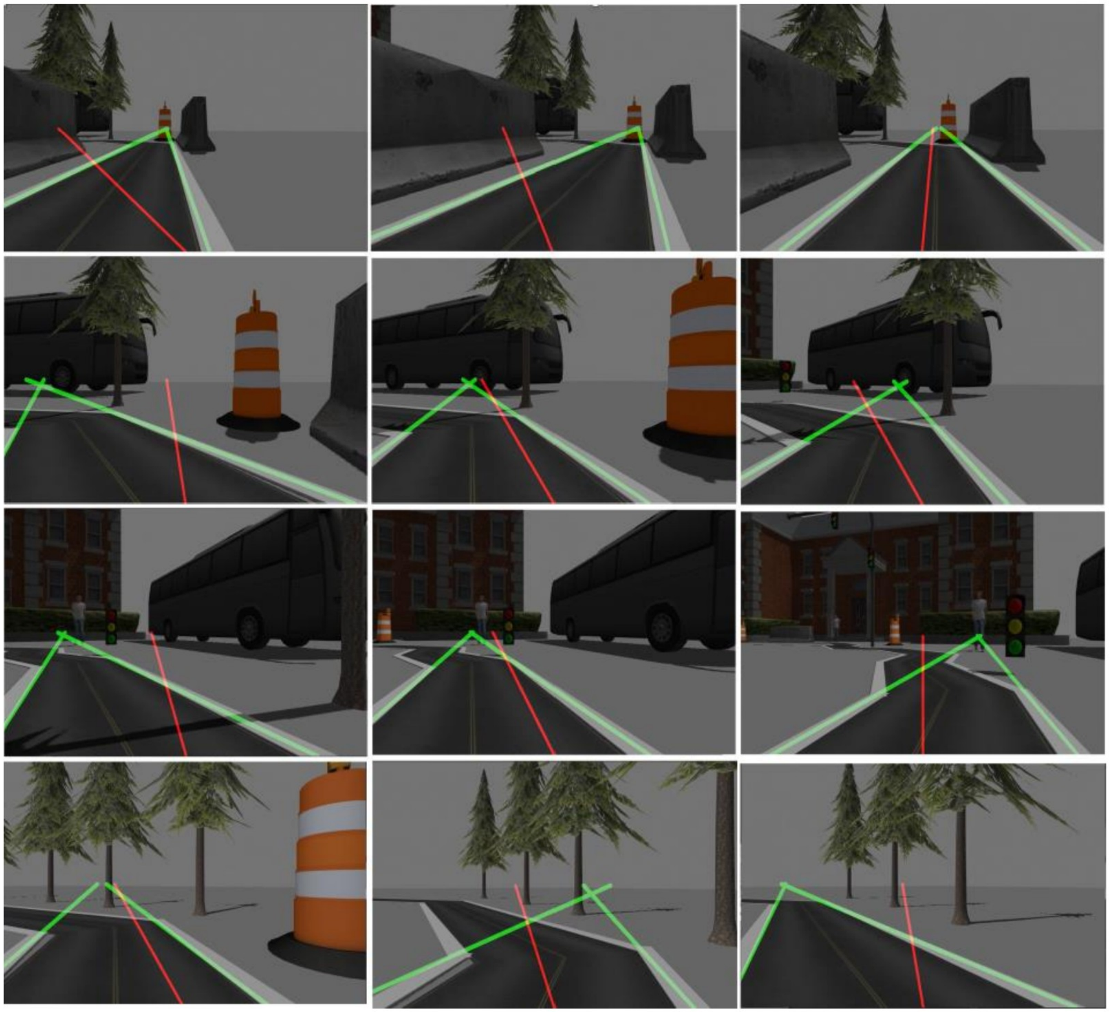


Fig. 9. Results of our Lane Detection System in the Virtual World.

