Semiconductor Optoelectronics, *Vol. 42 No. 1 (2023), 1037-1047*
https://bdtgd.cn/

Semiconductor Optoelectronics
ISSN: 1001-5868

A DEEP LEARNING BASED PREDICTIVE CONTROL FOR COLLISION AVOIDANCE IN UNKNOWN ENVIRONMENTS

To Van Binh

Faculty of Electronic Engineering Technology, University of Economics – Technology for Industries, Ha Noi, Viet Nam, *tv_binh@uneti.edu.vn

Abstract. Model predictive control (MPC) is a popular control method that uses real-time optimization problem solving to compute mathematical control actions. However, due to the risk of collisions and uncertainty in the operating environment, it becomes complicated to apply MPC when it comes to safety related decisions. This paper introduces an integrated solution for autonomous vehicles (Avs) in unknown environments, combining model predictive control and deep learning (DL) techniques to avoid collisions and dangerous objects. This solution uses deep neural networks to predict the accuracy and degree of requirements and calculate the cost and priority between tasks. By using different simulation examples, test to see the feasibility and stability of this integrated approach.

Keywords: Model Predictive Control, Deep Learning, Neural Network, Autonomous Driving, Collision Avoidance.

1 Introduction

AVs are a new application of automation technology in the field of transportation. This is a vehicle that does not need a driver to control but is equipped with automatic control systems and sensors to recognize and respond to the surrounding environment. AVs can be used to transport passengers in cities, airports and other public areas. They can be controlled remotely or automatically on a fixed route with a pre-set speed and stroke. The use of AVs has many benefits such as reducing traffic accidents caused by human error, reducing waiting times for passengers, increasing the convenience and efficiency of public transport. However, to make the development successful, it requires support and investment from the authorities and manufacturers to ensure the safety and stability of this system in all conditions. weather events. AVs are equipped with sensors to detect conflicting environments and make travel and speed decisions based on this data. These sensors include LiDAR sensor, camera, radar and GPS. These systems create a 3D map of the surroundings and allow the vehicle to automatically adjust its course to avoid obstacles or hazards. To ensure the safety of passengers, AVs are also equipped with detection and observation systems and remote driver monitoring systems to ensure safe and efficient operation. In the event of an emergency, this system can interact with station staff or drivers remotely to handle the problem. In many countries around the world, governments and technology corporations are investing in the development of AVs in public transport such as cars, taxis or electric trains. They are becoming the cutting-edge trend in transportation technology and a measure of sustainable development in this field.

MPC is a technique in the field of automatic control, widely used in AVs to make decisions about the vehicle's actions in the future. MPC computes monitoring check actions for vehicles by optimizing a predefined cost function, with no limitations such as minimizing the error between actual locations of the vehicle and the target location. MPC calculates control actions to adjust speed, direction and strut between different driving modes, from automatic to manual mode. It calculates actions based on input and output constraints and model parameters, such as engine, traction, and other locomotion parameters. With features like model predictive controllers (MPCs), AVs can safely and efficiently maneuver when encountering difficult road situations, such as storing multiple vehicle information at the same time. This helps to increase safety and stability for users and reduce the risk of traffic accidents. However, to implement MPC for AVs, it is necessary to have complete and accurate operating models, real-time data recognition variables and some Powerful computing system to perform optimization solving.

MPC integrated with DL model using LiDAR sensor is an advanced technology used in AVs. LiDAR measurements allow scanning of surrounding objects to be performed, generating 3D data points to create a detailed map of the environment. Deep neural network (DNN) model is used to analyze and extract information from LiDAR-generated data, helping AVs predict and deal with complex road situations, such as development or reverse direction. MPC is integrated to control the autonomous vehicle (AV) action switching and make minimal decisions about future action based on the data of the DNN. Combining these two technologies provides AVs with an intelligent control system that is able to predict and react quickly to reduce the risk of accidents and ensure the safety of passengers.

Therefore, in order to successfully develop a MPC integrated with DNN modeling based on LiDAR measurements, a large and complete database is required training and tuning the DNN model, different LiDAR devices and sensors are required to get more ambient data, and powerful computation is required to implement the optimization solution.

2 Related work

With the development of society, AVs increasingly require a higher level of perfection, including the ability to keep lanes and automatically avoid obstacles. Therefore, more and more research is using different methods to control AVs. In document [1,2], filters are used to reduce noise in the input data and solve problems such as minimizing the deviations between the original position and the actual position of the AV. The method uses neural network models to predict the behavior of AVs and solve problems such as identity recognition and detection of potentially conflicting events in documents [3-6]. The decision and planning system helps AVs to make decisions and act in accordance with the surrounding environment, through the work of calculating the optimal path, choosing the appropriate speed and direction of movement in the vehicle [7-11]. The reinforcement learning method allows AVs to automatically learn to control and make decisions based on experiences accumulated through road runs in the literature [12,13]. In addition, model prediction method calculates future control optimization based on kinematics model to solve the stated problems such as collision mitigation and AV motion optimization practice [14-16]. Inheriting the above works, the paper develops MPC together with DNN model and LiDAR sensor, which helps to improve computational efficiency and security of the whole system.

