International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 7s
DOI: https://doi.org/10.17762/ijritcc.v11i7s.7526
***Article Received**: 25 April 2023 **Revised:** 15 June 2023 **Accepted:** 30 June 2023*

Reinforcement Learning and Advanced Reinforcement Learning to Improve Autonomous Vehicle Planning

[1]Dr. Avinash. J. Agrawal, [2]Dr. Rashmi R. Welekar, [3]Dr. Namita Parati, [4]Dr. Pravin R. Satav, [5]Dr. Uma Patel Thakur, [6]Dr. Archana V. Potnurwar

[1]Shri Ramdeobaba College of Engineering and Management, Nagpur, Maharashtra, India
[2]Shri Ramdeobaba College of Engineering and Management, Nagpur, Maharashtra, India
[3]Maturi Venkata Subba Rao (MVSR) Engineering College, Hyderabad, Telangana State, India
[4]Government Polytechnic, Murtijapur, Maharashtra, India
[5]Jhulelal Institute of Technology, Nagpur, Maharashtra, India
[6]Priyadarshini College of Engineering, Nagpur, Maharashtra, India
[1]agrawalaj@rknec.edu, [2]welekarr@rknec.edu, [3]namianand006in@gmail.com, [4]prsatav@gmail.com, [5]umapatel21@gmail.com, [6]archanapotnurwar@gmail.com

Abstract: Planning for autonomous vehicles is a challenging process that involves navigating through dynamic and unpredictable surroundings while making judgments in real-time. Traditional planning methods sometimes rely on predetermined rules or customized heuristics, which could not generalize well to various driving conditions. In this article, we provide a unique framework to enhance autonomous vehicle planning by fusing conventional RL methods with cutting-edge reinforcement learning techniques. To handle many elements of planning issues, our system integrates cutting-edge algorithms including deep reinforcement learning, hierarchical reinforcement learning, and meta-learning. Our framework helps autonomous vehicles make decisions that are more reliable and effective by utilizing the advantages of these cutting-edge strategies.With the use of the RLTT technique, an autonomous vehicle can learn about the intentions and preferences of human drivers by inferring the underlying reward function from expert behaviour that has been seen. The autonomous car can make safer and more human-like decisions by learning from expert demonstrations about the fundamental goals and limitations of driving. Large-scale simulations and practical experiments can be carried out to gauge the effectiveness of the suggested approach. On the basis of parameters like safety, effectiveness, and human likeness, the autonomous vehicle planning system's performance can be assessed. The outcomes of these assessments can help to inform future developments and offer insightful information about the strengths and weaknesses of the strategy.

Keywords: Deep Neural Network, Autonomous vehicle, path planning, reinforcement learning, motion planning.

I. INTRODUCTION

Industry demand for automated driving technology is rising quickly and is promising for daily transportation needs [1,2]. However, more work needs to be done on vehicle motion planning for automated driving, particularly in tackling various restrictions in areas with scant information. While macro-level information is known but unknown in such contexts, there is little micro-level information available. This creates difficulties for autonomous driving's motion planning that is exact, effective, and safe. Due to a number of issues, including insufficient information, traffic disturbances, and sensor data constraints, real traffic conditions in automated driving scenarios are fundamentally uncertain. These elements lead to complex traffic conditions that are challenging to anticipate in the present. To meet these issues, it is imperative to improve motion planning algorithms. Advanced methods should be used to manage sparse and ambiguous data effectively so that autonomous vehicles can make choices in real time. The motion planning algorithms can guarantee the security, effectiveness, and precision of autonomous driving systems by taking into consideration different limitations and accounting for uncertainties.

Without a driver's input, autonomous cars are able to perceive and comprehend their surroundings. They are able to do this by making use of a variety of sensing technologies, such as radar, ultrasound, localisation, and computer vision. These sensors acquire information about the environment, enabling the car to learn about the route, objects, and other important details. The autonomous vehicle can efficiently plan and travel along the planned path towards the intended destination by interpreting this sensory data using cutting-edge control algorithms. Autonomous vehicles can make wise decisions in real-time to ensure safe and effective navigation by integrating sensor data analysis and control algorithms.

In grand challenge [8] competitions, participating cars had to independently navigate off-road courses in 2004 and 2005, as well as an urban area course in 2007. These incidents proved that completely autonomous off-road and city driving is technologically possible. The DARPA Grand Challenge's success stimulated participation in autonomous vehicle development by for-profit businesses, startups, and research institutions. Significant advancements in the creation of autonomous vehicles have been accomplished since those early turning points. Many businesses and academic institutions have started their own projects, furthering technology advancement and bringing deployment of autonomous vehicles into the real world. These ongoing initiatives continue to influence transportation policy and open the door to an era of fully autonomous mobility.

The emphasis is on perceiving and filtering environmental input at the perceptual level. To do this, a number of sensors must be used to acquire data about the environment and the state of the vehicle. These sensors' data are combined by the sensing system, which offers significant decision-making inputs. The filtering mechanism helps to produce accurate estimates for unmeasurable states by lowering noise and uncertainty in the sensor outputs [12].