V. Simulation and Results

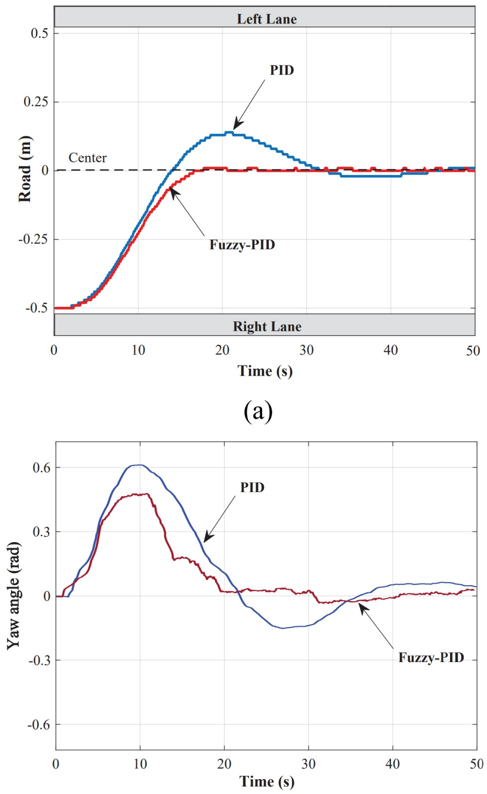
The virtual world is designed by using the software Gazebo/ROS. Fig. 8 shows the vehicle model, Gazebo environment, and the software used to apply the two proposed algorithms. We have performed two kinds of experiments to verify our lane detection algorithm and our steering control algorithm with the operating system Ubuntu 18.04.4, and CPU: Intel i7 3.4 GHz, GPU: Nvidia GTX 1650-4GB, RAM 16 GB. Our algorithm is implemented in Python language using the open source OpenCV and rospy library. The camera is simulated from the actual parameters of the WGE100 camera. The image size is 1024×600; the image was recorded at 30 frames per second. We tested that the lane detection algorithm is applied according to [17].

The lane detection results are like previous approaches. In the virtual world, the image data obtained with two-lane detection results are correct, as shown in Fig. 9. We also achieve good results for the estimated yaw angle when applying the steering angle estimation algorithm in Section 3.

We also verify the performance of the proposed steering angle controller by reading the odometry information of vehicle autonomous over ROS including trajectory and yaw angle. Fig. 10 and Fig. 11 show the lane-keeping performance of the PID steering controller, and the fuzzy-PID steering controller under low-speed, and high-speed conditions. It is easy to see that the proposed method can minimize the desired angle error and reduce the damping effects better than another method. Table II shows the performance comparison of the PID controller and fuzzy-PID controller. The overshoot of the fuzzy-PID controller is less than 5%, however, with the conventional controller, it is 25% more than 5 times. Therefore, we concluded that the proposed assist system obtains higher accuracy of lane-keeping performance and better stability.

VI. Conclusion

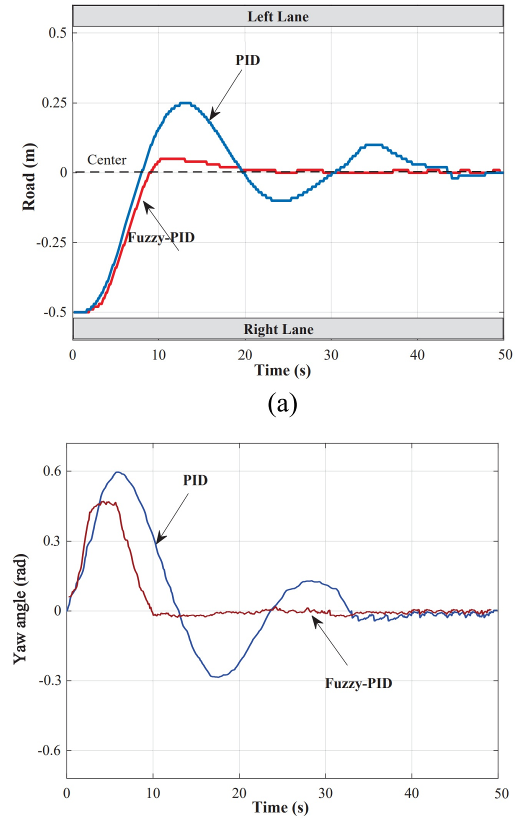
In this research, we develop a lane-keeping assist for an autonomous vehicle. The algorithm is based on a sequence of image processing, filtering, determination ROI, line segments detection, and steering angle estimation to detect lanes in the virtual world. All lanes are detected in still images at a high rate of 30Hz. Then, the steering controller with the fuzzy-PID algorithm is also proposed to reduce the steering angle error and minimize the damping effects. In the future, the algorithms



(a)

(b)

Fig. 10. Performance Comparison of the PID Steering Controller and Fuzzy-PID Steering Controller at Low Speed (2m/s). (a) The Vehicle's Trajectory on the Road, and (b) the Yaw Angle of the Vehicle.



(a)

(b)

Fig. 11. Performance Comparison of the PID Steering Controller and Fuzzy-PID Steering Controller at High Speed (4m/s). (a) The Vehicle's Trajectory on the Road, and (b) the Yaw Angle of the Vehicle.

will be tested and evaluated on real systems. In addition, deep learning algorithms will also be studied and applied.

TABLE II. OVERSHOOT AND SETTLING TIME OF THE PERFORMANCE COMPARISON

	Overshoot (%)		Settling time (s)	
	Low speed	High speed	Low speed	High speed
PID	25%	51%	28s	43s
Fuzzy-PID	1%	5%	12s	15s

REFERENCES

[1] R. Ramani, *Vehicle Dynamics and Control*, Springer, 2012.

