

Model predictive control combined reinforcement learning for automatic vehicles applied in intelligent transportation system

Vo Thanh Ha¹, Chu Thi Thuy²

¹Department of Cybernetics, Faculty of Electrical and Electronic Engineering, University of Transport and Communications, Hanoi, Vietnam

²Department of Control and Automation, Faculty of Electrical Engineering, University of Economics-Technology for Industries, Hanoi, Vietnam

Article Info

Article history:

Received Apr 24, 2023

Revised Dec 9, 2023

Accepted Dec 22, 2023

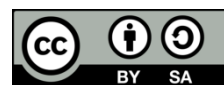
Keywords:

Automatic vehicles
Intelligent transportation system
Model predictive control
Reinforcement learning
Reinforcement learning-
proximal policy optimization

ABSTRACT

This paper presents the design of model predictive control (MPC) combined reinforcement learning (RL) applied in an intelligent transportation system (ITS). The car is to follow the reference path by the MPC control, and its parks in the parking by RL has been trained. The MPC controller constantly moves the vehicle along the reference path while the MPC algorithm searches for an empty parking spot. Meanwhile, the reinforcement learning-proximal policy optimization (RL-PPO) control will perform parking on demand if the MPC finds a parking position. This hybrid controller can quickly implement programming on MATLAB software by writing code. Furthermore, this hybrid controller simultaneously performs precise detection and avoidance of obstacles in tight parking spaces. The correctness of the theory is demonstrated through MATLAB/Simulink.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Vo Thanh Ha

Department of Cybernetics, Faculty of Electrical and Electronic Engineering

University of Transport and Communications

N03, Cau Giay District, Hanoi, Vietnam

Email: vothanhha.ktd@utc.edu.vn

1. INTRODUCTION

The automatic parking system is one of the modern driver support services in the intelligent traffic system. The role of this system is to assist the driver in safely and quickly parking the vehicle [1], [2]. Therefore, this system can reduce the skill requirements of the driver and human-caused accidents such as car collisions. Modern driver assistance system technologies typically perform three base steps: object detection (obstacle), decision-making, and control [3]–[5]. The automatic parking system consists of three parts: parking environment recognition, route planning [6]–[8], and track tracking [9]–[11]. The parking system controller automatically obtains information about parking positions and obstacles through various sensors, such as ultrasonic sensors, cameras, wheel speed sensors, and angle sensors driving [12]–[15]. Sensors measure the vehicle's distance from obstacles, real-time visual data, current vehicle speed, and steering angle. Finally, the controller decides whether the autonomous vehicle should stop or continue from the information through the multifunction sensor [16], [17]. Path planning, control, and monitoring constantly interact in the automatic car parking system. In particular, the path monitoring control algorithm is one of the critical technologies of the automated parking system. This supervisory control algorithm must ensure the accuracy of road monitoring, driving comfort when the vehicle changes direction and the driving position and orientation at the end of the parking operation. Therefore, this automated parking technology attracts many scientists to research and propose related control algorithms in theory and experiment.

The document [18] proposed an automatic parking line control method that considers time delay, solving the problem of the control model of the traditional automated parking system that is not related to the parking system vehicle control delay. Another study [19] has proposed a semi-automatic parking assistance system based on the driver navigation area, which can recognize the environment information in real-time through the sensor to sense the environment and optimize the parking space. Optimize parking routes to avoid collisions. In another study [20], a fuzzy controller that supports automatic parking without a model was proposed to monitor the parking path. In addition, the improved research [21], [22] combines fuzzy control with neurons. This controller only needs to know the parking configuration. The vehicle will monitor the path and assist in correct parking. The algorithms can control and monitor the vehicle's automatic path and parking on demand. However, these controllers cannot coordinate the vehicle speed and steering wheel control with the change of parking path during the vehicle's movement.

Therefore, the control solutions are still limited to the accuracy of the control and monitoring path. Thus, the article will propose a solution to apply a model predictive controller (MPC) using a vehicle dynamics model to predict how the vehicle will react to a particular control action within the expected range. This behavior is similar to the fact that a driver understands and predicts the behavior of a driver's vehicle. To perform optimal control motion computation, this MPC controller needs to consider all input and output constraints on the system, such as speed limit, safe distance after, physical limit of vehicle, maximum steering angle, and obstacles for the controller to avoid [23]–[26]. This paper will present a controller design that combines MPC for cars to follow the reference path in the parking lot with the reinforcement learning method (RL) trained to perform the parking maneuver. The MPC controller moves the vehicle constantly along the reference path while the MPC algorithm searches for an empty parking spot. Once the MPC control algorithm has found the location, the RL controller will perform the parking request. This hybrid controller performs simultaneous obstacle detection and avoidance in tight parking spaces without human intervention. This system uses an adaptive model predictive controller that updates both the predictive model and the mixed input and output constraints at each control interval. The correctness of the theory is proven through MATLAB simulation.