3 Neural Network based Model Predictive Control Method

3.1 Approach Overview

Discrete-time controlled MPC involves optimizing a problem around a set of state values for a model, in order to achieve an objective defined target. The resulting sequence of control signals is used as input to the system factory for a short period of time (system time step), after which new state values are measured and the procedure is repeated again. MPC is applied in autonomous driving, which uses DL to avoid velocity-based collisions in unknown environments. Data from the LiDAR sensor is used to gather information about the position, size, and velocity of the obstacle. A set of DNN is used to estimate the exact ratio of the collision rate, which is used by the controller to slow down in areas with high collision risk. The cost of a collision is calculated based on the fact of the collision and in the direction of the potential hazard, determined using the contrast velocities between the AV and the obstacle, forming the collision cones for each major obstacle.

3.2 Vehicle Modeling

The geometrical structure of the AV is depicted in Figure 1.

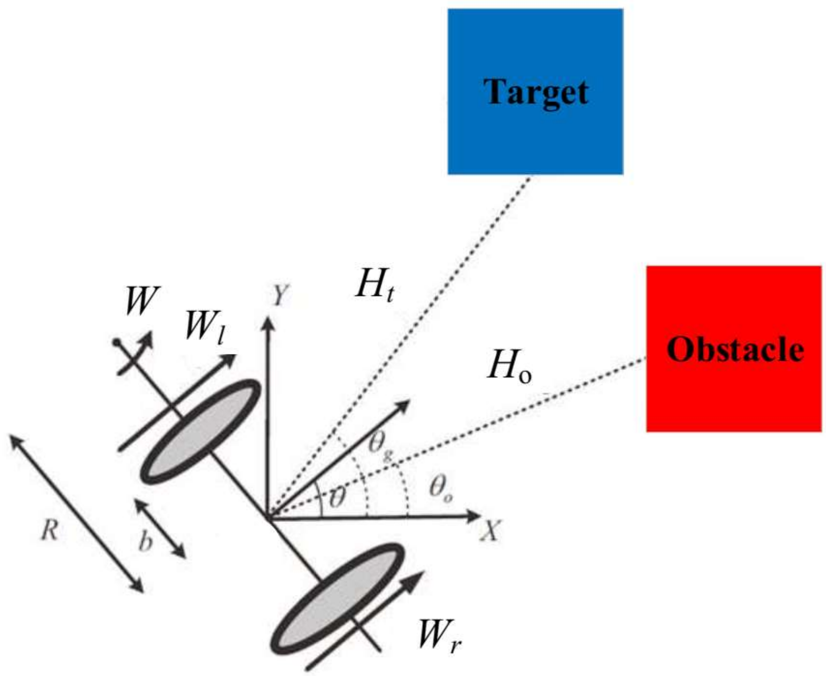

Fig. 1. Geometric structure of AV

The model of the AVs is shown as follows [6,17]:

$$P(k+1) = P(k) + \Delta T \times R \times Q(k) \tag{1}$$

Where $R = \begin{bmatrix} \cos\theta(k) & 0 \\ \sin\theta(k) & 0 \\ 0 & 1 \end{bmatrix}$;

$P(k)$, $P(k+1)$ are the positions of the autonomous vehicle at time k and $k+1$.

The position of AV described by (X, Y, θ), LiDAR sensor is responsible for measuring and collecting data.

$$P = [X(k), Y(k), \theta(k)]^T \tag{2}$$

and $Q(k) = [V(k), W(k)]^T$

$$\begin{cases} X(k+1) = X + V \times cos\,\theta \times T \\ Y(k+1) = Y + V \times sin\,\theta \times T \\ \theta(k+1) = \theta + W \times T \end{cases} \tag{3}$$

Where T is the sampling time. Suppose the desired trajectory of the Vehicle is:

$$P_r = [X_r(k), Y_r(k), \theta_r(k)]^T \tag{4}$$

We have

$$\begin{aligned} P_r(k+1) &= P_r + \Delta T \times \begin{bmatrix} \cos\theta_r & 0 \\ \sin\theta_r & 0 \\ 0 & 1 \end{bmatrix} \times Q \\ &= P_r + \Delta T \times \begin{bmatrix} \cos\theta_r & 0 \\ \sin\theta_r & 0 \\ 0 & 1 \end{bmatrix} \times \begin{bmatrix} V_r \\ W_r \end{bmatrix} \end{aligned} \tag{5}$$

in there $Q_r(k) = [V_r(k), W_r(k)]^T$.