[2] Y. Matsushi, and J. Mijura, *On-line Road Boundary Modeling with Multiple Sensory Features Flexible Road Model and Particle Filter*, Robotic and Autonomous Systems, vol. 59, pp. 274-284, 2011.

[3] Y. Kang, C. Roh, S. B.Suh, and B. Song *A Lidar-Based DecisionMaking Method for Road Boundary Detection Using Multiple Kalman Filters*, IEEE Transactions on Industrial Electronics, pp. 4360 – 4368, 2012.

[4] H. Fahmy, G. Baumann, M. AbdelGhany, and H. Mostafa.: *V2V-Based Vehicle Risk Assessment and Control for Lane-Keeping and Collision Avoidance*, in IEEE International Conference on Microelectronics (ICM 2017), Beirut, Lebanon, pp. 61-65, 2017.

[5] J. Huang, P. K. Choudhury, S. Yin and L. Zhu, *Real-Time Road Curb and Lane Detection for Autonomous Driving Using LiDAR Point Clouds,* in IEEE Access, vol. 9, pp. 144940-144951, 2021

[6] Pallavi V. Ingale and Prof. K. S. Bhagat.: *Comparative Study of Lane Detection Techniques*, in International Journal on Recent and Innovation Trends in Computing and Communication, vol. 4, no. 5, 2016.

[7] Ammu M Kumar and Philomina Simon.: *Review of Lane Detection and Tracking Algorithms in Advanced Driver Assistance System*, in International Journal of Computer Science & Information Technology (IJCSIT), vol. 7, no. 4, 2015.

[8] A. Broggi, P. Medici, P. Zani, A. Coati, and M. Panciroli.: *Autonomous vehicles control in the VisLab intercontinental autonomous challenge, Annual Reviews in Control*, vol. 36, no. 1, pp. 161–171, 2012.

[9] J. Ackermann, J. Guldner, W. Sienel, R. Steinhauser, and V. I. Utkin.: *Linear and Nonlinear Controller Design for Robust Automatic Steering*, IEEE Transactions on Control Systems, vol. 3, no. 1, 1995.

[10] R. Marino, S. Scalzi, and M. Netto.: *Nested PID Steering Control for Lane Keeping in Autonomous Vehicles*, Control Engineering Practice, vol. 19, no. 12, pp. 1459– 1467, 2011.

[11] W. Farag and Z. Saleh, *Tuning of PID track followers for autonomous driving,* 2018 Int. Conf. Innov. Intell. Informatics, Comput. Technol. 3ICT 2018, pp. 1–7, 2018.

[12] S. Surendharan, J. Jennifer Ranjani *Environment conscious automated vehicle navigation system using PID controller,* Indian J, Sci, Technol, 2016.

[13] C.K. Chandni, V.V. Sajith Variyar, K. Guruvayurappan, *Vision based closed loop PID controller design and implementation for autonomous car,* In Proceedings of the 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Udupi, India, pp. 1928–1933, 2017.

[14] A. Simorgh, A. Marashian, and A. Razminia, *Adaptive PID Control Design for Longitudinal Velocity Control of Autonomous Vehicles,* Proc. - 2019 6th Int. Conf. Control. Instrum. Autom. ICCIA 2019, pp. 1–6, 2019.

[15] V. Robila, L. Paulino, M. Rao, I. Li, M. Zhu, and W. Wang, *Design and Implementation of PID-Based Steering Control for 1/10-Scale*

Autonomous Vehicle, 2021 IEEE 12th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), pp. 0758-0762, 2021.

[16] M. E. Abed, M. Aly, H. H. Ammar, and R. Shalaby.: *Steering Control for Autonomous Vehicles Using PID Control with Gradient Descent Tuning and Behavioral Cloning*, 2020 2nd (NILES), pp. 583-587, 2020.

[17] Sharma, A., Kumar, M., Gupta, R.K., Kumar, R.: *Lane Detection using Python* - Anmol Sharma, Maneesh Kumar, Rajesh Kumar – IJIRMPS, vol. 9, 2021.

[18] Sekehravani, E.A., Babulak, E., Masoodi, M.: *Implementing Canny Edge Detection Algorithm for Noisy Image*. Bull. Electr. Eng. Inform, vol. 9, no. 4, pp. 1404–1410, 2020.

[19] P. Subhasri, S. Santhoshkumar and A. Sumath, *Edge Filtering through Recursive Application using Canny Edge Detector algorithm on small sub-blocks in an Image,* 2020 International Conference on Smart Electronics and Communication (ICOSEC), pp. 563-566, 2020.

[20] Messom C. H., Sen Gupta G. and Demidenko S.N.: *Hough Transform Run Length Encoding for Real-Time Image Processing*, IEEE Trans. Instrum. Meas., vol. 56, no. 3, pp. 962-967, 2007.