Before moving on to the planning level, three essential tasks must be finished. Mission planning entails figuring out the best routes and courses of action for completing tasks by addressing routing problems. Making decisions involves choosing the best course of action from a range of potential possibilities for the following time step. The creation of a trajectory the vehicle will follow in either space or time is the responsibility of path planning. In order to maintain vehicle stability and follow the desired path, signals from the planning level are finally received at the control level and put to use. The control system employs control algorithms to manage a number of vehicle factors, including braking, steering, and acceleration. The control system constantly modifies these parameters to make sure the vehicle properly follows the intended trajectory and continues to run safely and smoothly.

The goal of this research is to discuss the difficulty of planning for autonomous vehicles in traffic situations. The main goal is to imitate how experienced human drivers make decisions. The objective is to replicate the ideal driving strategy, which includes lane-shifting, lane and speed maintenance, acceleration, and braking, among other manoeuvres executed by expert drivers. Additionally, the planning strategy seeks to take into account the stochastic character of the driving styles displayed by other cars in the traffic environment. These aspects are taken into account in order to create a planning algorithm that enables autonomous vehicles to make choices and carry out manoeuvres in a way that is comparable to skilled human drivers when navigating through traffic.

II. REVIEW OF LITERATURE

The planning of autonomous automobiles has drawn an interest from the automaticdynamiccommunal [16], especially when it comes to data-driven strategies in complicated situations that are unknown and uncertain. For instance, in a study by Bernhard and Knoll [17], neural networks were used for autonomous vehicle planning to handle ambiguous input. However, their method presupposed total familiarity with the data of other cars, which might not be feasible in real-world settings. An innovative bi-level actor-critic strategy was developed by Zhang et al. [18] to successfully coordinate multi-agent decisions in highway merging scenarios.

Nick et al. [19] developed a method for categorizing traffic scenarios that incorporates CNN and RNN in order to forecast traffic circumstances for AVDMP. However, due to ambiguous and incomplete information, such as those seen in intersection situations, their method's stability may be jeopardized under intense traffic circumstances.

Chen et al. [21] created an end-to-end autonomous vehicle system using reinforcement learning (RL) and a sequential latent environmental descriptions expressed through a semantic bird-eye mask. Their method provided higher interpretability as compared to earlier ML techniques. In the context of automated driving, it is still necessary to take into account diverse restrictions and sparse information settings. Another study by Tang et al. [22] developed a soft actor-critic approach-based motion planning system for automated driving. Their approach tried to balance various approaches according to weights given to efficiency, comfort, and safety.

By combining rule-based and learning-based techniques, Zhu et al. [24] created a motion planning algorithm that took pedestrian distraction into account. According to their experimental findings, the learning-based approach worked better than the rule-based approach in addressing risky behaviours at signalized mid-block crosswalks. It was found that the learning-based approach occasionally led to irrational actions. These research show that motion planning for automated driving is still being improved. The suggested approaches have some potential, but there are still issues that need to be resolved, such as how to handle sparse information settings, take into account a number of limitations, and guarantee that actions are generated that are acceptable and safe. To increase the functionality and dependability of autonomous vehicle motion planning systems, more research and development is required in these areas.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 7s
DOI: https://doi.org/10.17762/ijritcc.v11i7s.7526
***Article Received**: 25 April 2023 **Revised:** 15 June 2023 **Accepted:** 30 June 2023*

A RL-based approach that incorporates negotiation tactics for autonomous vehicles operating in difficult contexts was developed by Shai et al. [27]. Two methodologies make up their approach: one that may be taught and the other linked to rigid limitations, including safety constraints. Although their method puts safety first, it might still need to be improved in order to be more adaptable in more complicated contexts, such as crossroads with a variety of heterogeneous cars and pedestrians. Using dynamic programming and Markov decision processes (MDP), Sarah M. Thornton [28] suggested a technique to regulate vehicle speed for safety while accounting for unsure pedestrians at crosswalks. The approach, however, could use improvements to handle even more unpredictable conditions and even narrow the search field for effective planning.

Even [29] while their approach worked well in their tests, it could still need to take into account the fact that automated vehicles frequently operate in areas with little information. Several different approaches to planning robot motion have been put forth, including rapid exploring random trees (RRT) for asymptotically optimal planning [33], Despite the fact that these techniques have made substantial advancements in robot motion planning, more work is still required to adapt them specifically for autonomous driving and take into account the particular characteristics of cars and autonomous driving situations.

The suggested method, RLTT, in this study varies from the aforementioned methodologies by enabling effective and safe automated driving in environments with little information while taking into account multiple limitations and multi-scale motion planning. In order to account for the variations in control freedom, navigation environments, and vehicle forms, the Trajectory Lattice Model (TLM), which was first developed for unmanned surface vehicles (USVs), has been modified for autonomous vehicles [15,39]. The Trajectory Point Selector (TPS), which limits the search space for RL navigation and boosts efficiency, is also introduced by the RLTT approach. The integration of TPS, RL, and TLM results in the proposal of a hierarchical framework, offering a unified approach for the RLTT method and model. The RLTT method, which incorporates multi-constraint and multi-scale motion planning while utilizing the developments in trajectory models and RL methods, offers a distinctive and all-encompassing solution.

