International Journal of Engineering Trends and Technology Volume 69 Issue 4, 204-208, April 2021
ISSN: 2231 – 5381 /doi:10.14445/22315381/IJETT-V69I4P228

Deep Reinforcement Learning for Computerized Steering Angle Control of Pollution-Free Autonomous Vehicle

M.Karthikeyan[1], S.Sathiamoorthy[2]

[1]*Division of Computer and Information Science, Annamalai University, Chidambaram*

[2] *Tamil Virtual Academy, Chennai*

[1]mkarthi82@gmail.com, [2]Ks_sathia@yahoo.com

ABSTRACT

Manpower cost is the major expense in Industrial and domestic applications, and hence the whole world is moving towards automation with the help of Artificial Intelligence (AI). AI techniques have a major role in making the process automated and advanced in modern industrial requirements. Smart devices, Smart vehicles, Smart home, Smart Factory, Smart home appliances, etc., are working with automated process based on the principle of artificial intelligence, and hence in this paper, one of the advanced AI techniques is chosen for the automated vehicle (AV) where steering angle is controlled in order to keep the vehicle within the lane. In this paper, an adaptive deep reinforcement learning algorithm for autonomous vehicles is presented, and the results have been analyzed. In this paper deep Q learning algorithm is used to control the steering angle of an autonomous vehicle. A transition model estimator is also developed to emulate the learning process using neural networks. This model helped this research work to utilize the available test data efficiently. This paper mainly focused on the objectives (i) Optimal learning policy as an adaptive learning system, (ii) Markov decision process (MDP) as a learning process in the learning system, and (iii) Numerical simulation of Deep Reinforcement algorithm with autonomous vehicle model. Continuation of this work would be the final stage of autonomous vehicle development.

Keywords: *Autonomous vehicle, Deep Reinforcement Algorithm, Steering Angle Control, Transition Model Estimator, and Adaptive learning.*

I. INTRODUCTION

The rapid development of autonomous vehicles is increasing day by day, and the potential growth of automation in the automotive industry is exponentially raising and influencing the market share over the past two decades. Automated driving is achieved by optimizing the control systems, optimizing the driving route, Lane control and planning, driving policies for critical navigation, and road planning [1]. Hence, Autonomous vehicles created a greater impact across the globe, and advanced research tasks are going on at various levels (Level-1 to Level-5). It has emerged in the automotive domain with automated steering control, automated tyre pressure control by monitoring road friction coefficient, automated power window control, automated braking system, automated body control, and automated safety system [2]. Prior to fully automated vehicles, semi-automated vehicles were very popular, and currently, they are in the market with advanced driver assist system (ADAS), which reduce the driver's workload and minimize the risks due to driver's negligence. Similarly, Artificial Intelligence (AI) has also reached several milestones in recent years, and now AI is an inevitable idea in automated cars. AI is helping automotive researchers to achieve the desired target levels such as Obstacle avoidance (OA), Forward collision warning (FCW), Lane Keep Assist System (LKA), Tyre pressure control (TPC), and automatic lighting control (AL). Among the AI techniques, very familiar techniques for AVs are Reinforcement algorithms and Deep Learning algorithm, which are mainly applied in AVs for self-learning and decision making to achieve the desired goals, for example, identification of obstacles accurately and decision making on the steering angle values to control the steering wheel in order to avoid the forward collisions and the application of braking required. At a nutshell, it has been evident that various AI techniques can provide desirable solutions for AVs in diagnosing the environment and driving the vehicle with appropriate decision making [3]. In this paper, one of AI techniques, deep reinforcement learning, is used to control the steering angle of AV, and the same is presented.

II. CONTROL OF AUTONOMOUS VEHICLE

A Closed-loop Control system integrated with the help of deep reinforcement learning is applied in the proposed autonomous car. LiDAR sensor is used to capture the images of obstacles, stored as data set and trained with the help of a reinforcement algorithm. Figure 1 shows the schematic of an autonomous vehicle with the key control systems components.

