World Journal of Advanced Research and Reviews
eISSN: 2581-9615 CODEN (USA): WJARAI
Cross Ref DOI: 10.30574/wjarr
Journal homepage: https://wjarr.com/

(RESEARCH ARTICLE)

Check for updates

Intelligent path planning technique for autonomous vehicles using improved harmony search optimized fuzzy control

Venkata Satya Rahul Kosuru [1,*] and Ashwin Kavasseri Venkitaraman [2]

[1] *Department of "Electrical & Computer Engineering", Lawrence Technological University. Southfield, MI 48075, USA.*
[2] *Department of" Engineering", University of Cincinnati. Cincinnati, OH 45221.*

World Journal of Advanced Research and Reviews, 2025, 25(01), 468-481

Publication history: Received on 25 November 2024; revised on 04 January 2025; accepted on 06 January 2025

Article DOI: https://doi.org/10.30574/wjarr.2025.25.1.0010

Abstract

Path planning is one of the most crucial elements of autonomous driving (AD). Due to its capacity to directly make judgments based on observation and learn from the environment, learning-based path planning techniques are of interest to many academics. The standard reinforcement learning approach of the deep Q-network has made major strides in AD since the agent normally learns driving tactics simply by the intended reward function, which is difficult to adapt to the driving scenarios of urban roadways. However, such methodologies rarely use the global path data to address the problem of directional planning, like turning around at an intersection. In addition, the link between different motion instructions like these might easily lead to an erroneous prediction of the route orders due to the fact that the steering and the accelerator are independently governed in a real-world driving system. This research proposes and implements a Provisional Cross-layered Deep Q-Network (PC-DQN) for path planning in end-to-end autonomous vehicles, where the universal path is employed to direct the vehicles from the starting point to ending point. We employ the concept of Improved Harmony Search optimized fuzzy control (HIS-FC) and propose a defuzzification approach to increase the stability of anticipating the values of various path instructions in order to manage the reliance of distinct path instructions in Q-networks. We carry out extensive tests in the CARLA simulator and contrast our approach with cutting-edge approaches. The suggested strategy outperforms existing methods in terms of learning efficiency and driving reliability, according to experimental findings.

Keywords: Autonomous vehicles; Path planning; Fuzzy logic; Provisional Cross-layered Deep Q-Network (PC-DQN); Improved Harmony Search optimized fuzzy control (HIS-FC)

1. Introduction

In recent years, study on vehicles has seen fast growth, and it has expanded to span a variety of fields, such as robotics, computer science, and engineering, among others. In addition, it is important to mention that scientific progress has been achieved by automobile manufacturers; yet, due to the financial sensitivity of their work, these companies do not often make the specifics of their methods or algorithms publicly available. The research on autonomous driving cars have been a subject of intense interest in both the business world and the academic world [1].An Autonomous vehicle systems could one day be able to take the place of human drivers and autonomously manage motion according to factors such as the status of the road and the vehicle[2]. Therefore, autonomous cars are being examined as potential solutions to increase both the effectiveness of roads and the safety of driving. It is anticipated that, as a consequence of advancements in sensing technologies, electronically controlled smart cars, technology, and machine learning will become more sophisticated and humanized [3]. Therefore, the development of choices- constructing and preparing systems for autonomously driving cars that are capable of navigating dynamic traffic conditions is an important area of study [4]. One of the primary responsibilities of the modules responsible for decision-making and planning is to carry out an obstacle avoidance operation based on information from mixed perception. Making decisions, mapping out

* Corresponding author: Venkata Satya Rahul Kosuru

possible routes, and keeping track of where you've been are three of the most important aspects of a method of avoiding obstacles [5]. Thus, obstacle avoidance route planning for AD offers practical usefulness and application potential. Modules for route planning might provide a reference path that could navigate around obstacles while still adhering to the requirements for safe driving and vehicle dynamics limits [6]. The primary goal of providing a vehicle is arranging. with a route that is secure and free of potential collisions on the way to its destination. This must be accomplished while considering the dynamics of the vehicle, their powers for maneuvering in the presence of obstacles, as well as traffic regulations Regarding the limits of roads. The planning process, which utilizes a significant amount of memory and places a significant demand on the computer's processing power is performed in tandem with the vehicle's other normal activities [7]. The Harmony Process (HS) method is now one of the most prominent metaheuristics that is used to address a vast range of diverse sorts of issues. In this research, we analyze the path planning for Autonomous vehicle using improved Harmony search optimized fuzzy control.

The further portion of the article includes part 2 indicates the Related works, part 3 describes the suggested work, part 4 indicates the result and discussion and part 5 indicates the conclusion.

2. Related Works

In this paragraph of the article, we will discuss the other pieces of art relevant to the subject. The purpose of the study [8] is to offer a dynamic real-time path planning system for independent vehicles that can avoid both stationary and moving impediments. The approach that has been developed for the design of paths finds not only the best path, also the right rate of acceleration and cruising speed for a certain vehicle. The study [9] developing a system for obstacle avoidance route planning, the initial step is to create a safety model of obstacle avoidance. This is done by evaluating the way in which a human driver navigates around obstacles. The research[10] organize the several approaches to for the unmanned aerial vehicles, development impact (UAVs) that are currently available into three primary groups: the representational techniques, the cooperative techniques, and the non-cooperative strategies. Coverage and connection of the UAVs' network communication are evaluated and discussed using these methodologies. An evaluation of the current ideas has also been carried out, with each type of UAV path planning serving as the basis for the evaluation. The paper [11] presented an implementation an analysis of the little cuckoo search method, later propose a new parallel communications plan. The unmanned robot's memory can be effectively preserved by the condensed scheme. The parallel approach can improve precision and speed up convergence. The key limits for path planning are restricted data transfer capabilities, electricity, and underwater sensor technologies. The maritime environment is exposed to a wide range of demanding variables, which can be classed as atmospheric, coastal, or gravitational. The undersea environment can be classified as predictable or unexpected depending on whether the influence of these components can be approximated [12].