The article is expressed in five parts. The first part introduces the study of automatic parking vehicle control. In the next part, the mathematical model of the car is given. Based on this mathematical model, an MPC controller combined with RL for vehicle movement and obstacle avoidance in section 3. The correctness of the control solution is shown through the proof. MATLAB simulation in section 4. Finally, the paper makes conclusions about the main features of the automatic parking solution and future research directions.

2. MATHEMATICAL MODEL OF CAR

In Figure 1, the article employs a rectangular automobile model with dimensions of 5 meters in length and 2 meters in breadth. The vehicle can assist in overcoming obstacles using a Lidar sensor. This sensor calculates how far the car is from any obstacles in its lane and in front of it. Blocks might be stationary, like a giant pit, or moving, like a slowly driving car. The most frequent driver behavior is briefly switching lanes, crossing an obstruction, and retracing their steps.

In Figure 1, it is noticed that the car coordinate model has four state variables such as x , y is the central position of the x , y -axis of the car; v is the speed of the vehicle; θ is the tilt angle of the car (value 0 when turning to the east, counterclockwise in the positive direction). Two variables interact such as T is the throttle (positive value when accelerating, negative when decelerating), and δ is the steering angle (value 0 when aligned with the car, positive counterclockwise); C_L is the length of the vehicle. The paper uses a simple non-linear model to describe car dynamics as (1):

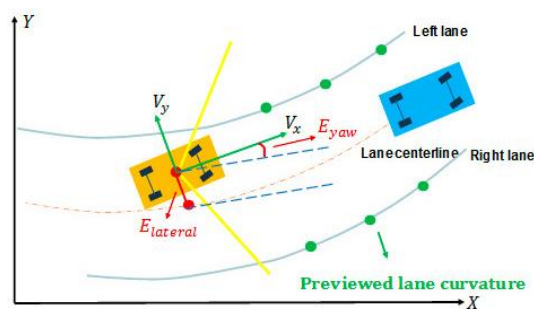


Figure 1. Coordinate model of cars

$$\begin{aligned}
 \dot{x} &= \cos(\theta) v \\
 \dot{y} &= \sin(\theta) v \\
 \dot{\theta} &= (\tan(\delta) C_L) \cdot v \\
 \dot{v} &= 0.5T
 \end{aligned} \tag{1}$$

According to Jacobian about the nonlinear state model used to build the linear predictive model at the operating point, the (2) is created:

$$\begin{aligned}
 \dot{x} &= -v \sin(\theta) \cdot \dot{\theta} + \cos(\theta) \cdot \dot{v} \\
 \dot{y} &= v \cdot \cos(\theta) \cdot \dot{\theta} + \sin(\theta) \dot{v} \\
 \dot{\theta} &= (\tan(\delta) C_L) \cdot v + (v(\tan(\delta)^2) + 1)(\tan(\delta) C_L) \\
 \dot{v} &= 0.5T
 \end{aligned} \tag{2}$$

3. MPC CONTROLLER DESIGN AND RL-PPO ADVANCED LEARNING

3.1. MPC model predictive controller

Model prediction controller uses object models, input and output noise to predict and estimate the state. The model structure used in the MPC controller is shown in Figure 2. The model prediction controller calculates the optimal control input by minimizing a cost function that penalizes deviations from the desired state trajectory. The predicted state is then used to update the control input in real time, allowing the controller to track the desired course accurately.

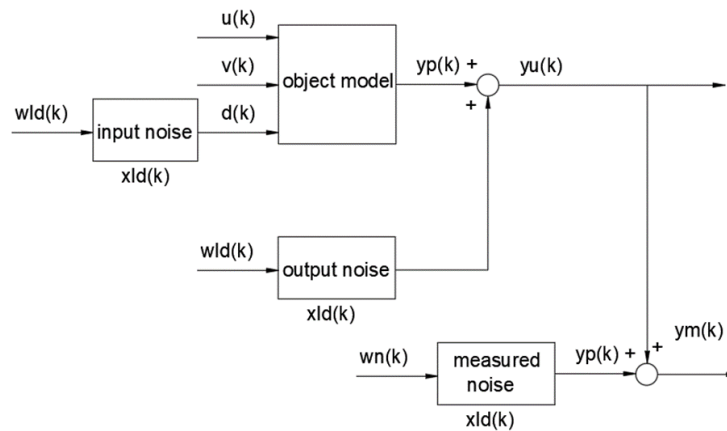


Figure 2. The MPC controller architecture

3.1.1. Object model

The car state model is written as the (3):

$$\begin{aligned}
 x_p(k+1) &= A_p x_p(k) + B S_i u_p(k) \\
 y_p(k) &= S_o^{-1} C x_p(k) + S_o^{-1} D S_i u_p(k)
 \end{aligned} \tag{3}$$

