Development and Key Technologies of the Planning and Control System of Autonomous Driving

Yuyang Zhang[a], Junliang Feng[b], Mingni Huang[c], Qinghao Liu[d], Wenxuan Fu[e]

Chang'an Dublin International College of Transportation, Chang'an University, Xi'an, China

[a]China15655230306@163.com, [b]junliang.feng@outlook.com, [c]mingni.huang@ucdconnect.ie, [d]qh_liu18@163.com, [e]1481343302@qq.com

Abstract. As autonomous driving technology gradually matures, it may become the most crucial driver assistance system in the near future. As an important part of autonomous driving technology, decision-making, path planning, and control technologies have become mainstream research trends. This paper focuses on the development and key technologies of autonomous driving planning and control system, based on published literature. The architecture of an autonomous driving planning and control system is usually divided into two subsystems: decision-making and planning system and control system. Our study provides a detailed description of the different methods and interconnections of these two subsystems. We then conduct a comparative analysis and discussion to review the advantages and limitations of the various schemes. Overall, we hope that this study provides timely and relevant information in a rapidly evolving field concerning planning and control technology for autonomous vehicles.

Keywords: Autonomous Vehicles, Decision-Making, Planning, Control.

1. Introduction

Automatic driving system (ADS), also known as autonomous vehicle technology, has made a breakthrough with the development of artificial intelligence technology. To promote autonomous vehicle technology, the Defense Advanced Research Projects Agency (DARPA) held three competitions from 2000 to 2010. The first, called DARPA Grand Challenge, was implemented in the Mojave Desert in the United States in 2004, requiring autonomous vehicles to cross 142 miles of desert roads in 10 hours, but no team made a success. After that, DARPA Challenge [1] was held in 2005, requiring autonomous vehicles to cross tracks with different terrains including tunnels and many sharp turns in a 132-mile route, with four cars in total completing the route within the specified time. Stanley from Stanford University [2] came in first, and Sandstorm and H1ghlander from Carnegie Mellon University ranked second and third respectively. The third competition was the DARPA City Challenge [3] held in 2007, which was the largest and most important in the field of autonomous driving at that time, mainly aimed at the greatest challenge to demand full-scene autonomous driving in complex urban environments. It required autonomous vehicles to travel 60 miles on a track simulating an urban environment conforming to California traffic rules. In this competition, 6 cars completed the route within the specified time, among which Boss [4] from Carnegie Mellon University won first place. Although this competition lacks complicated situations such as pedestrians in real urban scenes, which is much simpler than the challenges faced by daily urban driving, it has attracted wide attention and has been hailed as a milestone in the development of autonomous vehicles.

The advancement of ADS is accompanied by the development of planning and control technologies. These projects are popular, and there have been a large number of reviews on motion planning and control algorithms for autonomous vehicles [5]. Since the three modules of path planning, motion decision, and control are mutually integrated and interdependent, this paper considers them as a unified system called the planning and control system. This comprehensive architecture is generally divided into two modules, as illustrated in Figure 1, namely decision planning subsystem and control subsystem. In this paper, we demonstrate the typical architecture of autonomous vehicle planning and control systems, providing a structured and comprehensive overview of current and commonly used solutions for autonomous driving planning and control. The structure of the rest of this paper is as follows: the second chapter summarizes the related technologies of the decision planning system, which is divided into three parts: global path planning, motion decision, and local path planning. This includes the sub-tasks of path selection at different levels and corresponding automobile behavior selection. Chapter 3 summarizes the technologies related to control systems, including direct hardware drive control and path tracking control. Finally, in Chapter 4, we summarize the current state of the planning and control system for autonomous vehicles and highlight potential future development prospects and directions.

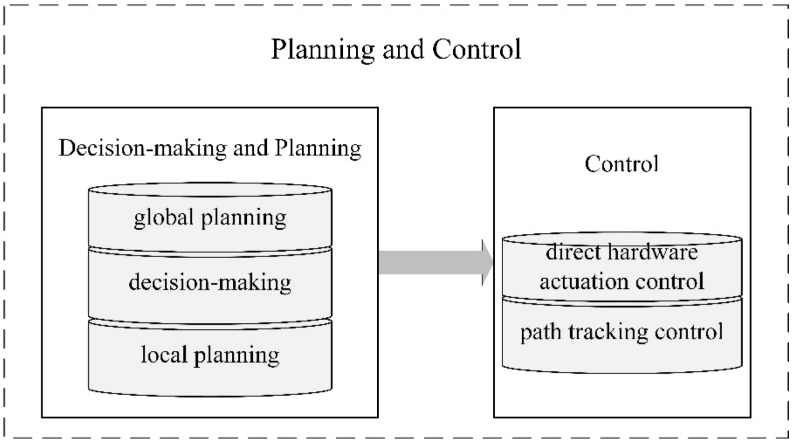

Figure 1. Overview of the typical architecture of the automation system

1.1 Global Path Planning

Global path planning aims to find the shortest route from the starting point to the destination in a large-scale road network. It usually involves determining the path of the entire environment, which is represented in the form of graphics or a grid. This process is typically carried out offline or at a low update rate. Compared to local path planning, the understanding of simultaneous environments is limited. This part will introduce the global planning system based on three effective and commonly used algorithms.