The article [13] intends to investigate and assess the studies that have already been conducted in the field of coverage path planning issues, particularly those that make use of unmanned aerial vehicles (UAVs). In the research [14]a new based on reinforcement learning Algorithm for grey wolf optimization known as RLGWO has been released. as a potential solution to this issue. The suggested approach includes reinforcement learning, in which the person is commanded to switch operations in an adaptive manner based on the collected performance. The author of [15] mentions that a path planning technique for autonomous vehicles. The goal of the algorithm is to generate man oeuvres that are both feasible and smooth in environments that are not structured. The study [16] take into consideration a more generic situation, which involves many ground vehicles and numerous unmanned aerial aircraft. In this paper, they formalized the “multi-vehicle-assisted multi-UAV path planning issue”, which is a dual problem including the planning of routes and the assignment of tasks (RPTSP).

The study [17] describes the research progress of path planning based on the multi-modality constraint. The research [18] offer a novel technique for path planning that is based on artificial potential field and ant colony optimization (ACO). The technique that was suggested considers both dynamic risks and static impediments in order to construct an artificial field that represents the environment for the purpose of creating a path that avoids collisions. The paper [19] discusses compare and contrast certain controls used by three different Omni wheeled firefighting robots because of the range of agility that each offer. The research [20] offered a method for predicting the paths that mobile robots would take in the visible plane by employing an above camera and utilizing Type of interval -2 fuzzy logic (IT2FIS). In this essay, they discuss an approach to obstacle-free path planning that is based on visual serving. Heuristic and conventional approaches are the two distinct classifications that may be used to mobile robots' approaches to the planning and execution of their paths. The fact that analytical approaches are too complicated for use in intangible applications is the primary flaw in the system; enumerative methods, on the other hand, are hampered by the sheer volume of the search area [21]. The study [22] makes use of fuzzy logic, taking into account a variety of criteria. After

that, in order to compute the route planning, four strategies are utilized: an initial suggestion referred to as fuzzy logic, attraction, a PSO algorithm, and an ANFIS algorithm.

2.1. Problem Statement

Motion planning, also known as path planning (PP), is a computational trouble with which the goal is to identify a series of valid configurations that moves an item from its origin to its destination. The term is often used in the fields of computational geometry, computer animation, robotics, and computer gaming. The problem of PP is an essential part of the method for planning UAV missions, which must determine the most efficient path across the complex environment. PP is one of the most critical problems in unmanned aerial vehicles (UAVs) for determining the best route between source and destination. Even if there are a lot of study suggestions on the problems of PP for UAVs in the existing literature, there are still problems with target localization and identification. In this research, we propose employing harmony search optimized fuzzy control to overcome the challenges associated with local path planning

3. Proposed methodology

As artificial intelligence technology has continued to advance, there has been a concomitant acceleration in the development of autonomous vehicle technology. The performance of it on highways, including the precise journey path, has been described. Hardware and software make up the two primary components that make up autonomous driving systems. The term "hardware" refers to the mechanical components of the system, include sensors, actuators, and Vehicle-to-Vehicle (V2V) hardware. Sensors are employed for the purpose of observing their surrounding environment. An actuator is utilized to run the subsystems of the vehicle. V2V equipment is installed on each vehicle to allow for communication and the sharing of data between the cars. Perception, route planning, and command and control systems are all components of the software modules. The data gathered from the sensors is compiled by the perception module into a three-dimensional map of the area immediately surrounding the vehicle. This gives the car the ability to comprehend its surroundings, as seen in. After that, a design of the route can be planned out based on the commands given by the user. After gathering information about the real-world setting, the control system will, as a last step, issue directives to the hardware of the vehicle. Detection and ranging of light, often known as (LiDAR), is an active-ranging technique that determines the separation between measuring items with the amount of time it takes for a laser light pulse to make a full circuit. Lasers have a minimal divergence, which helps to reduce power degradation; the measured distance can range up to 200 meters (m) even when exposed to bright sunshine.

3.1. Provisional Cross-layered Deep Q-Network (PC-DQN)

A PC-DQN form Reward-based learning that was used to manage the agent PC-DQN in so as to give commands acting devices like the gas pedal, the and the brake lever the steering wheel. This was accomplished through the use of a reinforcement learning approach called PC-DQN. In order to do this, descriptions of the development of the PC-DQN model procedure, and its reward system were necessary. PC-DQN can be effective when applied to difficult judgements but necessitates the collection of an enormous amount of data, such as extensive state input, intricate the environment, nuanced data on actions.

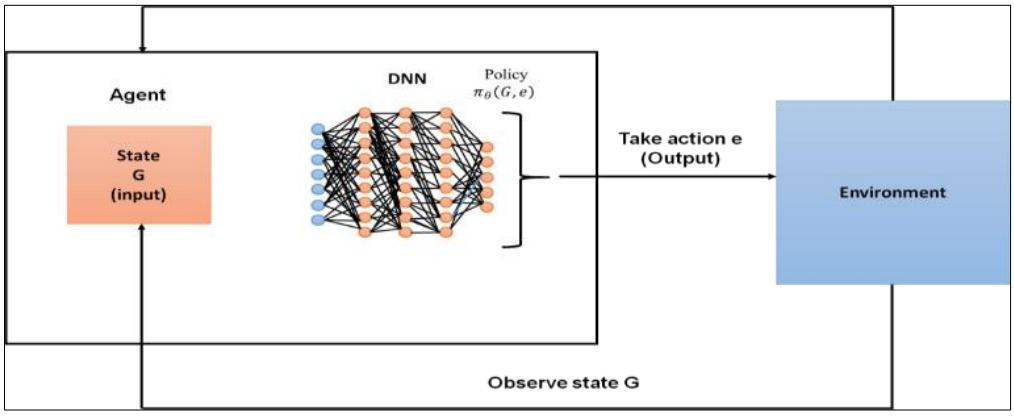

Figure 1 Deep Q-Learning process

In PC-DQN procedure, is depicted in Figure 1, involves training a neural network to make an approximation of the Q-value function. The present state is utilized as the input, and a list of potential actions is produced as the result. The PC-

DQN paradigm is responsible for computing the purpose of rewards in order to select the appropriate action for a certain state. It also stores its experience in a queue of memory. After that, the model makes use of the experiences stored in the memory buffer in order to train the target model, which determines the most effective method by which to maximize its reward.