Table 1: Related work comparison of different method and autonomous driving

Technique	Limitations	Technique Used	Advantages
RL with negotiation strategies [27]	May require improvement in complex environments	Reinforcement Learning	Incorporates negotiation strategies, considers safety constraints
MDP and dynamic programming [28]	Potential improvements for uncertain environments	Markov Decision Process	Considers uncertain pedestrians, focuses on safety
Conditional imitation learning [29]	Requires human guidance in sparse information environments	Imitation Learning	Effective in simulated and real environments
Probabilistic methods [30,31]	Further development needed for autonomous driving	Probabilistic Planning	Handles chance constraints and collision constraints effectively
Artificial potential fields [32]	Adaptation for autonomous driving required	Potential Fields	Offers a simple and intuitive solution
RRT [33]	Tailoring for autonomous driving needed	Sampling-based Planning	Provides asymptotically optimal planning in various scenarios
Fast marching methods [26]	Further development for autonomous driving required	Graph-based Planning	Efficient approach for path planning

III. TRAFFIC MODELING

A. The Markov Decision Processes:

The Markov decision process (MDP) can be used in a variety of fields, including robotics, economics, production, and automatic control. Bellman created a mathematical model called MDP that can demonstrate probabilistically how an agent interacts with his environment. The actor, who may be a student or a decision-maker, engages with the environment while keeping an eye on his or her state at all times and taking actions that could have an impact on later states.

A 6-tuple (S, A, T,y, D, R) is a common way to represent an MDP, and it looks like this:

- The changing conditions of the environment are represented by the (limited) set of potential states S.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 7s
DOI: https://doi.org/10.17762/ijritcc.v11i7s.7526
***Article Received**: 25 April 2023 **Revised:** 15 June 2023 **Accepted:** 30 June 2023*

- A denotes a (limited) set of possible actions from which the agent may choose under a specific circumstance.
- The state transit probability matrix, or T, gives the probabilities of changing between pairs of states.
- The reduction rate, which guarantees that total returns will converge over time, is [0, 1].
- The initial-state distribution, abbreviated as D, describes the likelihood of starting from each state.
- When specific actions are taken, certain states are rewarded according to the reward operate, or R.

An MDP provides a formal framework for modelling decision-making issues under uncertainty by defining these components. The agent's goal is normally to discover a strategy that optimizes its anticipated long-term cumulative profits. Based on the present condition, this policy chooses the agent's action. MDPs have shown to be useful tools for creating the best decision-making plans in a variety of situations. By taking into account the probabilistic nature of the environment and enabling agents to learn and adjust their behaviour depending on observable rewards and state transitions, they offer a strong framework for handling difficult issues.

According to MDPs, the result of performing an action at a particular state is purely decided by that state's action, and is unrelated to any precedingpositions or actions.

$$\mathrm{P}(\mathrm{st}+1|\mathrm{st},\mathrm{at},\mathrm{st}-1,\mathrm{at}-1,\ldots,\mathrm{s0},\mathrm{a0}) = \mathrm{P}(\mathrm{st}+1|\mathrm{st},\mathrm{at})$$

An MDP's main goal is to identify the agent's policy environment in which the policy is to:S →A details the street actions to be taken in light of the current situation. The goal is to figure out how to maximize the total reduced compensation over a specified period of time.

$$\pi * = \arg\max \pi\, \mathrm{E}\left[\sum \infty\, \mathrm{t} = 0\, \gamma\, \mathrm{t}\, \mathrm{R}\big(\mathrm{st}, \pi(\mathrm{st})\big)\right]$$

The discount rate, denoted by the term "γ" in the provided statement, controls how heavily future rewards are weighted in the cumulative discounted reward. The reward an agent receives for carrying out a policy-determined action at the current state π(st))is denoted by the word R(st, (st)). The MDP formulation (1) can be converted into a Markov chain by taking into account a particular policy, where the transition probabilities P represent the probability of transitioning between states in accordance with the selected policy.

B. Trajectory Lane System Modelling:

A key link between route planning and a vehicle's dynamic constraints is the TLM. It serves as a link to connect motion planning to actual environments. Algorithm 1 illustrates this connection. For precise and effective motion planning, the TLM is constructed using the vehicle's dynamic model and comprises special TLM rules.

Algorithm 1:

Input: *The acceleration a, velocity v, position (x, y), front wheel steering angle δ, differential time dt, interval time T, heading angle ψ, wheelbase $f\,rlen$, $i \leftarrow 0, t \leftarrow 0$;*

Output: *Trajectory lane model m;*

for *i* ***do*** $i \leftarrow i + 1$;

$x0 \leftarrow x + v * \cos(\psi)\, dt$;

$y0 \leftarrow y + v * \sin(\psi)\, dt$;

$\psi 0 \leftarrow \psi + \left(\frac{v}{f} rlen\right) * \delta * dt$;

$v0 \leftarrow v + a * dt$;

$x \leftarrow x0, y \leftarrow y0, \psi \leftarrow \psi 0, v \leftarrow v0$;