Where: x_p, y_p is the input and output variable of the object; A_p, B, C are state space matrices with constant zero delay; S_i is the input diagonal matrix; S_o is the output diagonal matrix; diagonal matrix of output scale factors; x_p is the state vector that includes all delay states; u_p is a vector of input variables consisting of manipulated variables, measured noise, and unmeasured input noise; y_p is a vector of output variables. State model (3) does not include input and output noise. So the car state model is rewritten as the (4):

$$\begin{aligned}x_p(k+1) &= A_p x_p(k) + B_{pu}(k) + B_{pv}(k) + B_{pd}(k) \\y_p(k) &= C_p x_p(k) D_{pu}(k) + D_{pv}(k) + D_{pd}(k)\end{aligned}\quad (4)$$

Where: $C_p = S_0^{-1}C$, B_{pu}, B_{pv}, B_{pd} is a parameter of BS_i ; D_{pu}, D_{pv}, D_{pd} is a parameter of $S_0^{-1}DS_i$; $(k), v(k), d(k)$ are the measured and unmeasured input noises.

The MPC controller is limited so $D_{pu} = 0$, means that the MPC controller does not allow direct transmission from any controlled variable to any output of the control object. Matrix A, B, C and D is determined as follows:

$$A = \begin{bmatrix} A_p & B_{pd}C_{id} & 0 & 0 \\ 0 & A_{id} & 0 & 0 \\ 0 & 0 & A_{od} & 0 \\ 0 & 0 & 0 & A_n \end{bmatrix}; B = \begin{bmatrix} B_{pu} & B_{pv} & B_{pd}D_{id} & 0 & 0 \\ 0 & 0 & B_{id} & 0 & 0 \\ 0 & 0 & 0 & B_{od} & 0 \\ 0 & 0 & 0 & 0 & B_n \end{bmatrix}$$

$$C = C_p \quad D_{pd}C_{id} \quad C_{od} \begin{bmatrix} C_n \\ 0 \end{bmatrix}; D = \begin{bmatrix} D_{pv} & D_{pd}D_{id} & D_{od} \end{bmatrix} \begin{bmatrix} D_n \\ 0 \end{bmatrix}$$

3.1.2. Input noise model

The input noise model is determined by the (5):

$$x_{id}(k+1) = A_{id}x_{id}(k) + B_{id}w_{id}(k) \quad (5)$$

$$d(k) = C_{id}x_{id}(k) + D_{id}w_{id}(k) \quad (6)$$

In there: A_{id}, B_{id}, C_{id} are constant state matrices; $x_{id}(k)$ is the vector of the measured input noise when $n_{xid} \geq 0$; $d_k(k)$ is the vector of input noise n_d can't measure; w_{id} is the input noise vector whose mean value is 0, when $n_{id} \geq 1$.

3.1.3. Output noise model

The output noise model is determined by the (7):

$$\begin{aligned}x_{od}(k+1) &= A_{od}x_{od}(k) + B_{od}w_{od}(k) \\y_{od}(k) &= C_{od}x_{od}(k) + D_{od}w_{od}(k)\end{aligned}\quad (7)$$

In there: $A_{od}, B_{od}, C_{od}, D_{od}$ are constant state matrices; $x_{od}(k)$ is the vector of the measured output noise when $n_{xod} \geq 0$; $d_k(k)$ is the vector of the output noise n_y can't measure; w_{od} is the vector of input noise whose mean value is 0, when $n_{od} \geq 1$.

3.1.4. Measured noise pattern

The measured noise pattern is determined by the (7):

$$x_n(k+1) = A_n x_n(k) + B_n w_n(k) \quad (8)$$

Where: A_n, B_n, C_{od} are constant state matrices; $x_n(k)$ is the vector of the measured noise when $n_{xn} \geq 0$; $y_n(k)$ is the output noise vector n_{ym} ; $w_n(k)$ is the input noise vector whose mean value is 0, when $n_n \geq 1$.

3.2. Reinforcement learning

The structural principle of the reinforcement learning strategy is depicted in Figure 3. Machine learning's RL area investigates how an agent in a given environment should decide what behaviors to perform to maximize a particular reward over the long term. The RL algorithms seek a policy connecting the world's states to the actions the agent should do in each state. The RL algorithms used in this context are closely related to dynamic programming methods since the environment is often represented as a limited set of conditions. Unlike supervised learning, RL lacks good input/output pairings and does not explicitly assess near-optimal behaviors as true or false. Additionally, the action in question involves the pursuit of a balance between discovery (an untried condition) and exploitation (a known form). A set of "rewards" with no value is used to educate agents to execute a set of actions in a group of environmental conditions in the RL paradigm. The reward evaluates how well the last series of steps achieved the task goal. The agent has two parts, including a training algorithm and a policy.