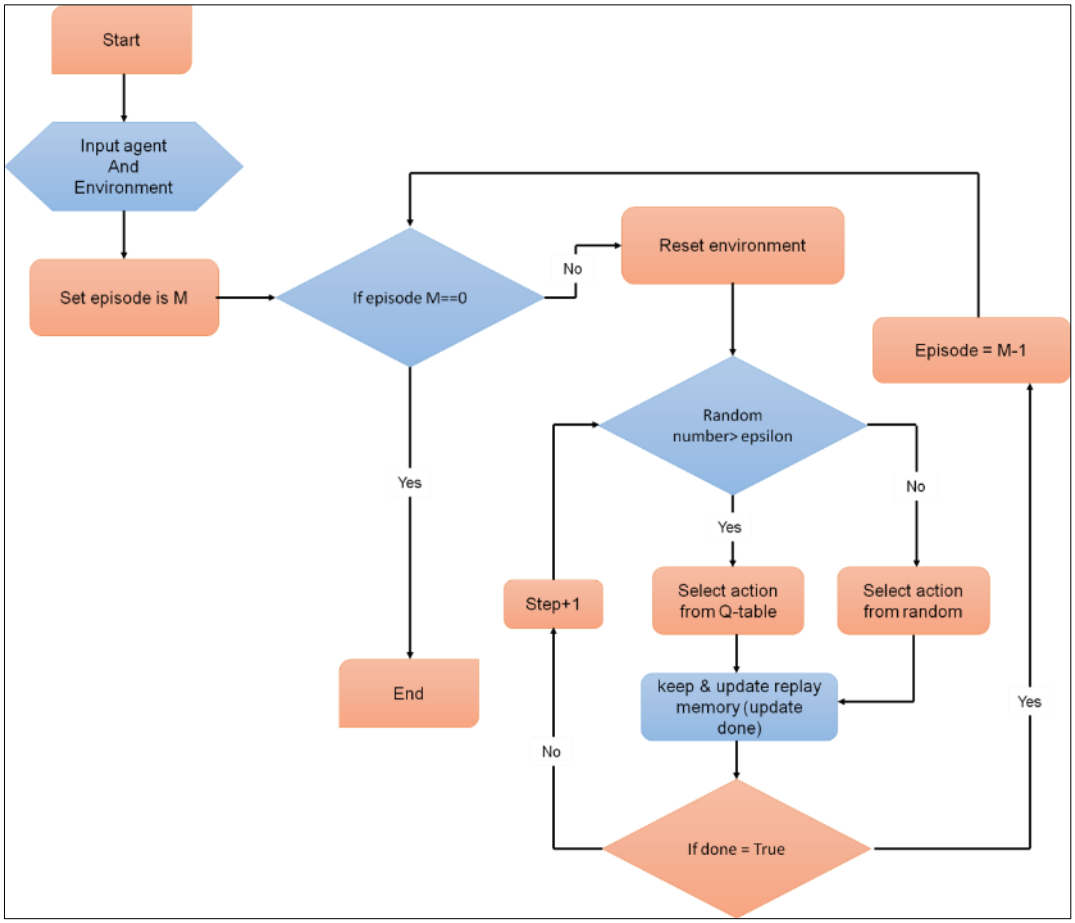

Figure 2 Process to select the action in PC-DQN model

Figure 2 shows the PC-DQN technique is trained by accumulating the optimal reward and state in a memory buffer. Model configuration, action configuration, reward configuration, and hyper parameter configuration make up the rest of this section. These four factors play crucial role in ensuring that the PC-DQN model learns in a way that is suitable for the trials. Below are the specifics for designing the control system.

3.1.1. Model Setting

PC-DQN is a neural network model with dense and buried layers and nodes. For each layer, we employ an activation function that is tailored to the specific user-defined data. The activation function will be set to a linear function unless the user changes it. Minimizing gradient vanishing and explosion requires careful consideration of the activation function configuration.

3.1.2. Action Setting

We define four actions in the action setting stage: forward, braking, and steering wheel movements within a 35-degree range. Depending on what the PC-DQN algorithm thinks, the gas and brake pedals will be pressed or released.

3.1.3. Reward Setting

Three rewards can be found in the reward configuration: the base reward(k_n), the penalty reward (k_b), and the bonus reward (k_{ay}). These should be specified as continuous functions to promote smooth and steady learning by the algorithm. We used a hyperbolic function with different reward weights for each possible outcome. Providing a satisfactory answer will result in a bonus payment of twice the original amount. A negative reward will be given if the

agent comes up with a bad answer. Each reward condition serves a different purpose during training. The full compensation is presented as

$$K_D = K_n + K_b + K_{ay} \text{..... (1)}$$

As shown in Equation, the primary incentive (t_n) is calculated based on the travel time between the vehicle and its final destination (t_n). (2), to create this incentive, the quadratic polynomial was used to assign a negative value to the farthest position and the highest value to the closest. This motivates the agent to travel there. By iteratively plugging values into the quadratic polynomial equation, the maximum reward was found to be 3000, and the minimum found to be 9500.

$$t_n = -e(t_n - p)^2 + v \text{..... (2)}$$

$$t_n = \sqrt{(y_t - y_c)^2 + (x_t - x_c)^2} \text{..... (3)}$$

where$t_n$is the travel distance, $x_t$and $y_t$are the x and y coordinates of the destination, $x_c$and $y_c$are the x and y coordinates of the vehicle, and a, b, and c have values of 2.00, 98.00, and 3000, respectively.