$$\begin{cases} X_r(k+1) = X_r + V_r \times cos\,\theta_r \times T \\ Y_r(k+1) = Y_r + V_r \times sin\,\theta_r \times T \\ \theta_r(k+1) = \theta_r + W_r \times T \end{cases} \tag{6}$$

Tracking error

$$\begin{cases} X_{TE}(k) = [X(k) - X_r(k)] \times cos\,\theta_{TE}(k) + [Y(k) - Y_r(k)] \times sin\,\theta_{TE}(k) \\ Y_{TE}(k) = [X(k) - X_r(k)] \times sin\,\theta_{TE}(k) + [Y(k) - Y_r(k)] \times cos\,\theta_{TE}(k) \\ \theta_{TE}(k) = \theta(k) - \theta_r(k) \end{cases} \tag{7}$$

From there we have:

$$\begin{cases} X_{TE}(k+1) = X_{TE} \times [V_1 - V_r + Y_{TE}W_r] \times T \\ Y_{TE}(k+1) = Y_{TE} \times [V_2 - X_{TE}W_r] \times T \\ \theta_{TE}(k+1) = \theta_{TE} + [W - W_r] \times T \end{cases} \tag{8}$$

Where:

$$V_1 = V_r(k)\,cos\,\theta_{TE}(k) \tag{9}$$

$$V_2 = V_l(k)\,cos\,\theta_{TE}(k) \tag{10}$$

AV moves by two wheels driving the right and left with the linear velocity of the two wheels respectively V_r, V_l.

$$V = \frac{V_r + V_l}{2} \tag{11}$$

$$W = \frac{V_r - V_l}{2b} \tag{12}$$

$$R = \frac{V_r + V_l}{b(V_r - V_l)} \tag{13}$$

Where R is the turning radius and the surrounding is not clear with obstacles and targets. To design the controller, useful information is included including the distance and angle between the vehicle and the target, as well as the distance and angle between the vehicle and the nearest obstacle. Then, the observed model shows the following:

$$O(k) = \left[H_{target}, \theta_g, H_{obstacle}, \theta_o\right]^T \tag{14}$$

3.3 Deep learning based model for obstacle avoidance

We explicitly represent the collision $p_{collision}$ as information about the probability of a collision. Determining this is computed by a set of deep neural networks like the example in

[18], and the information is computed immediately when the MPC decides to plan. Each is a factor, as the medium is difficult to access during training and modeling over a long period of time.

In this paper, the architecture of the network neural network can be represented as a fully connected multi-layer neural network, depicted in Figure 2. The input of the network consists of 4 nodes $s = (s_1, s_2, s_3, s_4) = \left(D_{target}, D_{obstacle}, \theta_g, \theta_o\right)$ to pass the input information down to the next layers. The input layer is created by combining the measurement variables from the LiDAR sensor and the vehicle's current state parameters. All input values have been normalized to be in the range $[-1, 1]$.

The neural network is used to predict the collision point and the $p_{collision}$ performance. The first two hidden layers consist of 200 nodes, while the remaining hidden layers consist of 100 nodes. The nodes in the neural network are activated by the RELU function, which is a common activation variable used in fully connected neural networks [18]. The output nodes have a value between 0 and 1, which corresponds to the screen and determines the $p_{collision}$ collision.

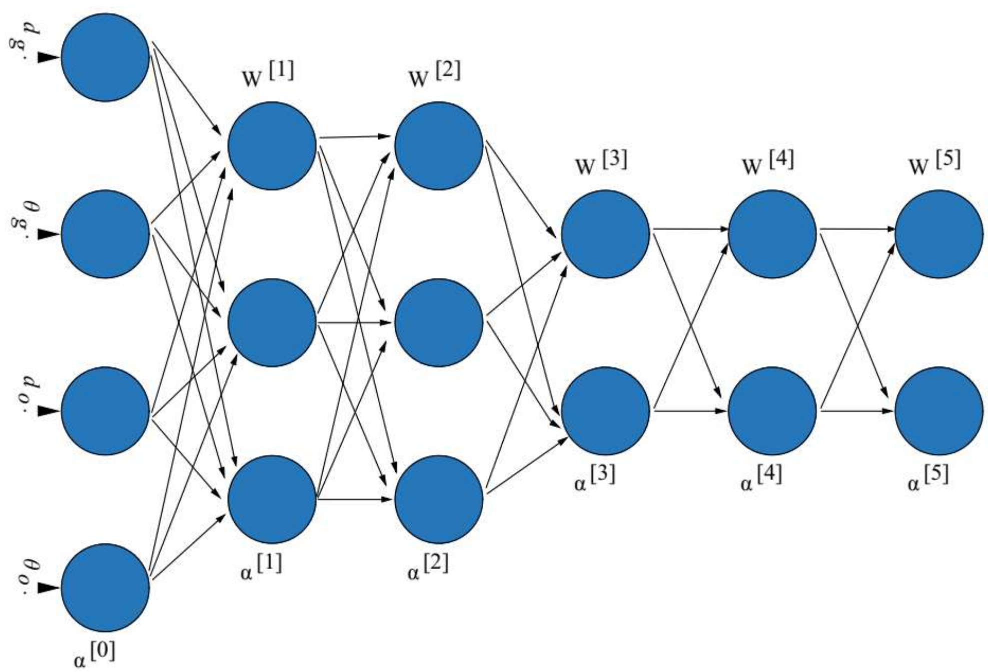

Fig. 2. Neural network structure with six layers.