1.1.1 Dijkstra Algorithm

The Dijkstra algorithm is a well-known graph search algorithm that has served as the theoretical foundation for many systems, including the later-introduced A* algorithm. It is primarily used to find the shortest path between the source node and all other nodes in a weighted directed graph with non-negative edge weights. Rosyidi et al. [6]proposed a Dijkstra algorithm based on simultaneous dynamic weights to improve the implementation of global path planning in large cities. The author proposes a dynamic traffic factor based on a time function, which can assist in calculating the shortest distance and the smallest fuel consumption using the Dijkstra algorithm, taking into account real-time traffic conditions. According to simulation results, the improved Dijkstra algorithm has a shorter search time and is more fuel cost efficient, especially in large cities with complex traffic conditions. On the other hand, since finding the shortest path in an uncertain environment is a significant challenge in global path planning, a suitable approach is to utilize fuzzy numbers and fuzzy set theory. However, [7]the classical Dijkstra algorithm performs much worse in fuzzy number optimization compared to clear numbers. This indicates that the Dijkstra algorithm requires improvement when applied in a fuzzy environment. To address this issue, Deng et al. [8]extended the conventional representation by incorporating triangular fuzzy number operations into the classical Dijkstra algorithm. This modification aims to solve the problem of finding the shortest path when dealing with blurred arc lengths. The new method utilizes a hierarchical mean integral representation based on fuzzy numbers, specifically under the condition of fuzzy arc length. It directly calculates the sum and order of fuzzy numbers, thereby significantly enhancing the efficiency and accuracy of processing fuzzy set data. In addition, the Dijkstra algorithm can not only calculate geometric distance, but also handle fuzzy variables such as time and weather. This feature has great potential for application in transportation systems. With the development of new algorithms, they can be applied to optimize the global path modeling and execution efficiency of the Dijkstra algorithm.

1.1.2 A* Algorithm

The A* algorithm is a classical method for finding the shortest path between two points in a graph. The traditional A* algorithm is actually a heuristic implementation of the Dijkstra algorithm, which is essentially a best-first search algorithm. The A* algorithm [9] applied in autonomous driving path planning is a standard goal-oriented path planning algorithm. Sariff et al. [10] summarize the principles of the traditional A* algorithm, which states that the A* algorithm combines the uniform cost of using a fitness function with the advantages of a greedy search. The A* algorithm, in addition to its ability to judge the existence of a free path, can also moderately estimate the actual cost of reaching the target distance, resulting in a high-quality search. The A* algorithm, however, has the disadvantage of running slowly due to the potential lack of heuristic information prior to reaching the goal. This can result in the generation of a large number of nodes, ultimately slowing down the search speed. To solve this problem, Warren [11] proposed an improved A* algorithm based on vector search, which combined the advantages of the high accuracy of the traditional A* algorithm and the advantages of the high-speed vector method across cluttered space. It is proven that using the new method to search on a three-dimensional road network with chaotic obstacles can greatly improve efficiency, usually by three to four orders of magnitude. However, for fine grid search, the resolution of obstacle mapping in the new method may decrease, which will lead to the decline of optimal path length. To optimize the smoothness of the search path and the obstacle avoidance cost of the A* algorithm, Montemerlo et al. [12] proposed a hybrid A* algorithm based on a four-dimensional discrete grid. Figure 2 shows a graphical comparison of three search algorithms. A* is on the left, and it only accesses the state corresponding to the center of the grid cell. The middle graph represents the Field D* algorithm. The right graph shows hybrid A*, which is capable of associating a continuous state with each grid in order to create a smooth path. The hybrid A* algorithm has been proven to achieve high path smoothness and planning speed in environments with numerous corners and obstacles. For the problem of incomplete forward strategy of polygonal moving objects, Fan et al [13] proposed an improved A* algorithm based on depth-first search. The improved A* algorithm incorporates the h' function of the depth-first search setting to enhance the search for the optimal path. It also utilizes object boundary convex polygons to approximate the shape of irregularly shaped objects. Experiments show that enhancing the A* algorithm not only takes into account the shortest path that objects need to rotate, but also significantly reduces the search time compared to the traditional exhaustive depth-first search.