k_b, or penalty reward, is determined by comparing the distance learned from the LiDAR to a predefined target (4). The agent will be able to make judgements more quickly while still accurately detecting obstacles if the quantity of the LiDAR point cloud information is reduced. The movement towards the destination was slowed down by the introduction of a penalty incentive to encourage the dodging of obstacles. This equation is set up as a polynomial of the fourth degree, which is meant to simulate human behavior. This penalty reward has a little positive value if the LiDAR finds the obstruction at its farthest location. The algorithm's decision-making is influenced by the penalty reward, which has a negative value when the space between the vehicle and an obstacle is lesser than 50 m and rapidly declines when it is less than 15 m and approaches 0 m (the critical point). Distances needed to halt from 40 kilometers per hour are used to determine the crucial zones. By using trial and error, we were able to calculate penalty reward coefficients that strike a balance between the primary reward and the separation between the vehicle and the obstacle.

$$K_b = e_1 t_f^4 + e_2 t_f^3 + e_3 t_f^2 + e_4 t_f + e_5 \text{.....(4)}$$

$$t_f = min\sqrt{y_0^2 + x_0^2 + h_0^2}, \text{..... (5)}$$

Wherex_o, $y_o$and $h_o$are scalars of distance vectors from the vehicle to the barrier and signify the nearest identified places. Coefficients a1, a2, a3, and a4 are 0.0011, 0.15, 6.60, and 145, while a5 is 3040. If the agent is able to avoid collisions throughout the episode, they will get an additional prize (K_{ay}) on top of their regular payout. When the vehicle is able to stop within 10 meters of the obstruction, an additional prize is added. The motivation was intended to influence safe driving practices. In terms of swaying, one's decision, this reward is substantial. The incentive is shown as

$$K_{ay} = p_1 y_f + p_2, \text{..... (6)}$$

Where $y_f$is the distance between the vehicle and the obstacle. Both $p_1$and p_2 are assigned the values of 200.00 and 2200.

3.1.4. Hyper parameter setting

When considering the network architecture and the training process, the hyper parameter values may be broken down into two distinct categories. Number of hidden layers, number of units, dropout, network weight initialization, and activation function are all examples of network structure hyper parameters.

3.2. Improved Harmony Search optimized fuzzy control (HIS-FC)

3.2.1. Harmony Search

In 2001, Zong Woo Geem was motivated to create the harmony search algorithm after seeing the improvisational process of jazz bands. There is a one-to-one correspondence between the musicians and the decision factors, with the pitch range of each instrument representing the possible values for those variables. For this example, the band's ability to create musical harmony is represented by a solution vector, and the degree to which their performance is well received by the public is represented by the objective function. The five-stage HS method is as follows: At this stage, the optimization problem and the values for the HS technique's variables are set as the harmony memory size (HMS), which is equivalent to the solution vectors saved in the memory; HMCR denotes harmony memory considering rate; PAR

denotes pitch adjusting rate. To begin, the harmony memory (HM) is given a random set of solution vectors created by the HMS.

An improvised new harmony: Here, we construct a New Harmony vector by using a trifecta of methods: randomization, memory recall, and pitch manipulation. Memorandum on the subject of harmony: The New Harmony vector replaces the poorest harmony of the HM if and only if it has a better objective function value. The highest goal function solution is chosen after all iterations.

3.2.2. Improved HSA

The HSA has five phases: initialization of parameters, harmony memory initialization, improvisation of a New Harmony, harmony memory update, and termination criteria verification. We propose modifications to the algorithm's overall structure and a New Harmony improvisation step in an effort to enhance the result quality, decrease computational load, and make the technique less sensitive to the HS parameters. In its simplest form, HS constructs a New Harmony vector using memory consideration, random selection, and pitch modification. The IHS technique uses memory consideration and random selection to create a New Harmony vector without pitch modification. The second difference is how the two procedures are done. The New Harmony vector is innovated using a single technique that applies to all its components, unlike the traditional HS, which uses a different procedure for each vector part. As a result, "good" solution vectors fill up the HM memory much more quickly than they do in the standard HS algorithm.

Where n is the total amount of decision variables, the optimization problem is first identified. $G_i^l \le G_i \le G_i^U, i = 1,2, \dots, n$.Defines the possible values for the i^{th}decision variable. In this stage, the HS algorithm parameters are also specified: The solution vectors (HMS) and HMCR are what make up the harmony memory.

$$HM = \begin{bmatrix} G_1^1 & G_2^1 & G_3^1 \dots & G_m^1 \\ G_1^2 & G_2^2 & G_3^2 \dots & G_m^2 \\ \dots & \dots & \dots & \dots \\ G_1^{HMS} & G_2^{HMS} & G_3^{HMS} & G_m^{HMS} \end{bmatrix} \dots\dots\dots(7)$$

The inspection teams are dispersed at random around the city, and each building block is given to the inspection crew that is geographically closest to it.

New Harmony improvisation

In third step, a New Harmony vector (NHV) is created by a combination of random selection and taking memory into account.

The i^{th}design variable (the location where thei^{th} inspection team will begin its work) is picked at random using a random selection technique with probability 1-HMCR and using a random selection from HM memory with probability HMCR. These two methods for creating a NHV may be used to both discrete and The second difference is how the two procedures are done. The New Harmony vector is innovated using a single technique that applies to all its components, unlike the traditional HS, which uses a different procedure for each vector part. optimization issues. Thus, the following is a description of how the New Harmony vector production is done for discrete optimization problems:

$$G_j^{New} = \begin{cases} G_j \in \left[G_j^F, G_j^W\right] with probability 1 - HMCR \\ G_j \in HM = \left\{G_j^1, G_j^2, \dots., G_j^{ZNG}\right\} with probability HMCR \end{cases} \dots\dots\dots\dots\dots(8)$$

Where each new solution Vector s is defined according to either the first or the second functions of Eq (8).