3.4 Deep learning Model Predictive Control

Every time the MPC plans a new trajectory, the vehicle reads measurements from the LiDAR sensor. Points belonging to the obstacle object will be detected by the LiDAR sensor and indicated by blue dots in Figure 3. For each detected point, a circle $Circ_k$ will be detected created and centered at the corresponding point. These circles will be represented by the circles marked in Figure 3. If any of the points are above, shown by Figure 3, create a circle $Circ_k$, title label will specify the best sample data format. If there is a collision within the optimization range, it will be 1, otherwise if there is no collision, the target label will be 0. The vehicle moved to, the measurement to the right of the LiDAR sensor has been read and the item label will be reassigned to the description above. Note that the model will be trained, retrying from scratch at every new vehicle position, allowing the LiDAR sensor to be updated and optimized for step-by-step obstacle avoidance.

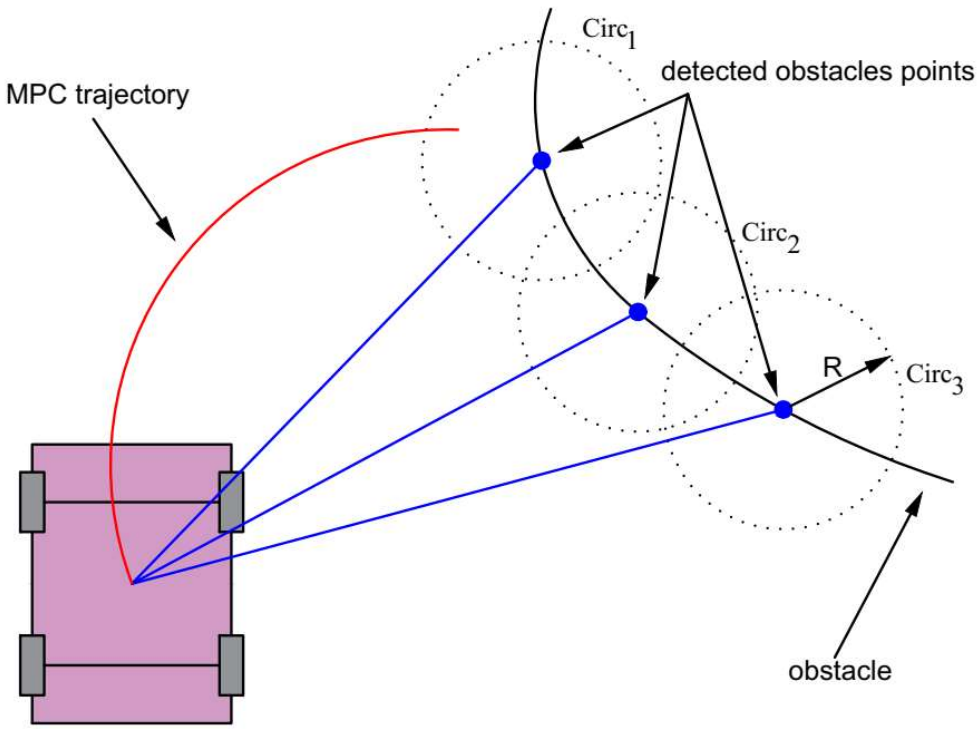

Fig. 3. AV uses LiDAR sensor to scan obstacles

4 Experimental Results

Our aim was to demonstrate the results obtained from the offline learning phase in a simulated simulation environment using Gazebo [19] and OpenAI Gym [20]. We assigned a navigation task, where the autopilot needed to learn how to avoid obstacles in a large maze of randomly placed obstacles, wide paths, and 90-degree left/right angles. The simulation environment is used in Gazebo and rendered using the Erle-Rover platform, a Linux-based smart car powered by APM autopilot and with Robot Operating System (ROS) support [19], as could be show in Figure 4.

Obstacles, paths and turns can be randomly generated in the simulated environment. All in all, we deploy 250 learning stages to train the neural network. Of these, the first 100 stages were used to train with a set of obstacles with a maximum total recovery length of 186 m. The remaining 150 stages are used for a different set of obstacles, paths and turns with a total track length of 146 m.

Figures 5 and 6 depict simulation results for different scenarios with dense and cluttered environments. For dense environments, this is one of the problems self-driving cars have trouble dealing with. The starting and destination locations are shown with yellow rectangles and blue dots. The target starts without collision. Figures 5 and 6 show that the Erle-Rover self-driving car successfully completed the tasks.