$t \leftarrow t + dt$;

if $t \le T$ then

Break

end

end

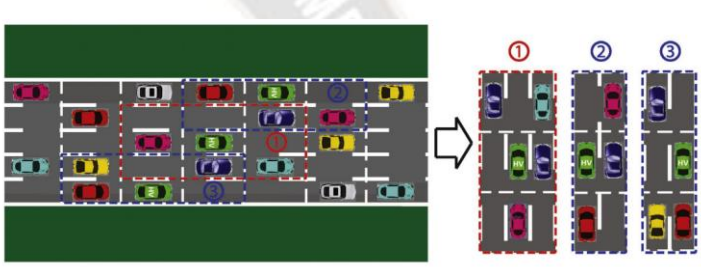


Figure 1. Lane StateRepresentation

As shown in Figure 1, the positions of the High Vehicle (HV) and the quantity and distribution of the nearby Electric Vehicles (EVs) establish the state representation of the Markov Decision Process (MDP). The HV is shown by the green car, and the road is separated into small cells by white dashed lines. The three situations in Figure 4 that the MDP states relate to are:

- HV is represented by nine cells in the middle lane.
- HV close to the left side of the road's edge.
- HV close to the right side of the road's edge.

Six cells are used to represent the current state for criteria 2 and 3. There are 256 internal-lane states (28 = 256) and 32 left/right-boundary states (25 = 32) when all conceivable combinations are taken into account. Thus, there are 256 + 2 32 = 320 MDP states in all. The fact that each vehicle is regarded as a point of mass and takes up a single cell must be kept in mind. This strategy is easily adaptable to highways with a variety of lanes and vehicles. The HV driver might favor passing the pink automobile on the left as opposed to the right. This work considers three types of roads: left-turn, right-turn, and straight roads in order to explore how road geometry affects observed driving behaviour. Consequently, there are 320 + 3 = 960 total states. Although just three road geometries

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 7s
DOI: https://doi.org/10.17762/ijritcc.v11i7s.7526
***Article Received**: 25 April 2023 **Revised:** 15 June 2023 **Accepted:** 30 June 2023*

are taken into account in this paper, the method can be expanded to include more road characteristics, such as various slopes (downhill, uphill, flat roads, etc.), as needed.

C. Search for Topological Paths in Automated Driving:

A topological map that is made expressly for high path search efficiency is created to enable large-scale and effective path searching. This topological map was produced using the Open Geospatial Consortium (OGC) path finding method. In relational databases, OGC specifies a straightforward feature model that works well for storing and retrieving geographic features. Nodes, lines, and surfaces are the three components that make up a spatial map and are crucial to its construction. Nodes are points without any spatial attributes, such as size and form. Lines have spatial characteristics like shape and length and are made up of a number of nodes. Surfaces are closed circles with shape and area characteristics.

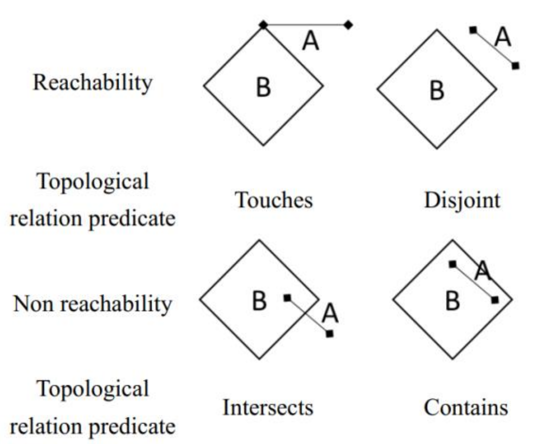


Figure 2: Representation of topological path

Algorithm 2:

```
Input: The set of obstacles On;
the start point S and the end point E,
the number of the ith obstacle of all points OPi , path point MP,
i ← 0;
Output: Macro − scale routing point MPn;
for i do
  i ← i + 1;
  j ← 0;
   for j do
     j ← J + 1;
      Compute to judge the relationships between the MP and Oij of the ith obstacle Oi ;
    if j = OPi then Break;
  end
 end
   if j = On then Break;
 end
end
Set the safety buffer and store the topological map;
Call Dijkstra algorithm and compute function for topological path MP, and
add MP to MPn;
return MPn.
```

Geographic information organization, querying, evaluation, and reasoning are all based on spatial relationships, which are the typical relationships between geographical entities. These connections include topological connections, metric connections, and directional connections. Spatial relationships have been represented by a number of models. The nine-cross model is employed in this work to depict the spatial topological links that were used to build the topological map.

IV. REINFORCEMENT LEARNING

The two primary kinds of reinforcement learning techniques are tabular solution techniques and function approximation techniques. For MDP issues with a limited or finite number of states and actions, tabular solution approaches are appropriate. These techniques include Monte Carlo, dynamic programming, and learning from temporal differences. The value function is used by dynamic programming methods to find the best policies. These algorithms include value iteration and policy iteration, for instance. They cannot be used to solve issues involving continuous state and action spaces since they necessitate complete environmental knowledge.