3.2.2.2 Harmony memory update

Harmony vector is superior to the worst harmony vector of the HM in terms of the value of the goal function, the worst harmony is replaced by the New Harmony vector. Once all possible iterations have been exhausted, the solution with the highest value of the objective function is selected, and the pathing problem

$$L(b) = \sum_{j=1}^{M-1}\left(t_{b(j),b(j+1)}\right) + t_{b(M),b(1)} \ (9)$$

$$min\left[\sum_{r=1}^{m_{GP}^{(j)}-1} t(GP_r, GP_{r+1}) + t\left(GP_{m_{GP}^{(j)}}, GP_1\right)\right], j = 1, \dots, M_{JS} \dots\dots\dots\dots (10)$$

The path problem of equations (9) and (10) is formed and solved for each district based on the best solution obtained once all iterations are complete and the objective function value is known. Figure 3: The fundamentals of the IHS methodology.

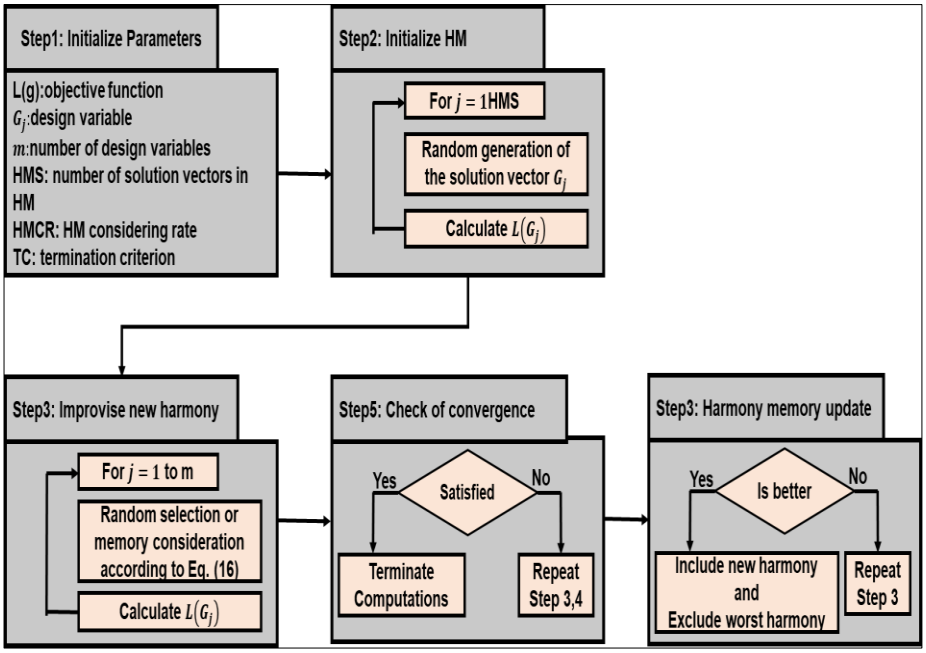

Figure 3 Flowchart of the improved harmony search algorithm

3.2.3. Fuzzy Logic

In a real-world driving system, motion instructions like the steering angle and the accelerator are individually regulated, but the outgoing commands are related with one another since an action may be chosen in the condition PC-DQN. Applying motion instructions directly in autonomous driving will result in poor precision and erratic behavior. Multiple control parameters are often returned by fuzzy control systems. When used in control systems, defuzzification of the fuzzy outputs solves the problem of many association instructions in PC-DQN. Based on the principles of fuzzy control, we present a fuzzy logic approach in PC-DQN and include a defuzzification technique into our methodology.

The steering angle and the accelerator are considered as motion instructions in our motion planning technique. For the steering angle, five fuzzy variables are created: $E^g = \{e^g_{MP}, e^g_{MG}, e^g_{HA}, e^g_{BG}, e^g_{BP}\}$, where e^g_{HA} denotes a straight line. Turning to the right by a lesser margin is denoted by e^g_{BG}, whereas a wider margin is indicated by e^g_{BP}. For a turn to the left, a lower e^g_{MG} value indicates a more subtle turn than a bigger e^g_{MP} value. Also, for the accelerator, we have five discrete fuzzy variables represented by the equation $E^e = \{e^e_{MP}, e^e_{MG}, e^e_{HA}, e^e_{BG}, e^e_{BP}\}$, where e^e_{HA} stands for no acceleration and no deceleration. Acceleration of a lesser magnitude, denoted by e^e_{BG}, and a bigger magnitude, denoted by ee^e_{BP}. A lesser rate of slowing is indicated by e^e_{MG}, whereas a bigger rate of slowing is indicated by e^e_{MP}.

At each time step, the output layer of the Q-network is used in existing DQN-based systems for autonomous driving to generate an action that couples the steering angle and the accelerator. We use two fuzzy sets since the steering angle and the accelerator are both output motion instructions. We refer to the two halves of the action representation of motion instructions, E^g and E^e, as fuzzy sets.

Let $W^g = \{w(e^g_{MP}), w(e^g_{MG}), w(e^g_{HA}), w(e^g_{BG}), w(e^g_{BP})\}$ and $W^e = \{w(e^e_{MP}), w(e^e_{MG}), w(e^e_{HA}), w(e^e_{BG}), w(e^e_{BP})\}$ be the values returned by the units that make up the accelerator and the steering angle, respectively.