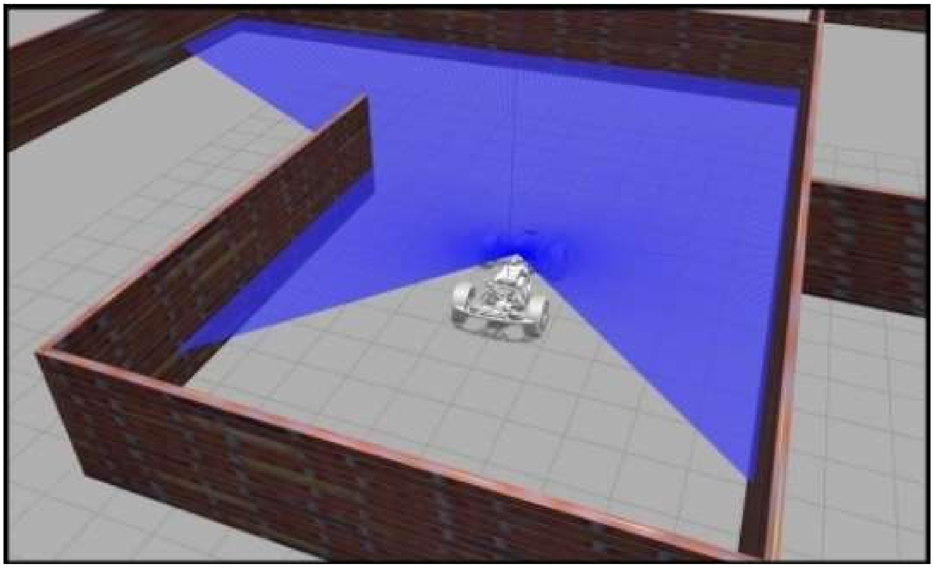

Fig. 4. Simulation in Gazebo with the blue area being the space monitored by the LiDAR sensor