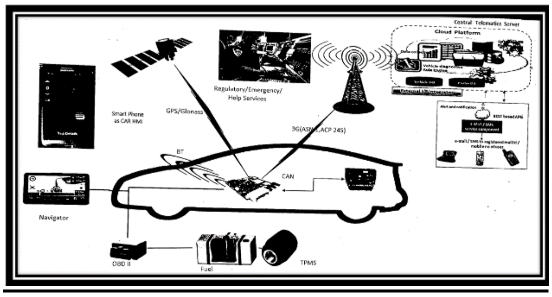

Figure 1: Structure of Autonomous Vehicle with the key components

While LiDAR collects the obstacle images, navigation information is also captured so that vehicle will reach the destination precisely. The smartphone has the vehicle HMI dashboard from which autonomous cars can be accessed/controlled remotely. Cloud telematic server is sued to store the vehicle diagnostics information such as (i) Detecting faults at the running conditions and fix the same remotely (ii) Vehicle data like fuel level, oil level, battery storage level, cabin temperature, and driving speed, this data related to on-board diagnostics and off-board diagnostics data is mainly used for servicing the vehicle.

Both on-board and off-board diagnostics data are stored in the cloud server [5][6]. During emergency conditions, warning signals will be transferred to the central service station that is monitoring AVs and also to regulatory bodies and based on the severity of the warning, and respective actions will be taken.

III. DYNAMIC STEERING ANGLE CONTROL USING DEEP REINFORCEMENT LEARNING (DRL) ALGORITHM

Figure 2 shows the flow diagram of the deep reinforcement learning algorithm for autonomous vehicle position control within the lane by controlling the steering angle. The training objective is to keep the vehicle traveling along the centerline of the lanes by adjusting the front steering angle. DRL is applied for controller performance improvement, Lane tracking and root maintenance, dynamic speed control to perform the tasks such as Lane keep assist, development of driving policies as per navigation, policies learning & training as per road conditions, cross-sections, associations with joints, and gaps, learning with deep reinforcement learning from the trained set of data of the captured images of traffic conditions such as animals, human interventions, trucks, bicycles, and safety with risk mitigation ensured by proper policies learning.

Figure 2 shows the deep reinforcement learning algorithm where the deep deterministic policy gradient (DDPG) algorithm is used, and it is a model-free, online, off-policy reinforcement learning method that is highly suitable for autonomous vehicles. DDPG agent is an action value-based reinforcement learning agent which calculates the policy that extends the long-term reward. This agent is highly suitable for this application [7-18]. In figure 2, LiDAR inputs are observations that are given to Agent. The policy is updated by the DDPG agent, and steering angle values are derived.

Based on the calculated steering angle values, steering wheel movement will be activated in the vehicle, and vehicle movement will be decided accordingly. From this, new LiDAR inputs are given to the Agent, and desired steering angle is achieved precisely. In addition to the DDPG agent, the DQN agent is also tried in order to implement the lane keep assist system. Actually, the DDPG agent is suitable for steering angle control, and the DQN agent is for the lane keep assist system. Both of them are modeled, simulated, and the results have been analyzed in this paper.

Secondly, for this learning, Markov Decision Process (MDP) is used as a framework, and DDPG and DQN are activated along with this process. The discrete-time stochastic control process is adopted in MDP. It provides a numerical framework where partial decision-making is made.

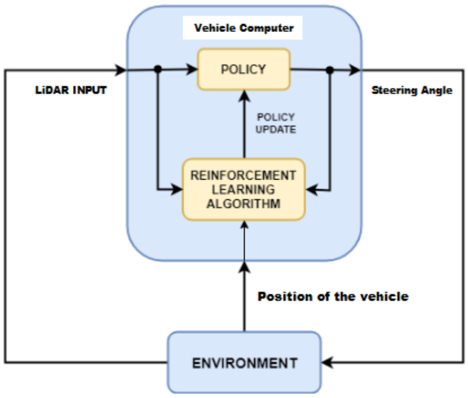

Figure 2: Deep Reinforcement Learning Algorithm

In this paper, based on the traffic condition, steering angle control is made, and hence the partial decision is based on the moving object across the road, and MDP is used to determine the framework to accommodate the agent and policy.