Since the output layer employs the SoftMax activation function, the values in W^g and W^ealso span the same 0–1 interval as the probability of the motion instructions in E^g and E^e. From a fuzzy logic perspective, the degrees of membership of motion commands may be thought of as belonging to the probability sets W^g and W^e. Since the core of motion planning for autonomous driving is a kind of automated control problem, maximum defuzzification is a well-known and widely applied technique in the area of automatic control. Furthermore, because all of the FC layer's neural nodes contribute

equally, we utilize the mean value of the five fuzzy variables outputted by the FC layer's neural nodes as the numerical values of the steering angle and accelerator at a time step d, which can be written as (3).

$$\begin{cases} e_d^g = \frac{1}{M_g}\sum_{j=1}^{M_g} e_j^g, e_j^g = \underset{e^g \in E^g}{max} w(e^g) \\ e_d^e = \frac{1}{M_e}\sum_{j=1}^{M_e} e_j^e, e_j^e = \underset{e^e \in E^e}{max} w(e^e) \end{cases} \quad \ldots\ldots\ldots (11)$$

where the ultimate values of the steering angle and the accelerator at time step d, are e_d^g and e_d^e, respectively. The degree function W(.), which is also the likelihood of the associated motion instructions, is a measure of how closely a group of people are connected. Maximum degrees of membership for the actions of steering angle M_g and accelerator M_e are denoted by their corresponding numeric values.

4. Result and discussion

This section provides a description of our experiments and a report on the performance of our technique in the CARLA, which is an open simulator for autonomous driving that helps with the creation, training, and validation of autonomous urban driving systems. CARLA has visuals that are more realistic, urban layouts, a multiplicity of vehicle models, buildings, people, street signs, and other elements, which makes it a better option for testing directional planning for autonomous driving.

4.1. Training Setup and Data Analysis

The CARLA models a town with a two-way road that has lanes for both vehicles and pedestrians. Figure 4 depicts a map of the town (a). The two predetermined routes for training are represented by the red curve and the blue line, while the four predefined routes for testing are represented by the green curve, the cyan curve, the orange curve, and the black curve. The routes' starting points and final destinations are shown by the hollow and solid circles, respectively. The image of the driver's perspective acquired by the front RGB camera is shown in Figure 4(b). Therefore, we do not include elements like as traffic signals, speed restrictions, or impediments like other vehicles or pedestrians in the scenarios in order to focus on testing the performance of turning as directional planning and lane following. To do this, we installed an RGB camera facing the driver to collect data about the surrounding area. The simulator's original image was taken at 800 pixels by 600 pixels in size.

Figure 4 CARLA simulator. (a) Map with defined routes (b) an image of the driver view captured by the front facing RGB camera [26]

We record at a frame rate of 5 frames per second (fps) throughout both the training and testing phases to minimize data redundancy. As shown in Fig.7(a), we create two routes for training that comprise a variety of road types such as straight line and junctions so that the model may learn to drive in a straight line, turn left, and turn right. Both paths add up to a total of 402 m, however the shorter one is just 214 m long. Both of these routes need drivers use a variety of maneuvers, such as a straight line, left turn, and right turn, to reach their destinations. Prior to the actual training, the routes are manually segmented into individual, consecutive points. Our trials include millions of iterations of training to teach the model the best strategy for motion planning. Each round, the vehicle randomly chooses a global course from the two

predetermined routes and travels until it crashes, finishes the route, or drives off the road (over 80% overlap with the sidewalk).

The vehicle motion choice is randomly selected from the first 200,000 allowable actions during training. From step 200,000 to 1,000,000, the vehicle motion choice is randomly determined with a probability or created using the suggested PC-DQN. In this phase, probability linearly declines from 0.99 to 0.05 and stays at 0.05. The suggested strategy is tested using four more routes (R3, R4, R5, and R6), indicated by the green, cyan, orange, and black curves, respectively. To verify the effectiveness of the proposed method, we compare our proposed method with three existing methods. Existing methods such as SVM [23], CNN[24], and LSTM[25]. The proposed system's accuracy (A) is defined as the proportion of the total number of correctly planned path. Figure 5 and Table 1 depict the proposed method's prediction accuracy on various paths planning technique. From the figure 5, it is clear that, the proposed method has a higher accuracy when compared to conventional methods.

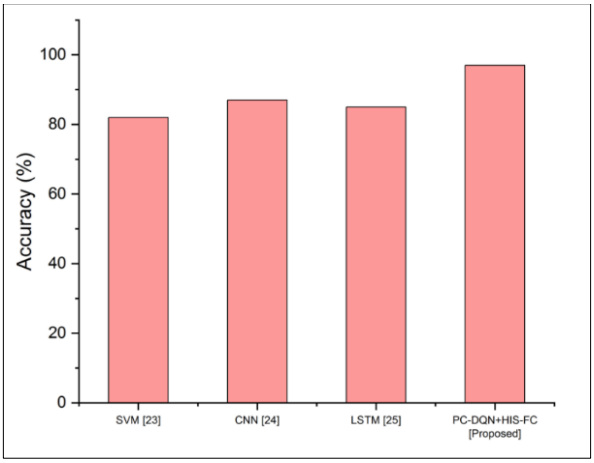

Figure 5 Comparison of accuracy

Table 1 Accuracy computation analysis

Methods	**Accuracy (%)**
SVM [23]	82
CNN [24]	87
LSTM [25]	85
PC-DQN+HIS-FC [Proposed]	97

The term "successful episode" refers to how well an autonomous vehicle's planning process went in getting it to its destination without any complications. Figure 6 and table 2 depicts the comparison of the proposed method's successful episode with conventional methods. Figure 6 shows that the proposed method has a higher successful episode than conventional methods.

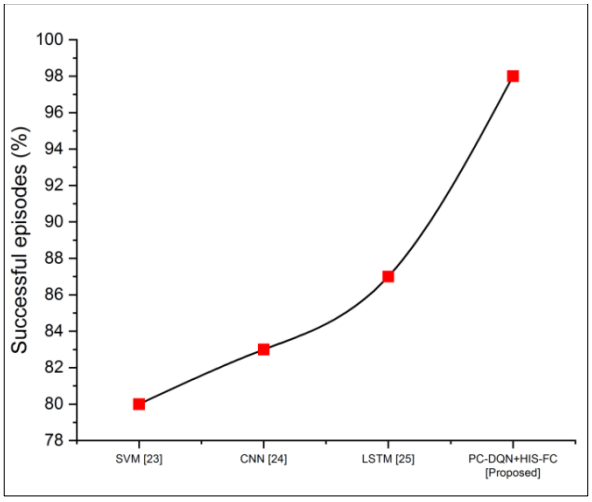

Figure 6 Comparison of successful episodes

Table 2 Successful episodes computation analysis

Methods	Successful episodes (%)
SVM [23]	80
CNN [24]	83
LSTM [25]	87
PC-DQN+HIS-FC [Proposed]	98

The difficulties an autonomous vehicle encounters on the way to its destination are referred to as "collision episodes." The collision episode between the suggested approach and traditional methods is shown in Figure 7 and table 3. The proposed method has a much lower collision episode than traditional methods, as seen in the figure 7.