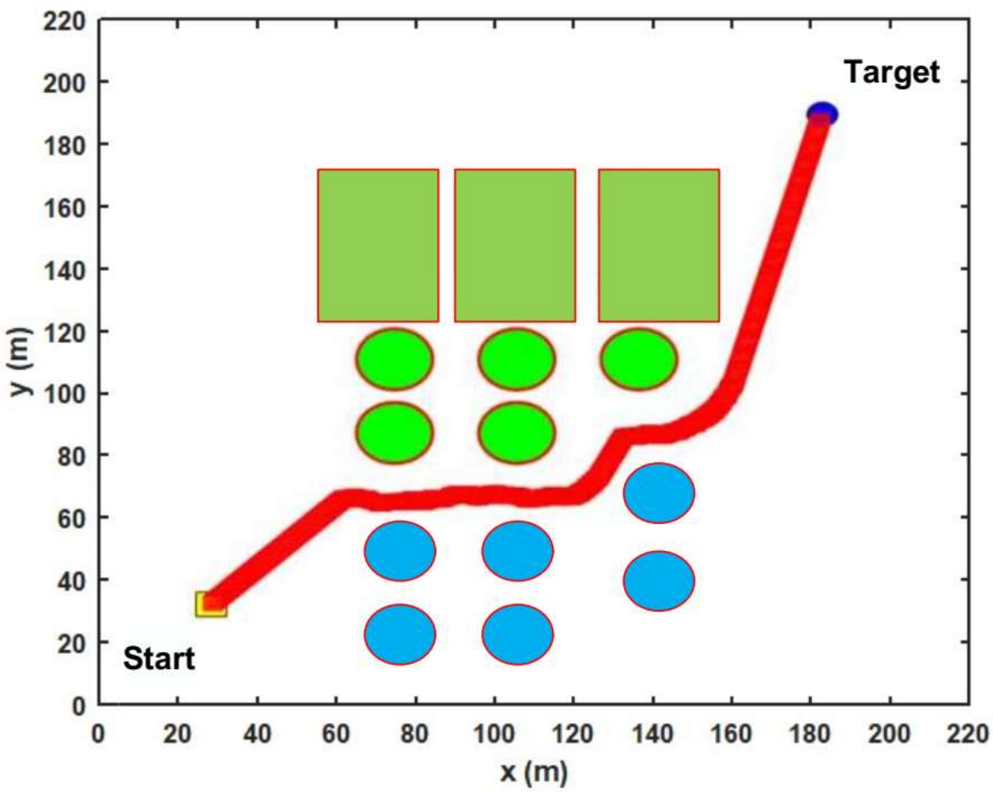

Figure 5. Obstacle avoidance in cluttered environments

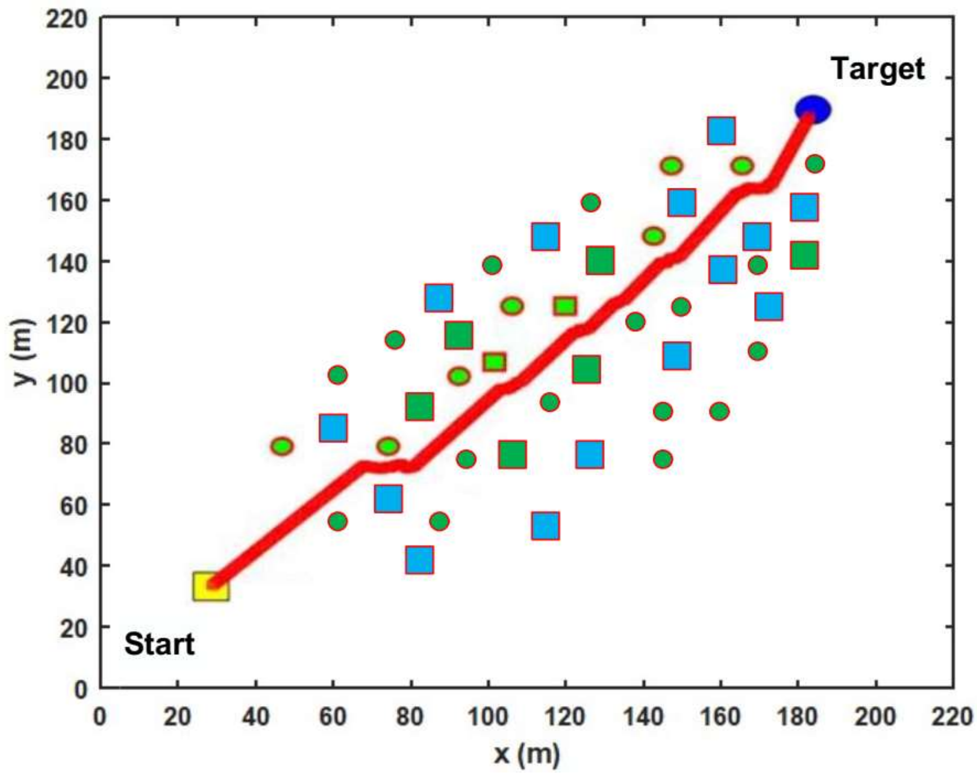

Figure 6. Obstacle avoidance in dense environments

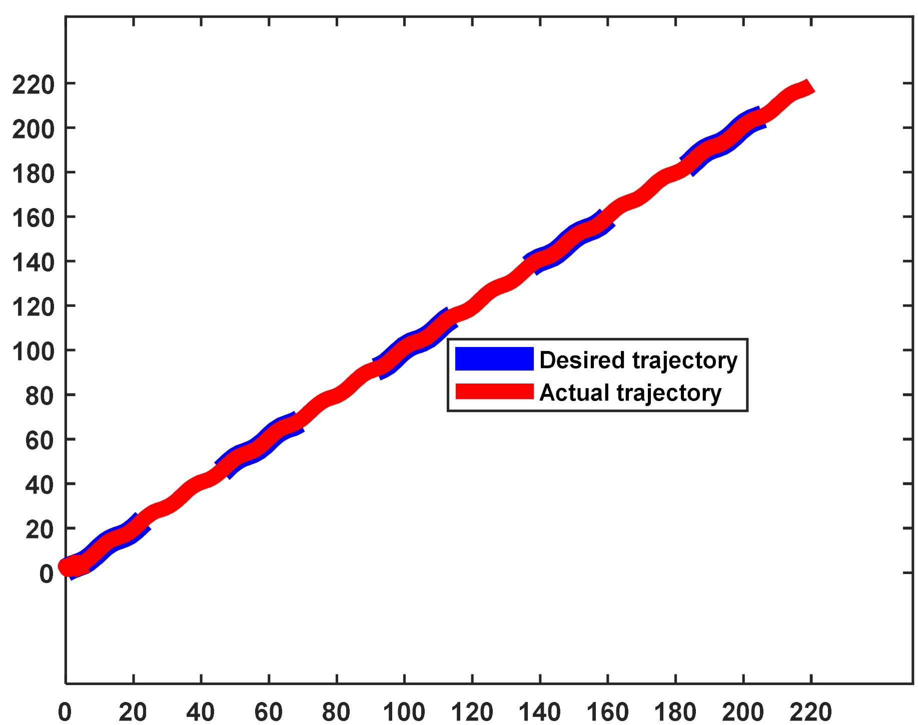

Figure 7. Stick to the desired trajectory

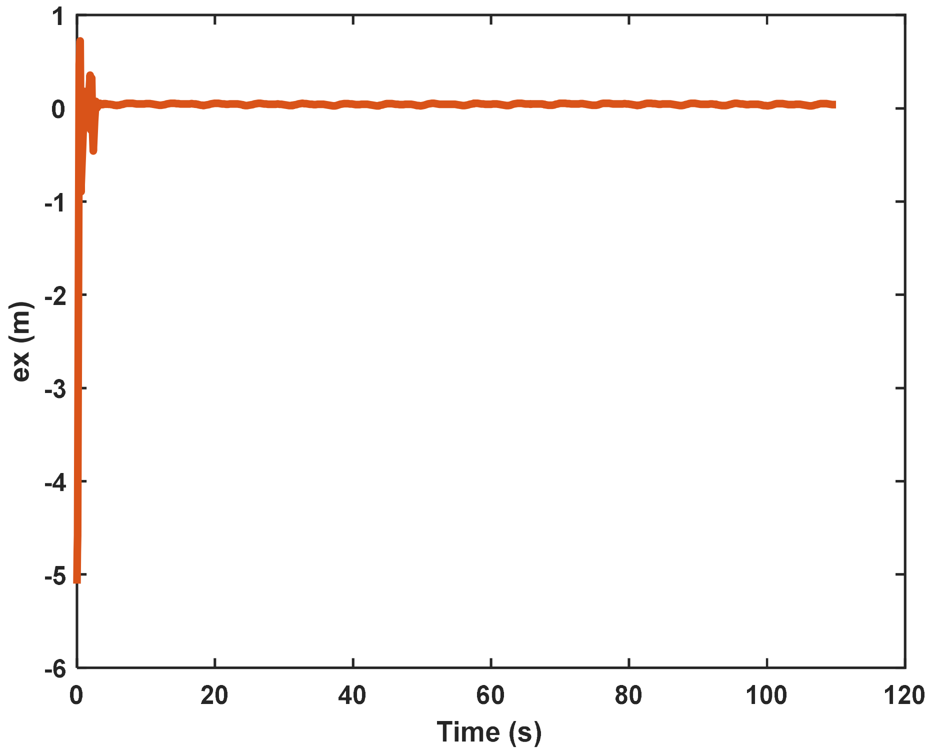

Figure 8. Error in x-axis

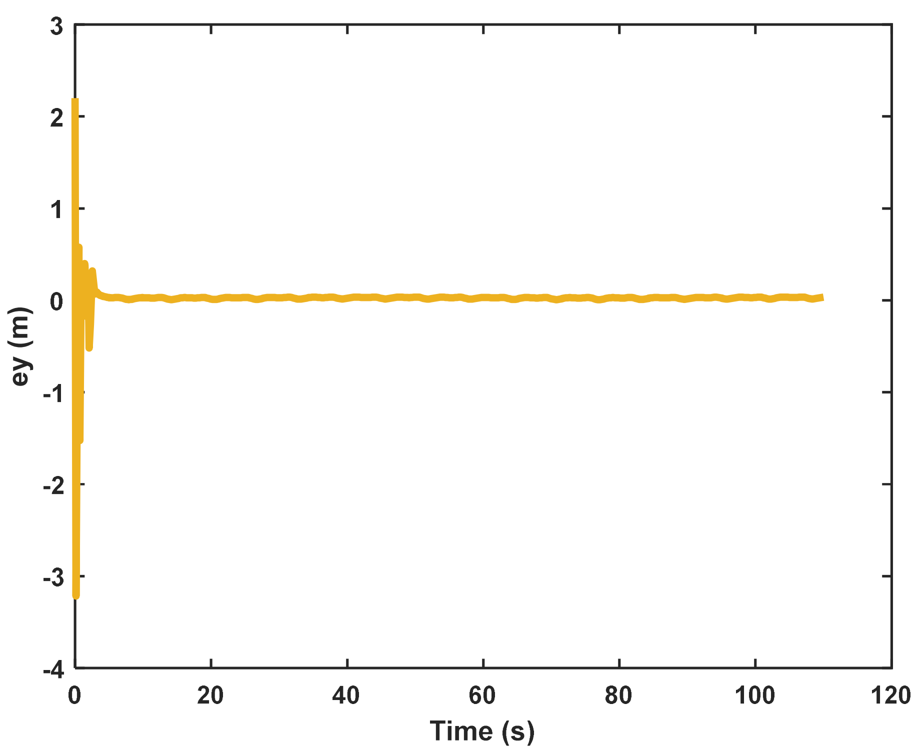

Figure 9. Error in y-axis

From the results of Figure 7, we see that the self-propelled vehicle has followed the desired trajectory. The error of tracking along the x and y axes is approximately zero Figure 8, 9.

5 Conclusion

In this paper, we present a solution that combines autonomous driving in an unknown environment, combining control prediction and deep learning techniques for obstacle avoidance. The obstacle avoidance goal is determined by the MPC objective function using a neural network. This method reduces speed in areas with obstacles, such as in bends or cones, where a high-impact novel speed determination determines the model's predicted change to be low. Many of these simulate each other and for different scenarios. Therefore, the stability and applicability of the automated driving method can be demonstrated.