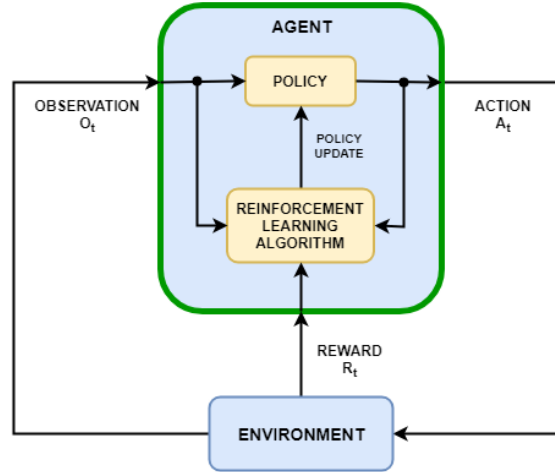


Figure 3. Structure of RL

3.2.1. Automated parking design

The parking space for training is 22.5 m long and 20 m wide, with the target at the horizontal center. Shaft fault observation position X_e, Y_e of the car with the desired vehicle position. Value \cos, \sin true to the actual tilt angle θ of the vehicle and the lidar sensor. To determine the distance of the ego vehicle from other cars in the environment. The lidar sensor is modeled using geometric relationships. The length of the Lidar sensor is measured along 12 radial lines from the center of the self-driving vehicle. When a line crosses an obstacle, Lidar shows the size of the block to the car. The space that can be measured along any road segment is 6 m.

Parking speed is 2 m/s. The driving angle is limited from $\pm 45^\circ$ to $\pm 15^\circ$. A vehicle is considered parked if the required position and posture errors are within the specified tolerances of ± 0.75 m (location) and ± 10 degrees (direction). The process of stopping the training if the vehicle goes out of the bounds of the training parking area or collides with an obstacle, or is successfully parked. The reward at each time t is determined by (7):

$$r_t = 2e^{-(0.05X_e^2 + 0.04Y_e^2)} + 0.5e^{-40\theta_e^2} - 0.05\delta^2 + 100f_t - 50g_t \quad (9)$$

where: X_e, Y_e, θ_e and the errors in the position and angle of the car's inclination determined from the required position; δ is the steering angle; f_t (0 and 1) indicates whether the vehicle is parked or not at the time t ; g_t (0 and 1) indicates whether the vehicle collides with an obstacle at time t .

Coordinate transformations when observing vehicle positions (X, Y, θ) of different parking spot locations are determined as follows:

- Coordinates 1-14: no conversion
- Coordinates 15-22: $\bar{X} = Y, \bar{Y} = X, \bar{\theta} = \theta - \pi/2$
- Coordinates 23-36: $\bar{X} = 100 - X, \bar{Y} = 60 - Y, \bar{\theta} = \theta - \pi$
- Coordinates 37-40: $\bar{X} = 60 - X, \bar{Y} = X, \bar{\theta} = \theta - \frac{3\pi}{4}$
- Coordinates 41-52: $\bar{X} = 100 - X, \bar{Y} = 30 - Y, \bar{\theta} = \theta + \pi$
- Coordinates 53-64: $\bar{X} = X, \bar{Y} = Y - 28, \bar{\theta} = \theta$

3.2.2. Augmented agent design

The article proposes to design RL agents based on asymptotic proximal policy optimization (PPO). This is an online, model-free, gradient training method. This algorithm is a kind of policy gradient training that alternates between sampling the data through the environment interaction and optimizing the objective function using a random gradient function. The PPO RL agent is created by a neural network consisting of an input layer that receives information from the observer and an output layer. This neural network is trained empirically as agent training. The number of iterations steps is set to 200, and the number of training episodes is 150. The learning rate parameter of 0.2 improves the stability of the training. And a discount factor of 0.997 to maximize demand. The loss factor is 0.01. Calculate the output variance using the GAE advantage estimation method of 0.95. Conduct PPO training as follows: train up to 10000 episodes, each lasting up to 200-time steps. Movement stops when maxing out the target average of 80 episodes or more.

4. RESULTS OF SIMULATION AND ASSESSMENT

Automatic parking based on an MPC controller combined with RL-PPO is modeled according to the structure shown in Figure 4. They simulated the MPC controller with sampling time $T_s = 0.1$. Output weight (2, 2, 3). The control variable constraints are [min là -5; max +5]. The obstacle in this paper is the assumption of an immobile object in the middle of the center lane of the same size car. The MPC controller matrices have the following values:

$$A_d = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}; B_d = \begin{bmatrix} 0.1 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}; C_d = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}; D_d = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