Algorithm 3: Q Learning

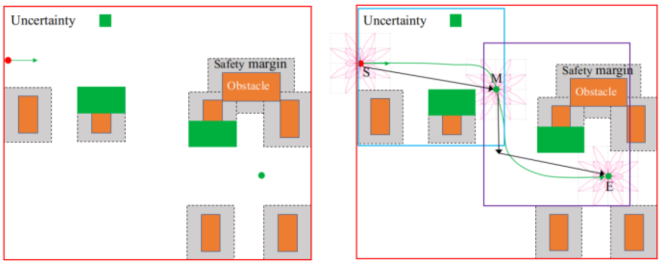
```
Input: The start point Sij and end point Eij of each segment from the MPn set
(i = 1,2,...,n, j = 1,2,...,n 0 ), S 0 represents the reachable points;
Output: The Q(s, a)in each segment;
Initialize the Q value (s, a), ∀ s ∈ S, a ∈ A, set parameter α, γ ;
  for episode z = 0 to Z do
    Select the action a according to the initial state s and e − greedy strategy;
    while s is not the terminal
      do
        Conduct a, then get reward r, next angle ψ 0 and next state s 0 according to
        transition strategy f , angle ψ, and state s;
        if next state s0 ∈ S 0 then
        Q(s, a) ← Q(s, a) + α[r + γmaxQ(s 0 , a 0 ) − Q(s, a)];
        s ← s 0 , a ← a 0 ;
      end
  end
         if s is terminal then
        Break;
       end
 end
return Q(s, a)
```

The transition function f of Q-learning is incorporated into the Trajectory Line Model (TLM) and the Trajectory Plan System (TPS). The function takes an action (a), the current position (Pc), an angle (A), and the final position (PT), and produces the next location (Pn), the reward (r), the completed status (T), and the next angle (A). The following is the transformation process.

If the current position Pc collides with an obstruction O, the function recovers the current position s, recognition r, completed state T, and angle A. The function returns the current state, compensation, final state, and angle A when the current position Pc reaches the final position PT. The function sets the following location Pn to be the present position Pc and

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 7s
DOI: https://doi.org/10.17762/ijritcc.v11i7s.7526
***Article Received**: 25 April 2023 **Revised:** 15 June 2023 **Accepted:** 30 June 2023*

the following angle An to be the present angle A if the angle A and action a match the corresponding angles At and At of the trajectory. The corresponding reward, r, is then calculated. The function returns the state s, reward r, done status T, and angle A if the next location Pn has hit an obstacle Oobstacle. When the next position Pn reaches the terminal position PT, the function returns the state s, reward r, done status T, and angle A. In order to gather pertinent path data, this reasoning is applied repeatedly. For each, the function returns the subsequent location Pn, reward r, finished status T, and subsequent angle An.



(a) (b)

Figure 3: Vehicle information environments that are certain and ideal (a) Without constraint (B) with constraint

Traditional approaches might not work as well in situations when unknown circumstances and vehicle limits are taken into account. Such a situation is shown in Figure 3, where the green arrow denotes dynamic limitations, such as direction and steer constraints, and the red circle denotes the start position. The green circle denotes the finish point.In this scenario, 35% of the obstacles randomly extend upwards or downwards by 35% when the environment is 65% certain and 35% uncertain. Specifically, 35% of the barriers' positions and shapes are undetermined, whereas 65% of them have known positions and shapes. 35% of the barriers alter randomly as the car travels around them.

By effectively navigating the uncharted territory of micro-scale motion planning, the Reinforcement Learning for Trajectory Tracking (RLTT) technique performs well in such settings. For large-scale and macro-scale routing planning (shown by the red box and black arrow from the start point to the end point), it makes use of the Trajectory Planning System (TPS). Additionally, it makes use of the Trajectory Lane Model (TLM) (seen as the deep pink line) for precise and micro-scale motion planning.The green arrow from the start point to the finish point in Figure 3 shows an example motion planning trajectory. The RLTT technique successfully addresses the vehicle restrictions and uncertain environment, ensuring effective and precise motion planning at both the macro and micro scales.

V. RESULTS AND DISCUSSION

To thoroughly evaluate and examine the data, more experiments were run in the identical settings, the RLTT approach for motion planning in automated driving, and comparison trials in both certain and uncertain contexts were the main focuses of the experiments. To evaluate the efficacy of the suggested approaches, experiments were also conducted to compare RLTT with other RL algorithms and conventional algorithms. Finally, RLTT was used to plan movements in unpredictable corridor situations, illustrating the approach's usefulness.

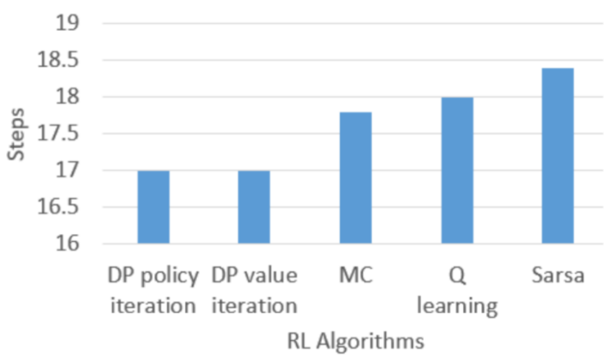


Figure 4: Different Path search RL algorithm

Figure 4 shows how several RL path planning techniques are applied in automatic driving. The outcomes show that dynamic programming (DP) value and policy iterations [8] achieve the fewest steps, demonstrating their capacity to identify the shortest path for automated driving. Iterations of the DP value and policy, however, might not be ideal for automatic driving in ambiguous and uncharted terrain. This is because the knowledge necessary for these methods knowledge of the probability and reward transitions may not be present in such situations. These approaches often need accurate knowledge of the environment model, which may not be possible in real-world situations.