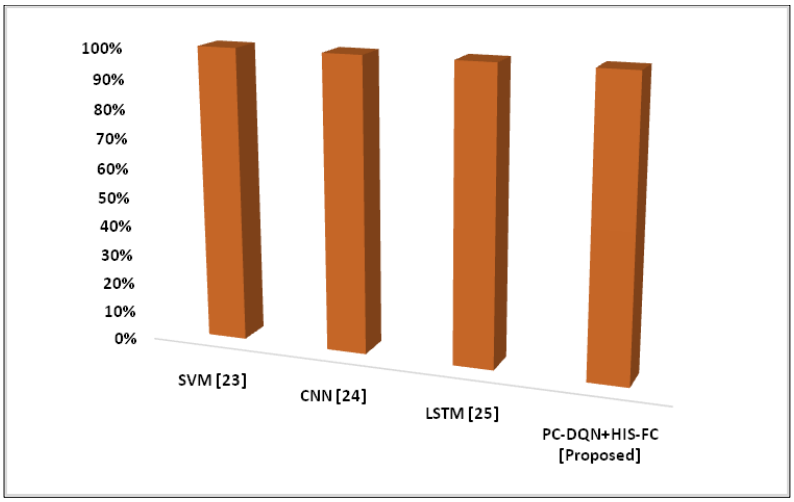

Figure 7 Comparison of collision episodes

Table 3 Collision episodes computation analysis

Methods	Collision episodes (%)
SVM [23]	25
CNN [24]	27
LSTM [25]	30
PC-DQN+HIS-FC [Proposed]	9

To describe a sequential decision-making process, the learning efficiency method is utilized to define the motion planning issue for autonomous driving. Figure 8 and table 4 illustrate the comparative effectiveness of the suggested method and the more traditional approaches of learning efficiency. As can be seen from the figure 8, the learning efficiency of the suggested method is significantly higher than that of more traditional methods.

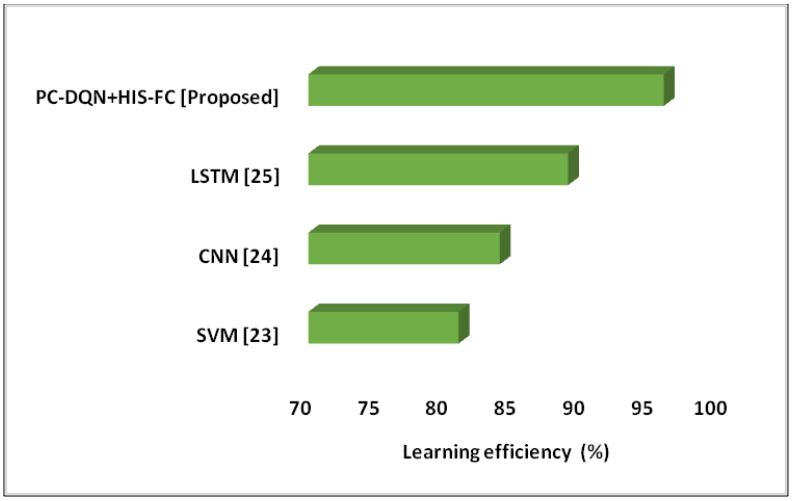

Figure 8 Comparison of learning efficiency

Table 4 Learning efficiency computation analysis

Methods	Learning efficiency (%)
SVM [23]	81
CNN [24]	84
LSTM [25]	89
PC-DQN+HIS-FC [Proposed]	96

An autonomous vehicle's execution time is the amount of time it needs to carry out its plan to determine the other vehicle's trajectory or course and take appropriate action to prevent a collision. Figure 9 and table 5 illustrate the comparative effectiveness of the suggested method and the more traditional approaches of execution time. As can be seen from the figure 9, the execution time of the suggested method is lower than that of more traditional methods.

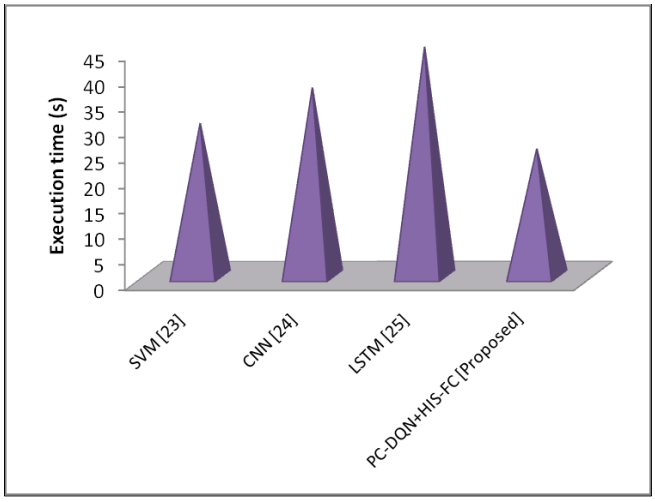

Figure 9 Comparison of prediction time

Table 4 Prediction time computation analysis

Methods	Execution time (s)
SVM [23]	30
CNN [24]	37
LSTM [25]	45
PC-DQN+HIS-FC [Proposed]	25

The suggested approach is compared to certain existing techniques, including SVM [23], CNN [24], and LSTM [25], in Figures 4 to 8. The recommended technique outperforms the already used methods in terms of performance due to the shortcomings of the latter. The strategies now in use have the following drawbacks. When a dataset overlaps, the SVM method performs poorly, CNN fails to capture object orientation and position, and LSTM is not well suited for learning tasks like route prediction when the input data is not a sequence.