IV. MODELLING AND SIMULATION OF DRL MODEL

Deep reinforcement learning with the autonomous vehicle is modeled using Matlab /Simulink tool from which M-scripting functions and block sets are used for training policies using DDPG agent in DRL algorithm. These policies are used to implement a steering angle control system. The policies are implemented in this paper with polynomials, deep neural networks, and look-up tables. A deep learning toolbox is used for this work to train policies by enabling them to interact with an environment where the vehicle is moving around is represented by Matlab/Simulink models (Dynamic Vehicle models) for this research work. Figure 3 shows the Simulink model developed for the proposed deep Reinforcement Learning Algorithm. This model has three main systems as Signal Processing, RL agent, and vehicle & environment.

To evaluate DDPG and DQN algorithms, test with parameter settings, and monitor training progress, this work is carried out on the Simulink platform. To improve training performance, simulations in parallel GPUs with Parallel Computing Toolbox have been carried out. Existing policies are derived from deep learning frameworks with the help of Deep Learning Toolbox. In this paper, the MDP framework, along with the policies and DDPG & DQN agents, is applied. Optimized C and CUDA codes are generated to deploy trained policies on a microcontroller and GPUs. Following are the setting to be made for the simulation.

- Setting to be made for training
- To learn the DRL configuration
- Representation of policy
- Representation of Reward signals
- Signals to be observed and the corresponding actions to be taken
- Dynamics of Road conditions related to the environment

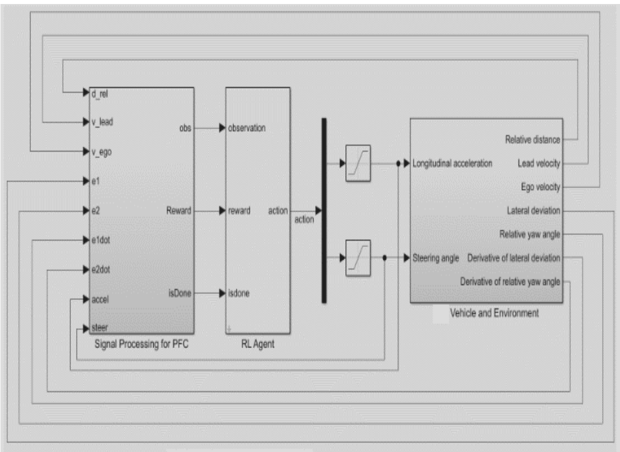

Figure 3: Matlab/Simulink Model of Deep Reinforcement Algorithm

RL Agent subsystem creation includes the following steps in this paper:

1. **Agent Creation** — Creation of agent with defined policy and DRL algorithm.
2. **Agent Training** — Training of agents using the defined traffic & climate conditions (Environment).
3. **Validation of Agent** — Verification, and Validation of agents (DDPG & DQN)
4. **Deploy Policy** — Deploy the trained policy representation.

V. RESULTS & ANALYSIS

DDPG and DQN agents are trained, and the results have been captured. As mentioned in the previous section, the Simulink model has been compiled and the results captured on Reinforcement Learning Episode Manager. Figure 4 shows the deep reinforcement algorithm learning process with the DDPG agent. To train the agent, the train function is used in Matlab/Simulink. This is a computationally-intensive process and took 7 minutes to complete. This plot in figure 4 shows the episode rewards with respect to episode number for episode reward, average reward, and episode Q_0.