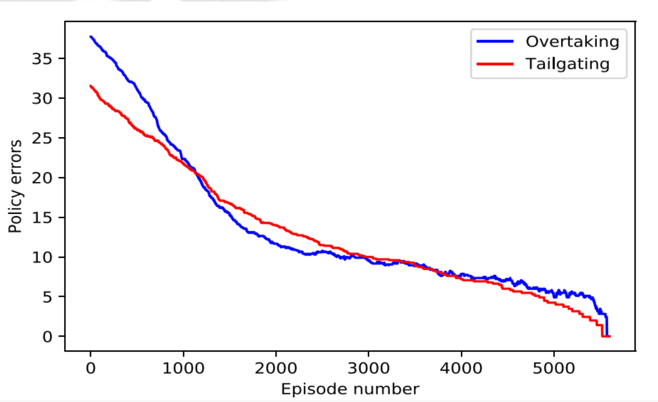


Figure 5: Performance of learning process during overtaking and tailgating

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 7s
DOI: https://doi.org/10.17762/ijritcc.v11i7s.7526
***Article Received**: 25 April 2023 **Revised:** 15 June 2023 **Accepted:** 30 June 2023*

Various contexts, including both certain environments with known knowledge and uncertain environments with unknown information, were used for the experiments. These habitats, shown in Figure 6, were identical save for the randomly placed obstacles. In comparison to five certain trials, which travelled an average distance of 71.98 meters, five doubtful experiments covered an average distance of 54.87 meters. The outcomes of these tests show that the RLTT approach was capable of navigating and coping with the environmental uncertainties efficiently.

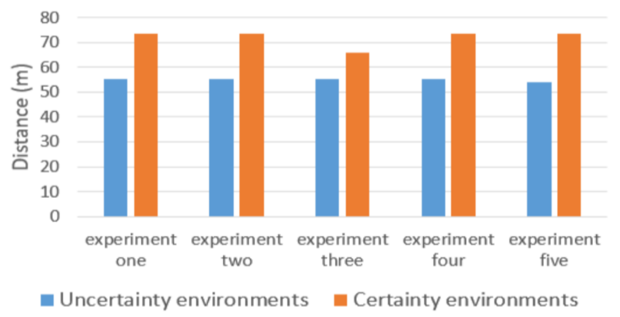


Figure 6: Automated driving for uncertain and certain environments

Both the RLTT approach and the Q-learning algorithm, a well-known traditional algorithm for vehicle navigation, were applied in five experiments in unexplored locations. Figure 7 shows the computing time for every experiment as well as the time spent on average computing for the five tests. The motion planning averaged calculation times for the RLTT technique and the algorithm for Q-learning were found to be 0.425 seconds as well as 54.279 seconds, respectively, respectively. Furthermore, it was noted that the trajectory distance acquired with the Q-learning approach was typically greater than the trajectory distance produced with the RLTT method.

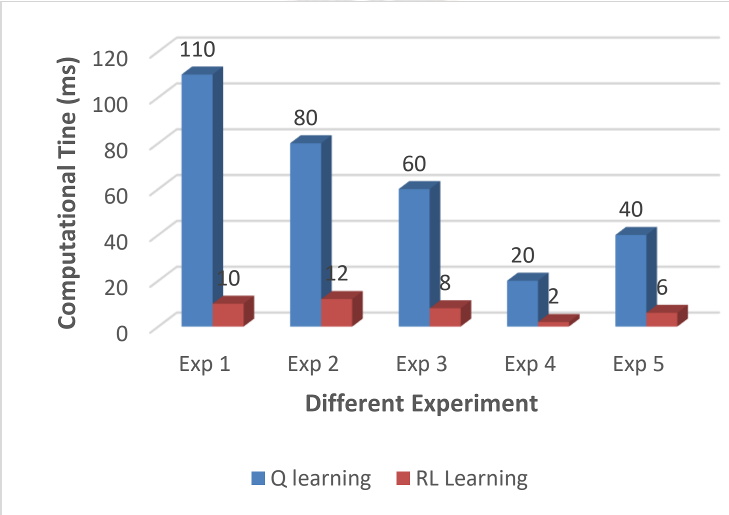


Figure 7: Utilizing the Q learning algorithm and RLTT for motion planning

One random experiment was chosen, and the trajectory distances were calculated in order to further examine the performance. The distance was determined to be 95.834 meters using the RLTT method and 110.735 meters using the Q-learning algorithm. According to these results, the RLTT approach performed better than the traditional RL algorithm, obtaining a lower trajectory distance while taking less time for path discovery.

VI. CONCLUSION

This article describes the creation of a constrained RL method and the theoretical analysis employed therein, which is based on TLM and TPS hypotheses. The goal was to produce an efficient and safe movement plan for autonomous driving in poorly informed areas while taking into account safety, fluidity, and dynamic restrictions. A number of experiments were used to test the efficiency of the suggested method. The experimental findings show that the suggested approach is not only practical but also adaptable to different kinds of vehicle navigation and control. It can be used, for instance, with ground robots for parking manoeuvres and logistics settings. The RLTT method's effectiveness in these studies creates new opportunities for robotics research and development. Future research will involve running more tests and attempting to improve the suggested strategy by taking on more challenging challenges. These initiatives will help the RLTT method's capabilities and adaptability to diverse real-world situations.