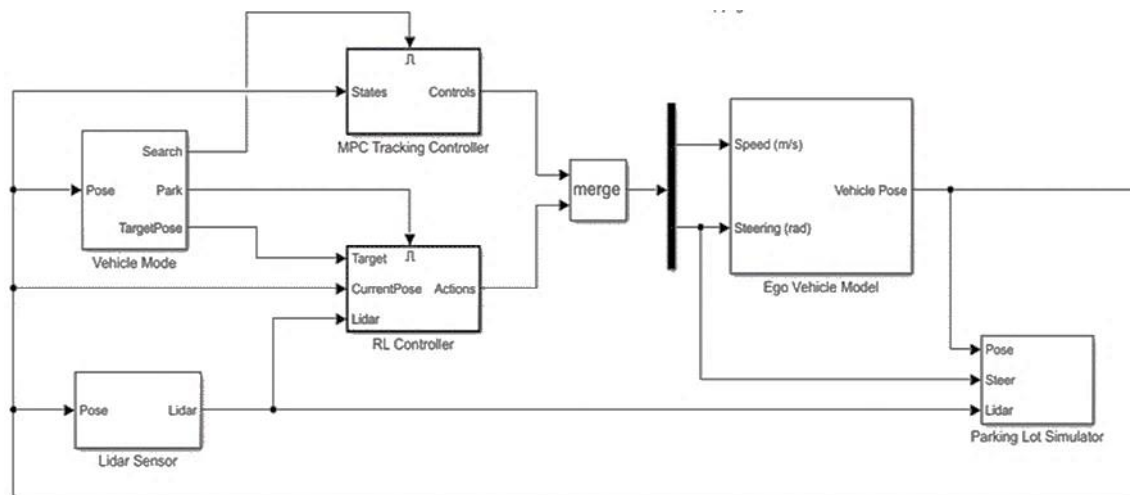


Figure 4. MATLAB simulation structure of automatic parking based on the MPC controller combined with RL-PPO

The RL results are shown in Figure 5 and the automatic parking design results are in Figure 6. Figure 5 shows that the average number of steps achieved through each episode of 80 random executions, the training time is 3551.2 seconds. Observation Figure 5 shows that in the first 200 episodes, the car only went under 20 steps. From the 200th to the 900th episode, the vehicle's object avoidance continuously improved and increased step count. While the step maximum is reached starting with the 900th episode, this maximum is not always come in subsequent episodes. This level becomes more and more likely to go as the episode increases. The reasons for the results are: in the first episodes, the car did not know how to avoid static and dynamic obstacles, so it caused a very early collision. The show will stop the attack if a crash occurs and the vehicle speed is constant at 2 m/s. Therefore, the low step count corresponding to Figure 5 indicates that the vehicle is not responding well to obstacle avoidance. The more you train, that is, for more extensive episodes, the number of steps increases over time. This means that even though the vehicle moves continuously in an environment with static and dynamic obstacles, it can regulate the knowledge it has learned and make increasingly accurate decisions, avoiding the obstacles. Body. The step value reaches saturation at 1000 with increasing probability showing that the vehicle can operate well in a complex environment and achieve the maximum number of steps in future training times. This result proves that the algorithm has been installed successfully. With the Q-learning RL algorithm, the vehicle was able to train itself to achieve the skill of avoiding static and moving objects.

Figure 6 shows the advantage of an RL controller the vehicle has moved along the path and is in the correct parking position. The elapsed time of the car is 10.8 s. However, the part of the parked car is still wrong on the Y axis (meaning the parked car is slanted).

The response of the x and y positions and the tilt angle of the vehicle are shown in Figure 7. Through simulation results, the required location is (50.125; 4.9 -1.5709). Thus, the car reaches the target position within the allowable error of +/-0.75 m (site) and +/-10 degrees (direction) valid request. The assist feature helps the ego to stop after 10.8 seconds. The response of the driving angle is shown in Figure 8. From this simulation result, the steering angle shows that the controller reaches a steady state after about 4.2 seconds with a vehicle speed of 2 m/s by the requirements.