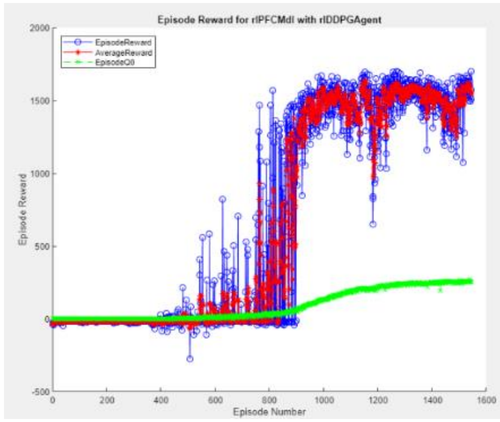

Figure 4: Deep Reinforcement Algorithm Learning Process with DDPG Agent

Figure 5 shows the deep reinforcement algorithm learning process with the DQN agent. The plot in figure 5 shows the episode rewards with respect to episode number for episode reward, average reward, and episode Q_0. Table 1 shows the RL training progress report, which has the episode numbers and rewards.

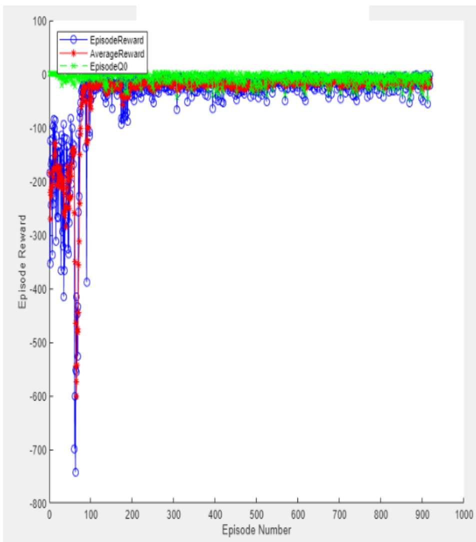

Figure 5: Deep Reinforcement Algorithm Learning Process with DQN Agent

Table:1 Deep Reinforcement Learning Training Progress

Training Progress Report	DDPG	DQN
Episode Number	1545	918
Episode Reward	1701.4486	-0.50702
Episode Steps	600	150
Episode Q0	255.0619	1.1219
Total Number of Steps	447581	129444
Average Reward	1562.3685	-8.8044
Average Steps	600	150
Window Length for averaging	5	5
Hardware Resources	CPU	CPU
Learn Rates	0.00001	0.001
Maximum number of Episodes	10000	5000
Maximum steps per episode	600	185
Training Stopped by	Episode Reward	Episode Reward
Training stopped at value	1700	-1
Elapsed Time	32741 sec	15342

Figure 6 shows the acceleration and steering angle of the autonomous vehicle using the DDPG agent. In the first 35 seconds, the relative distance is greater than the safe distance; thus, car tracks set velocity. To speed up and reach the set velocity, acceleration is mostly non-negative. From 35 to 42 seconds, the relative distance is mostly less than safe distance; thus car tracks the minimum lead velocity and sets velocity. Since lead velocity is less than set velocity to track lead velocity, acceleration becomes non-zero. From 42 to 58 seconds, autonomous car tracks set velocity and acceleration remains zero. From 58 to 60 seconds, relative distance becomes less than safe distance; thus car slows down and tracks lead velocity. Figure 7 shows the lateral deviation. As shown in the plot, the lateral deviation is greatly decreased within one second. The lateral deviation remains less than 0.05 m. Figure 8 shows the yaw error variation with the DDPG agent.

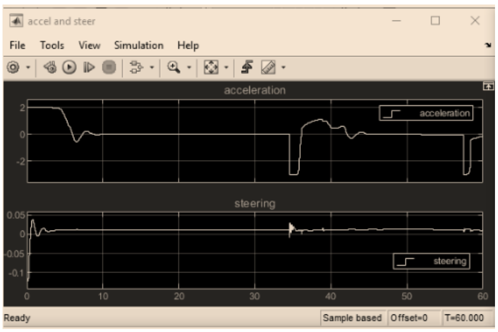

Figure 6. Acceleration and Steering angle variation (DDPG)