References

1. Nguyen, B., M., Fujimoto, H., Hori Y., 2014, "*Yaw angle control for autonomous vehicle using Kalman filter based disturbance observer*", In SAEJ. EVTeC and APE Japan.
2. Min, Wu, Cheng, & Zhao. (2019), "*Kinematic and Dynamic Vehicle Model-Assisted Global Positioning Method for Autonomous Vehicles with Low-Cost GPS/Camera/In-Vehicle Sensors*", Sensors, 19(24), 5430.
3. Kocić, J., Jovičić, N., & Drndarević, V. (2019), "*An End-to-End Deep Neural Network for Autonomous Driving Designed for Embedded Automotive Platforms*", Sensors, 19(9), 2064. doi:10.3390/s19092064.
4. Swain, S. K., Rath, J. J., & Veluvolu, K. C. (2021), "*Neural Network Based Robust Lateral Control for an Autonomous Vehicle*", Electronics, 10(4), 510. doi:10.3390/electronics10040510.
5. Mahmoud, Y., Okuyama, Y., Fukuchi, T., Kosuke, T., & Ando, I. (2020), "*Optimizing Deep-Neural-Network-Driven Autonomous Race Car Using Image Scaling*", SHS Web of Conferences, 77, 04002.
6. Han, Y., Zhu, Q., & Xiao, Y. (2018), "*Data-driven Control of Autonomous Vehicle using Recurrent Fuzzy Neural Network Combined with PID Method*", 2018 37th Chinese Control Conference (CCC). doi:10.23919/chicc.2018.8482696
7. Feng, S., Qian, Y., & Wang, Y. (2021), "*Collision avoidance method of autonomous vehicle based on improved artificial potential field algorithm*", Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering, 095440702110143. doi:10.1177/09544070211014319.
8. Wu, W., Jia, H., Luo, Q., & Wang, Z. (2019), "*Dynamic Path Planning for Autonomous Driving on Branch Streets With Crossing Pedestrian Avoidance Guidance*", IEEE Access, 7, 144720–144731. doi:10.1109/access.2019.2938232.
9. Lu, B., Li, G., Yu, H., Wang, H., Guo, J., Cao, D., & He, H. (2020), "*Adaptive Potential Field-Based Path Planning for Complex Autonomous Driving Scenarios*", IEEE Access, 8, 225294–225305.
10. Lin, P., Choi, W. Y., & Chung, C. C. (2020), "*Local Path Planning Using Artificial Potential Field for Waypoint Tracking with Collision Avoidance*", 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC). doi:10.1109/itsc45102.2020.9294717.
11. Prochowski, L., Ziubiński, M., Szwajkowski, P., Gidlewski, M., Pusty, T., & Stańczyk, T. L. (2021), "*Impact of Control System Model Parameters on the Obstacle Avoidance by an Autonomous Car-Trailer Unit: Research Results*", Energies, 14(10), 2958.

12. Levine, S., Finn, C., Darrell, T., and Abbeel, P., "*End-to-end training of visuomotor policies*", The Journal of Machine Learning Research, 17(1), 1334–1373, 2016.
13. Kahn, G., Villaflor, A., Pong, V., Abbeel, P., and Levine, S., "*Uncertainty-aware reinforcement learning for collision avoidance*", *arXiv preprint arXiv:1702.01182, 2017*.
14. Yao, & Tian. (2019), "*A Model Predictive Controller with Longitudinal Speed Compensation for Autonomous Vehicle Path Tracking*", Applied Sciences, 9(22), 4739. doi:10.3390/app9224739.
15. Trieu Minh, V., "*Predictive Control for Controlling and Driving Autonomous Vehicles*", International DAAAM Baltic Conference Industrial Engineering, 2014.
16. Liu, K., Gong, J., Chen, S., Zhang, Y., & Chen, H. (2018), "Model Predictive Stabilization Control of High-Speed Autonomous Ground Vehicles Considering the Effect of Road Topography", Applied Sciences, 8(5), 822.
17. Filipescu, A., Minzu, V., Dumitrascu, B., Filipescu, A., & Minca, E. (2011), "*Trajectory-tracking and discrete-time sliding-mode control of wheeled mobile robots*", 2011 IEEE International Conference on Information and Automation. doi:10.1109/icinfa.2011.5948958.
18. Goodfellow, I., Bengio, Y., and Courville, A., "Deep Learning", *MIT Press,* 2016.
19. Zamora, I., Gonzalez Lopez, N., Mayoral Vilches, V. and Hernandez Cordero, A., "*Extending the OpenAI Gym for robotics: A toolkit for reinforcement learning using ROS and Gazebo*", arXiv 2016, arXiv:1608.05742.
20. Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J.,Tang, J. and Zaremba, W. "*Openai gym*", arXiv preprint in arXiv:1606.01540, 2016.