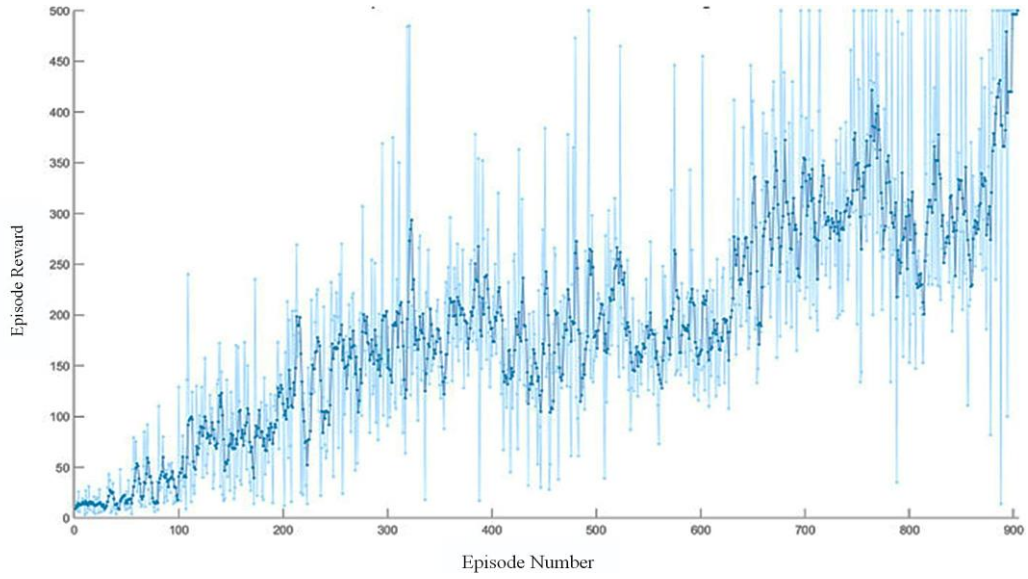


Figure 5. Training process of RL-PPO

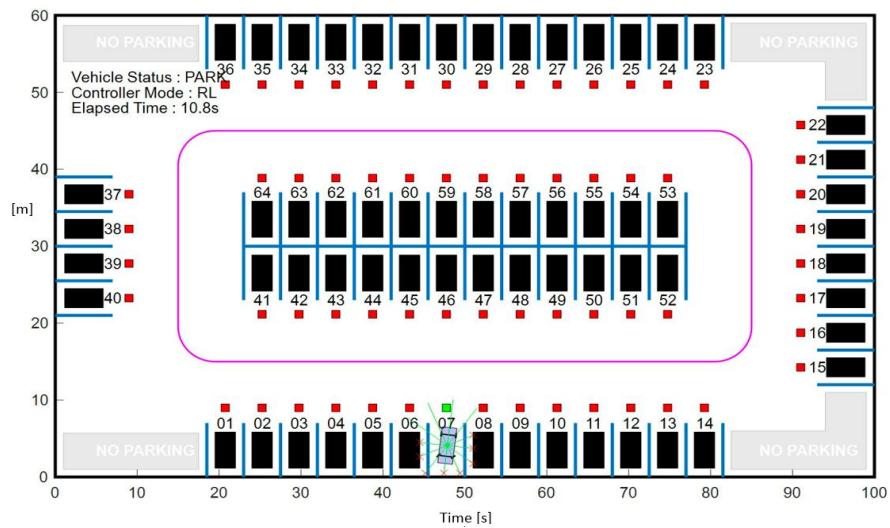


Figure 6. Automated parking lot model

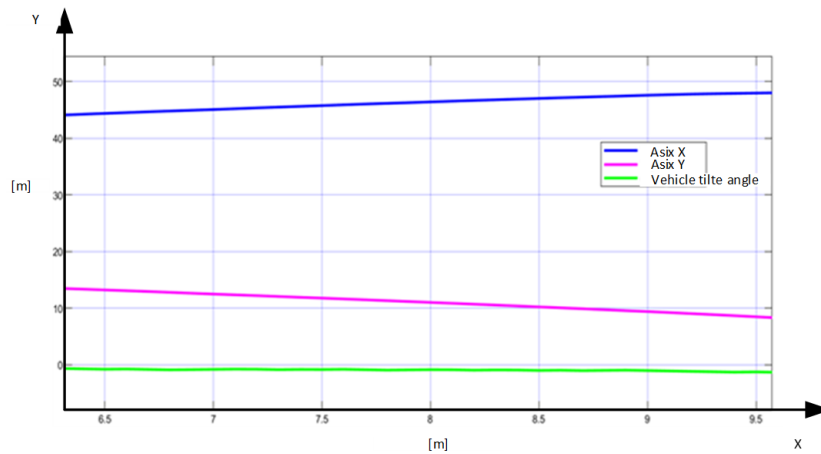


Figure 7. x, y axis response and vehicle tilt angle

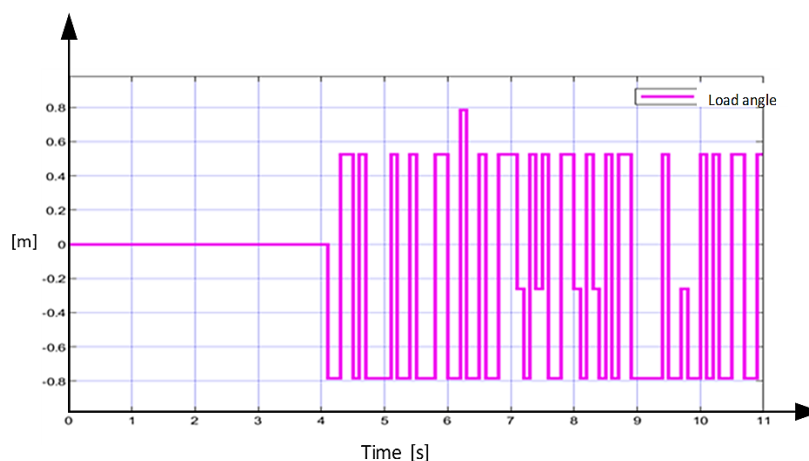


Figure 8. Driving angle response

5. CONCLUSION

This paper presents a successful controller design that combines the MPC model predictive control and RL method RL-PPO. This integrated controller has made the vehicle move to avoid obstacles and park the car as required with fast calculation time. The success of this research work has partly contributed to intelligent traffic systems, improving driver support services and traffic system management and administration agencies. However, to increase the convincingness and reliability of this smart control solution, the research work needs to be compared with other control methods such as deep learning RL (Q-deep learning), adaptive fuzzy tree (fuzzy tree), and the research results will be tested experimentally in the future. Furthermore, the comparison with other control methods will provide a comprehensive understanding of the strengths and weaknesses of the proposed integrated controller. Additionally, experimental testing in real-world scenarios will offer valuable insights into its practical applicability and performance under diverse conditions. This holistic approach will enhance the robustness and relevance of the intelligent control solution, thereby contributing to the advancement of intelligent transportation systems and autonomous vehicle technology.




REFERENCES

- [1] C. Vidal, O. Gross, R. Gu, P. Kollmeyer, and A. Emadi, "xEV Li-Ion Battery Low-Temperature Effects—Review," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 5, pp. 4560–4572, May 2019, doi: 10.1109/TVT.2019.2906487.
- [2] V. T. Ha, "Torque Control of an In-Wheel Axial Flux Permanent Magnet Synchronous Motor using a Fuzzy Logic Controller for Electric Vehicles," *Engineering, Technology & Applied Science Research*, vol. 13, no. 2, pp. 10357–10362, Apr. 2023, doi: 10.48084/etasr.5689.
- [3] G. Hong-bo, Z. Xin-yu, Z. Tian-lei, L. Yu-chao, and L. De-yi, "Research of intelligent vehicle variable granularity evaluation based on cloud model," *Acta Electronica Sinica*, vol. 44, no. 2, pp. 365–374, 2016, doi: 10.3969/j.issn.0372-2112.2016.02.018.