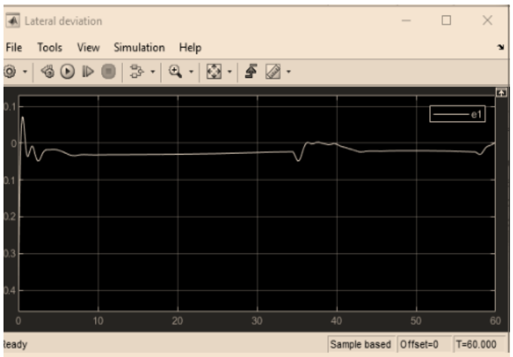

Figure 7: Lateral deviation (DDPG)

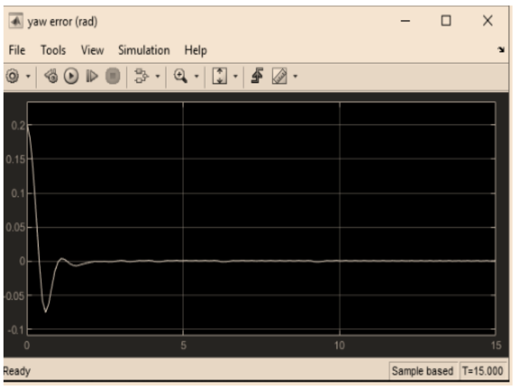

Figure 8: Yaw error variation (DDPG)

Figure 9 shows the lateral deviation of an autonomous vehicle with a DQN agent. The lateral deviation is less than 0.3 m. Similarly, figure 10 shows the steering angle values with the DQN agent.

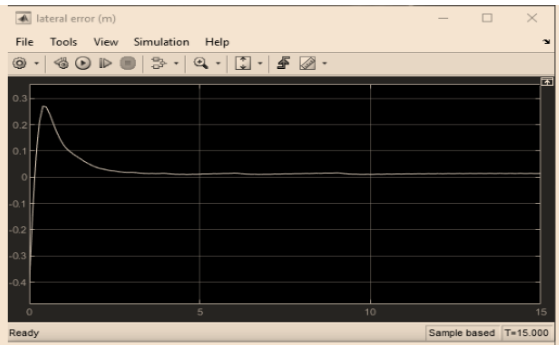

Figure 9: Lateral deviation (DQN)

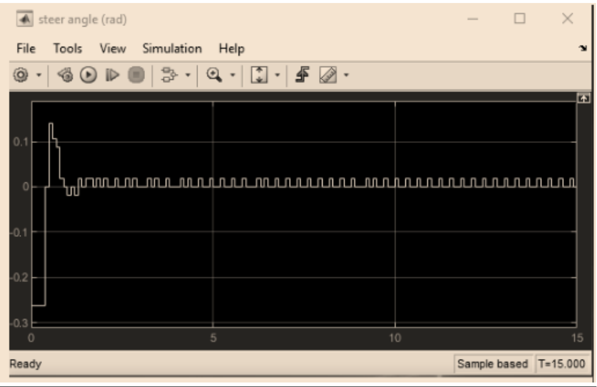

Figure 10: Steering angle variation (DQN)

VI. CONCLUSION

Autonomous vehicles with Artificial Intelligence are the most wanted transportation globally in order to achieve an accident-free and pollution-free environment. For more than a decade, researchers are performing extensive development in driverless cars in order to achieve the highest level with the help of AI techniques. The authors used the Deep Reinforcement algorithm technique with various agents and policies to achieve the goal. In this paper, the authors have developed a novel model of reinforcement deep learning with DDPG & DQN agent in Markov Decision Process frae work for the steering control of the autonomous vehicle. Contributions presented in this paper are summarized as (i) Development autonomous vehicle model with DRL algorithm using DDPG (ii) Development autonomous vehicle model with DRL algorithm using DQN (iii) Modelling and Simulation of the Proposed algorithm (iv) Steering angle values captured and the comparison. Further development of this research work will be carried out towards the connected autonomous cars.

REFERENCE

[1] Krogh, Bruce, and Charles Thorpe., Integrated path planning and dynamic steering control for autonomous vehicles., In Proceedings. 1986 IEEE International Conference on Robotics and Automation, 3(1986) 1664-1669. IEEE.