5. Conclusion

To address the problems of both continuous and discrete outputs in DQNs, this research proposes a unique Provisional Cross-layered Deep Q-Network for use in fully autonomous driving systems.

To further address the problem of independence among various motion commands, we suggest an HIS-FC approach for the proposed PC-DQN. The suggested PC-DQN with HIS-FC outperforms state-of-the-art approaches in making direction plans in accordance with a specified global route. The suggested strategy also outperforms other methods in terms of learning and driving stability. Since the suggested technique does not consider road impediments like vehicles and pedestrians, we want to address the problem of obstacle avoidance and include traffic signals into our model in our future work.

Compliance with ethical standards

Disclosure of conflict of interest

No conflict of interest to be disclosed.

References

[1] D. Feng et al., Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges, IEEE Trans. Intell. Transp. Syst., 2020, vol. 22, no. 3, pp. 1341-1360.

[2] M. Martínez-Díaz and F. Soriguera, Autonomous vehicles: Theoretical and practical challenges, Transp. Res. Procedia, 2018, vol. 33, pp. 275-282.

[3] N. Adnan and S. M. Nordin, bin Bahruddin, M.A. and Ali, M, How trust can drive forward the user acceptance to the technology? In-vehicle technology for autonomous vehicle., Transp. Res. A, 2018, vol. 118, pp. 819-836.

[4] Y. Liu et al., A novel lane change decision-making model of autonomous vehicle based on support vector machine, IEEE Access, 2019, vol. 7, pp. 26543-26550.

[5] X. Wu et al., The autonomous navigation and obstacle avoidance for USVs with ANOA deep reinforcement learning method, Knowl. Based Syst., 2020, vol. 196, p. 105201.

[6] Ravankar et al., Path smoothing techniques in robot navigation: State-of-the-art, current and future challenges, Sensors (Basel), 2018, vol. 18, no. 9, p. 3170.

[7] S. B. Atitallah et al., Leveraging Deep Learning and IoT big data analytics to support the smart cities development: Review and future directions, Comput. Sci. Rev., 2020, vol. 38, p. 100303.

[8] X. Hu et al., Dynamic path planning for autonomous driving on various roads with avoidance of static and moving obstacles, Mech. Syst. Signal Process., 2018, vol. 100, pp. 482-500.

[9] P. Wang et al., Obstacle avoidance path planning design for autonomous driving vehicles based on an improved artificial potential field algorithm, Energies, 2019, vol. 12, no. 12, p. 2342.

[10] L. Chen et al., Parallel planning: A new motion planning framework for autonomous driving, IEEE CAA J. Autom. Sin., 2018, vol. 6, no. 1, pp. 236-246,

[11] P. C. Song et al., A parallel compact cuckoo search algorithm for three-dimensional path planning, Appl. Soft Comput., 2020, vol. 94, p. 106443,

[12] M. Panda et al., A comprehensive review of path planning algorithms for autonomous underwater vehicles, Int. J. Autom. Comput., 2020 , vol. 17, no. 3, pp. 321-352.

[13] T. M. Cabreira et al., Survey on coverage path planning with unmanned aerial vehicles, Drones, 2019, vol. 3, no. 1, p. 4.

[14] C. Qu et al., A novel reinforcement learning based grey wolf optimizer algorithm for unmanned aerial vehicles (UAVs) path planning, Appl. Soft Comput., 2020, vol. 89, p. 106099.

[15] K. Chu et al., Real-time path planning of autonomous vehicles for unstructured road navigation, Int. J. Automot. Technol., 2015, vol. 16, no. 4, pp. 653-668.

[16] J. Song et al., Path planning for multi-vehicle-assisted multi-UAVs in mobile crowdsensing, Wirel. Commun. Mob. Comput., 2022, vol. 2022, 1-21.

[17] C. Zhou et al., The review unmanned surface vehicle path planning: Based on multi-modality constraint, Ocean Eng., 2020, vol. 200, p. 107043.

[18] C. Huang et al., A new dynamic path planning approach for unmanned aerial vehicles, Complexity, 2018, vol. 2018, 1-17.

[19] M. Soliman et al., Path planning control for 3-omni fighting robot using PID and fuzzy logic controller in The International Conference on Advanced Machine Learning Technologies and Applications (AMLTA2019) 4. Springer International Publishing, 2020, pp. 442-452

[20] M. Dirik et al., Visual-servoing based global path planning using interval type-2 fuzzy logic control, Axioms, 2019, vol. 8, no. 2, p. 58.

[21] M. N. Zafar and J. C. Mohanta, Methodology for path planning and optimization of mobile robots: A review, Procedia Comput. Sci., 2018, vol. 133, pp. 141-152.

[22] San Juan, V. Santos, M. and Andújar, Intelligent UAV map generation and discrete path planning for search and rescue operations. J.M., Complexity, 2018, vol. 2018,

[23] M. Morsali et al., Spatio-temporal planning in multi-vehicle scenarios for autonomous vehicle using support vector machines, IEEE Trans. Intell. Veh., 2020, vol. 6, no. 4, pp. 611-621.

[24] R. Shi et al., CNN-Transformer for visual-tactile fusion applied in road recognition of autonomous vehicles, Pattern Recognit. Lett., 2022.

[25] H. Wang et al., Risk assessment and mitigation in local path planning for autonomous vehicles with LSTM based predictive model,"IEEE Trans. Autom. Sci. Eng., 2021, vol. 19, no. 4, pp. 2738-2749

[26] Dosovitskiy et al., CARLA: An open urban driving simulator in Conference on robot learning, 2017, Oct., pp. 1-16. PMLR.