- [4] L. Figueiredo, I. Jesus, J. A. T. Machado, J. R. Ferreira, and J. L. M. de Carvalho, "Towards the development of intelligent transportation systems," in *ITSC 2001. 2001 IEEE Intelligent Transportation Systems. Proceedings (Cat. No.01TH8585)*, IEEE, 2001, pp. 1206–1211, doi: 10.1109/ITSC.2001.948835.
- [5] H. Gao, B. Cheng, J. Wang, K. Li, J. Zhao, and D. Li, "Object Classification Using CNN-Based Fusion of Vision and LIDAR in Autonomous Vehicle Environment," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 9, pp. 4224–4231, Sep. 2018, doi: 10.1109/TII.2018.2822828.
- [6] D. Li and H. Gao, "A Hardware Platform Framework for an Intelligent Vehicle Based on a Driving Brain," *Engineering*, vol. 4, no. 4, pp. 464–470, Aug. 2018, doi: 10.1016/j.eng.2018.07.015.
- [7] M. Cheon, W. Lee, C. Yoon, and M. Park, "Vision-Based Vehicle Detection System With Consideration of the Detecting Location," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 3, pp. 1243–1252, Sep. 2012, doi: 10.1109/TITS.2012.2188630.
- [8] R. O. Chavez-Garcia and O. Aycard, "Multiple Sensor Fusion and Classification for Moving Object Detection and Tracking," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 2, pp. 525–534, Feb. 2016, doi: 10.1109/TITS.2015.2479925.
- [9] G. Xie, H. Gao, L. Qian, B. Huang, K. Li, and J. Wang, "Vehicle Trajectory Prediction by Integrating Physics- and Maneuver-Based Approaches Using Interactive Multiple Models," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 7, pp. 5999–6008, Jul. 2018, doi: 10.1109/TIE.2017.2782236.
- [10] G. Hongbo, X. Guotao, Z. Xinyu, and C. Bo, "Autonomous parking control for intelligent vehicles based on a novel algorithm," *The Journal of China Universities of Posts and Telecommunications*, vol. 24, no. 4, pp. 51–56, Aug. 2017, doi: 10.1016/S1005-8885(17)60223-1.
- [11] H. G. Jung, Y. H. Lee, and J. Kim, "Uniform User Interface for Semiautomatic Parking Slot Marking Recognition," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 2, pp. 616–626, Feb. 2010, doi: 10.1109/TVT.2009.2034860.
- [12] A. Razinkova, H.-C. Cho, and H.-T. Jeon, "An Intelligent Auto Parking System for Vehicles," *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 12, no. 3, pp. 226–231, Sep. 2012, doi: 10.5391/IJFIS.2012.12.3.226.
- [13] Y. Hua, H. Jiang, Y. Cai, X. Zhang, S. Ma, and D. Wu, "Path Tracking Control of Automatic Parking Cloud Model considering the