[2] Ma, Yifang, Zhenyu Wang, Hong Yang, and Lin Yang., Artificial intelligence applications in the development of autonomous vehicles: a survey., IEEE/CAA Journal of Automatica Sinica 7(2)(2020) 315-329.

[3] Xia, Wei, Huiyun Li, and Baopu Li., A control strategy of autonomous vehicles based on deep reinforcement learning., In 2016 9th International Symposium on Computational Intelligence and Design (ISCID), 2(2016) 198-201. IEEE.

[4] Li, Xiao, Hanchen Xu, Jinming Zhang, and Hua-hua Chang., Deep Reinforcement Learning for Adaptive Learning Systems., arXiv preprint arXiv:2004.08410 (2020).

[5] Sallab, Ahmad EL, Mohammed Abdou, Etienne Perot, and Senthil Yogamani., Deep reinforcement learning framework for autonomous driving., Electronic Imaging 19(2017) 70-76.

[6] Wang, Sen, Daoyuan Jia, and Xinshuo Weng., Deep reinforcement learning for autonomous driving., arXiv preprint arXiv:1811.11329 (2018).

[7] Kiran, B. Ravi, Ibrahim Sobh, Victor Talpaert, Patrick Mannion, Ahmad A. Al Sallab, Senthil Yogamani, and Patrick Pérez., Deep reinforcement learning for autonomous driving: A survey., IEEE Transactions on Intelligent Transportation Systems (2021).

[8] François-Lavet, Vincent, Peter Henderson, Riashat Islam, Marc G. Bellemare, and Joelle Pineau., An introduction to deep reinforcement learning., arXiv preprint arXiv:1811.12560 (2018).

[9] Duan, Yan, Xi Chen, Rein Houthooft, John Schulman, and Pieter Abbeel., Benchmarking deep reinforcement learning for continuous control., In International conference on machine learning, (2016) 1329-1338. PMLR.

[10] Van Hasselt, Hado, Arthur Guez, and David Silver., Deep reinforcement learning with double q-learning., In Proceedings of the AAAI Conference on Artificial Intelligence, 30(1)(2016).

[11] Gupta, Jayesh K., Maxim Egorov, and Mykel Kochenderfer., Cooperative multi-agent control using deep reinforcement learning., In International Conference on Autonomous Agents and Multiagent Systems,(2017) 66-83. Springer, Cham.

[12] Tampuu, Ardi, Tambet Matiisen, Dorian Kodelja, Ilya Kuzovkin, Kristjan Korjus, Juhan Aru, Jaan Aru, and Raul Vicente., Multiagent cooperation and competition with deep reinforcement learning., PloS one 12(4)(2017) e0172395.

[13] Li, Hongjia, Tianshu Wei, Ao Ren, Qi Zhu, and Yanzhi Wang., Deep reinforcement learning: Framework, applications, and embedded implementations., In 2017 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), (2017) 847-854. IEEE.

[14] Wei, Shiyin, Yuequan Bao, and Hui Li., Optimal policy for structure maintenance: A deep reinforcement learning framework., Structural Safety 83(2020) 101906.

[15] Nguyen, Thanh Thi, Ngoc Duy Nguyen, Peter Vamplew, Saeid Nahavandi, Richard Dazeley, and Chee Peng Lim., A multi-objective deep reinforcement learning framework., Engineering Applications of Artificial Intelligence 96 (2020) 103915.

[16] Schwarting, Wilko, Javier Alonso-Mora, and Daniela Rus., Planning and decision-making for autonomous vehicles., Annual Review of Control, Robotics, and Autonomous Systems (2018).

[17] P.Suguna, B. Kirubagari, R. Umamaheswari., An Effective Cluster-Based Outlier Detection with Optimized Deep Neural Network for Epileptic Seizure Detection and Classification Model., International Journal of Engineering Trends and Technology 69.3(2021) 76-84.

[18] Amit Sagu, Nasib Singh Gill, Preeti Gulia., Artificial Neural Network for the Internet of Things Security., International Journal of Engineering Trends and Technology 68.11(2020) 129-136.