REFERENCES:

[1] Williams, G.; Drews, P.; Goldfain, B.; Rehg, J.M.; Theodorou, E.A. Information-theoretic model predictive control: Theory and applications to autonomous driving. IEEE Trans. Robot. 2018, 34, 1603–1622.

[2] Likmeta, A.; Metelli, A.M.; Tirinzoni, A.; Giol, R.; Restelli, M.; Romano, D. Combining reinforcement learning with rule-based controllers for transparent and general decision-making in autonomous driving. Robot. Auton. Syst. 2020, 131, 103568.

[3] Borkar, P., Wankhede, V.A., Mane, D.T. et al. Deep learning and image processing-based early detection of Alzheimer disease in cognitively normal individuals. Soft Comput (2023). https://doi.org/10.1007/s00500-023-08615-w

[4] Ajani, S.N., Mulla, R.A., Limkar, S. et al. DLMBHCO: design of an augmented bioinspired deep learning-based multidomain body parameter analysis via heterogeneous correlative body organ analysis. Soft Comput (2023). https://doi.org/10.1007/s00500-023-08613-y

[5] Hang, P.; Lv, C.; Huang, C.; Cai, J.; Hu, Z.; Xing, Y. An Integrated Framework of Decision Making and Motion Planning for Autonomous Vehicles Considering Social Behaviors. IEEE Trans. Veh. Technol. 2020, 69, 14458–14469.

[6] Sutton, R.S.; Barto, A.G. Reinforcement Learning: An Introduction; MIT Press: Cambridge, MA, USA, 2018.

[7] Gu, S.; Kuba, J.G.; Wen, M.; Chen, R.; Wang, Z.; Tian, Z.; Wang, J.; Knoll, A.; Yang, Y. Multi-agent constrained policy optimisation. arXiv 2021, arXiv:2110.02793.

[8] Gu, S.; Yang, L.; Du, Y.; Chen, G.; Walter, F.; Wang, J.; Yang, Y.; Knoll, A. A Review of Safe Reinforcement Learning: Methods, Theory and Applications. arXiv 2022, arXiv:2205.10330.

[9] S. N. Ajani and S. Y. Amdani, "Probabilistic path planning using current obstacle position in static environment," 2nd International Conference on Data, Engineering and Applications (IDEA), 2020, pp. 1-6, doi: 10.1109/IDEA49133.2020.9170727.

[10] S. Ajani and M. Wanjari, "An Efficient Approach for Clustering Uncertain Data Mining Based on Hash Indexing and Voronoi Clustering," 2013 5th International Conference and Computational Intelligence and Communication Networks, 2013, pp. 486-490, doi: 10.1109/CICN.2013.106.

[11] Brunke, L.; Greeff, M.; Hall, A.W.; Yuan, Z.; Zhou, S.; Panerati, J.; Schoellig, A.P. Safe learning in robotics: From learning-based control to safe reinforcement learning. Annu. Rev. Control. Robot. Auton. Syst. 2021, 5, 411–444.

[12] Silver, D.; Huang, A.; Maddison, C.J.; Guez, A.; Sifre, L.; Van Den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; et al. Mastering the game of Go with deep neural networks and tree search. Nature 2016, 529, 484–489.

[13] Sunil Kumar, M. ., Sundararajan, V. ., Balaji, N. A. ., Sambhaji Patil, S. ., Sharma, S. ., & Joy Winnie Wise, D. C. . (2023). Prediction of Heart Attack from Medical Records Using Big Data Mining. International Journal of Intelligent Systems and Applications in Engineering, 11(4s), 90–99. Retrieved from https://ijisae.org/index.php/IJISAE/article/view/2575

[14] Zhou, C.; Gu, S.; Wen, Y.; Du, Z.; Xiao, C.; Huang, L.; Zhu, M. The review unmanned surface vehicle path planning: Based on multi-modality constraint. Ocean. Eng. 2020, 200, 107043.

[15] Claussmann, L.; Revilloud, M.; Gruyer, D.; Glaser, S. A review of motion planning for highway autonomous driving. IEEE Trans. Intell. Transp. Syst. 2019, 21, 1826–1848.

[16] Bernhard, J.; Knoll, A. Robust stochastic bayesian games for behavior space coverage. In Proceedings of the Robotics: Science and Systems (RSS), Workshop on Interaction and Decision-Making in Autonomous-Driving, Virtual Session, 12–13 July 2020

[17] Zhang, H.; Chen, W.; Huang, Z.; Li, M.; Yang, Y.; Zhang, W.; Wang, J. Bi-level actor-critic for multi-agent coordination. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 7325–7332.

[18] Harmening, N.; Biloš, M.; Günnemann, S. Deep Representation Learning and Clustering of Traffic Scenarios. arXiv 2020, arXiv:2007.07740.

[19] Zhang, L.; Zhang, R.; Wu, T.; Weng, R.; Han, M.; Zhao, Y. Safe reinforcement learning with stability guarantee for motion planning of autonomous vehicles. IEEE Trans. Neural Netw. Learn. Syst. 2021, 32, 5435–5444.