- Influence of Time Delay,” *Mathematical Problems in Engineering*, vol. 2017, pp. 1–14, 2017, doi: 10.1155/2017/6590383.
- [14] H. Wang, T. Zhang, X. Zhang, and Q. Li, “Observer-Based Path Tracking Controller Design for Autonomous Ground Vehicles With Input Saturation,” *IEEE/CAA Journal of Automatica Sinica*, vol. 10, no. 3, pp. 749–761, Mar. 2023, doi: 10.1109/JAS.2023.123078.
- [15] B. Guo-xing, L. Wei-dong, L. Li, M. Yu, G. Qing, and L. Kai-lun, “Current status and progress of path tracking control of mining articulated vehicles,” *Chinese Journal of Engineering*, vol. 43, no. 2, pp. 193–204, 2021, doi: 10.13374/j.issn2095-9389.2020.07.14.003.
- [16] H. Van Hasselt, A. Guez, and D. Silver, “Deep Reinforcement Learning with Double Q-Learning,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, Mar. 2016, doi: 10.1609/aaai.v30i1.10295.
- [17] N. M. Nakrani and M. M. Joshi, “A Comparative Analysis Between Proposed Neuro-fuzzy, Fuzzy, and Heuristic-Neuro-fuzzy Controller for Autonomous Vehicle Parking in the Dynamic Environment,” *SN Computer Science*, vol. 3, no. 6, p. 487, Sep. 2022, doi: 10.1007/s42979-022-01382-9.
- [18] M. B. Oetiker, G. P. Baker, and L. Guzzella, “A Navigation-Field-Based Semi-Autonomous Nonholonomic Vehicle-Parking Assistant,” *IEEE Transactions on Vehicular Technology*, vol. 58, no. 3, pp. 1106–1118, Mar. 2009, doi: 10.1109/TVT.2008.928643.
- [19] S.-J. Huang and G.-Y. Lin, “Parallel auto-parking of a model vehicle using a self-organizing fuzzy controller,” *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 224, no. 8, pp. 997–1012, Aug. 2010, doi: 10.1243/09544070JAUTO1366.
- [20] K. Demirli and M. Khoshnejad, “Autonomous parallel parking of a car-like mobile robot by a neuro-fuzzy sensor-based controller,” *Fuzzy Sets and Systems*, vol. 160, no. 19, pp. 2876–2891, Oct. 2009, doi: 10.1016/j.fss.2009.01.019.
- [21] S. J. Qin and T. A. Badgwell, “A survey of industrial model predictive control technology,” *Control Engineering Practice*, vol. 11, no. 7, pp. 733–764, Jul. 2003, doi: 10.1016/S0967-0661(02)00186-7.
- [22] F. Borrelli, P. Falcone, T. Keviczky, J. Asgari, and D. Hrovat, “MPC-based approach to active steering for autonomous vehicle systems,” *International Journal of Vehicle Autonomous Systems*, vol. 3, no. 2/3/4, pp. 265–291, 2005, doi: 10.1504/IJVAS.2005.008237.
- [23] H. Wang, B. Liu, X. Ping, and Q. An, “Path Tracking Control for Autonomous Vehicles Based on an Improved MPC,” *IEEE Access*, vol. 7, pp. 161064–161073, 2019, doi: 10.1109/ACCESS.2019.2944894.
- [24] T. T. Thuong and V. T. Ha, “Adaptive Control For Mobile Robots Based On Intelligent Controller,” *Journal of Applied Science and Engineering*, vol. 27, no. 5, pp. 2481–2487, doi: 10.6180/jase.202405_27(05).0012.
- [25] T. T. Thuong, V. T. Ha, and L. N. Truc, “Intelligent Control for mobile robots based on Fuzzy Logic Controller,” *Intelligent Systems and Networks*, pp. 566–573, 2023. doi:10.1007/978-981-99-4725-6_67.
- [26] L. Fridman, J. Terwilliger, and B. Jenik, “DeepTraffic: Crowdsourced Hyperparameter Tuning of Deep Reinforcement Learning Systems for Multi-Agent Dense Traffic Navigation,” *arXiv preprint arXiv:1801.02805*, 2018.

BIOGRAPHIES OF AUTHORS



Vo Thanh Ha    is a lecturer in the Faculty of Electrical and Electrical Engineering, University of Transport and Communications. She received her B.S degree in Control, and Automation Engineering from the Thai Nguyen University of Technology, Vietnam, in 2002. She received her master’s degree from the Hanoi University of Science and Technology, Vietnam, in 2004. She received a Ph.D. degree from the Hanoi University of Science and Technology, Vietnam, in 2020, both in Control and Automation Engineering. Her research interests include the field of electrical drive systems, power electronics, and electric vehicles. She can be contacted at email: vothanhha.ktd@utc.edu.vn.



Chu Thi Thuy    is graduated with an Electrical Engineering degree from the University of Transport and Communications, Vietnam, in 2020. She works at the University of Economics-Industrial Technology, Hanoi, Vietnam. Her research interests include power electronic and electrical powertrain systems. She can be contacted at email: ctthuy@uneti.edu.vn.