[20] Chen, J.; Li, S.E.; Tomizuka, M. Interpretable End-to-End Urban Autonomous Driving With Latent Deep Reinforcement Learning. IEEE Trans. Intell. Transp. Syst. 2021, 23, 5068–5078.

[21] Tang, X.; Huang, B.; Liu, T.; Lin, X. Highway Decision-Making and Motion Planning for Autonomous Driving via Soft Actor-Critic. IEEE Trans. Veh. Technol. 2022, 71, 4706–4717.

[22] Haarnoja, T.; Zhou, A.; Abbeel, P.; Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In Proceedings of the International Conference on Machine Learning Research, Stockholm, Sweden, 10–15 July 2018; pp. 1861–1870.

[23] Zhu, H.; Han, T.; Alhajyaseen, W.K.; Iryo-Asano, M.; Nakamura, H. Can automated driving prevent crashes with distracted Pedestrians? An exploration of motion planning at unsignalized Mid-block crosswalks. Accid. Anal. Prev. 2022, 173, 106711

[24] Achiam, J.; Held, D.; Tamar, A.; Abbeel, P. Constrained policy optimization. In Proceedings of the International Conference on Machine Learning, PMLR, Sydney, Australia, 6–1 August 2017; pp. 22–31.

[25] Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal policy optimization algorithms. arXiv 2017, arXiv: 1707.06347.

[26] Dr. S. Praveen Chakkravarthy. (2020). Smart Monitoring of the Status of Driver Using the Dashboard Vehicle Camera. International Journal of New Practices in Management and Engineering, 9(01), 01 - 07. https://doi.org/10.17762/ijnpme.v9i01.81

[27] Shalev-Shwartz, S.; Shammah, S.; Shashua, A. Safe, multi-agent, reinforcement learning for autonomous driving. arXiv 2016, arXiv:1610.03295.

[28] Thornton, S. Autonomous Vehicle Speed Control for Safe Navigation of Occluded Pedestrian Crosswalk. arXiv 2018, arXiv: 1802.06314.

[29] Codevilla, F.; Miiller, M.; López, A.; Koltun, V.; Dosovitskiy, A. End-to-end driving via conditional imitation learning. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; pp. 1–9.

[30] Rajesh Patel, Natural Language Processing for Fake News Detection and Fact-Checking , Machine Learning Applications Conference Proceedings, Vol 3 2023.

[31] Dai, S.; Schaffert, S.; Jasour, A.; Hofmann, A.; Williams, B. Chance constrained motion planning for high-dimensional robots. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 8805–8811.

[32] Thomas, A.; Mastrogiovanni, F.; Baglietto, M. Probabilistic Collision Constraint for Motion Planning in Dynamic Environments. arXiv 2021, arXiv:2104.01659.

[33] Mohanan, M.; Salgoankar, A. A survey of robotic motion planning in dynamic environments. Robot. Auton. Syst. 2018, 100, 171–185.

[34] Webb, D.J.; Van Den Berg, J. Kinodynamic RRT*: Asymptotically optimal motion planning for robots with linear

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 7s
DOI: https://doi.org/10.17762/ijritcc.v11i7s.7526
***Article Received**: 25 April 2023 **Revised:** 15 June 2023 **Accepted:** 30 June 2023*

dynamics. In Proceedings of the 2013 IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, 6–10 May 2013; pp. 5054–5061.

[35] Gammell, J.D.; Srinivasa, S.S.; Barfoot, T.D. Informed RRT*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic. In Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, USA, 14–18 September 2014; pp. 2997–3004.

[36] Janson, L.; Schmerling, E.; Clark, A.; Pavone, M. Fast marching tree: A fast marching sampling-based method for optimal motion planning in many dimensions. Int. J. Robot. Res. 2015, 34, 883–921

[37] Du, Z.; Wen, Y.; Xiao, C.; Zhang, F.; Huang, L.; Zhou, C. Motion planning for unmanned surface vehicle based on trajectory unit. Ocean. Eng. 2018, 151, 46–56.

[38] Zhu, M.; Xiao, C.; Gu, S.; Du, Z.; Wen, Y. A Circle Grid-based Approach for Obstacle Avoidance Motion Planning of Unmanned Surface Vehicles. arXiv 2022, arXiv:2202.04494.

[39] Gu, S.; Zhou, C.; Wen, Y.; Xiao, C.; Knoll, A. Motion Planning for an Unmanned Surface Vehicle with Wind and Current Effects. J. Mar. Sci. Eng. 2022, 10, 420

[40] Gu, S.; Zhou, C.; Wen, Y.; Zhong, X.; Zhu, M.; Xiao, C.; Du, Z. A motion planning method for unmanned surface vehicle in restricted waters. Proc. Inst. Mech. Eng. Part M J. Eng. Marit. Environ. 2020, 234, 332–345.

[41] McNaughton, M.; Urmson, C.; Dolan, J.M.; Lee, J.W. Motion planning for autonomous driving with a conformal spatiotemporal lattice. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 4889–4895.

[42] Khetani, V. ., Gandhi, Y. ., Bhattacharya, S. ., Ajani, S. N. ., & Limkar, S. . (2023). Cross-Domain Analysis of ML and DL: Evaluating their Impact in Diverse Domains. International Journal of Intelligent Systems and Applications in Engineering, 11(7s), 253–262.