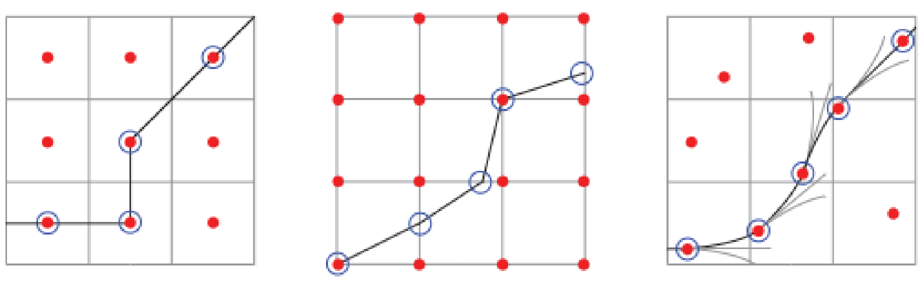

Figure 2. Graphical comparison of search algorithms

Left: A* Algorithm Middle: Field D * (Ferguson & Stentz, 2005) Right: Hybrid A* Algorithm [14]

1.1.3 Algorithm Based on Hierarchical Technology

Hierarchical technology primarily utilizes the hierarchical road structure. Generally speaking, roads can be divided into different levels such as expressways, main roads, and local roads successively. As for route planning, the importance of hierarchy increases as the distance between the starting point and destination increases. Additionally, the speed of expressways is generally much faster than that of regular roads in most areas. In other words, global long-distance planning must take into account the integration of multi-level roads in order to calculate the shortest and fastest route. To solve this problem, Geisberger et al. [14] designed a two-way Dijkstra algorithm to add the inherent hierarchy of the road network, which used the shrinking hierarchical structure to calculate the shortest path in the whole road network. Because it only needs to visit hundreds of nodes, the high-speed search can be realized in a large distance space. In practical application, the electronic map database of the road network needed for global path analysis is very large, while the computational capacity of the vehicle-mounted path planning system is limited, which may lead to problems such as long search time and inaccurate results. To solve this problem, Jiang et al. [15] design an improved Dijkstra algorithm based on hierarchical road networks. This algorithm primarily utilizes multi-layer maps and hierarchical search technology to partition the road network into high-resolution and low-resolution maps. Firstly, the route to the destination is initially mapped out on a low-resolution map. Then, the overall terrain is analyzed using a high-resolution map, and detailed planning is conducted for each specific area. Finally, a complete optimal global path is synthesized (see Figure 3). The author demonstrated the reliability and efficiency of the 1:250,000 network map using the new algorithm.

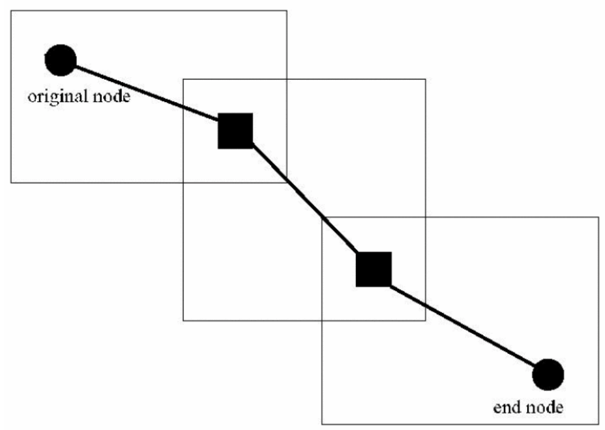

Figure 3. Schematic diagram of hierarchical path search [33]

1.2 Behavioral Decision-Making

Decision-making is to achieve two purposes by selecting vehicle behavior and dynamic constraints, especially speed control. One is to avoid collision between obstacles and moving objects within the decision-making time range. The other is to meet the driving safety, comfort, and high fitting of the shortest path as much as possible. At present, there are two main technologies, that is, technology based on a finite state machine (FSM) and that based on the partially observable Markov decision process (POMDP).

1.2.1 Behavior Decision-Making Technology Based on Finite State Machine (FSM)

In the method based on a finite state machine (FSM), the decision-making based on rules is mainly used to select vehicle actions in different scenarios (Figure 4). In this approach, the behavior is represented by state and the transition is based on rules derived from data transmitted by the environment-aware localization system. Jo et al. [16] developed a behavior decision-making technology based on traffic rules and FSM. The main rules considered in the decision-making of this technology are traffic rules, including road driving specifications, traffic lights, and keeping the speed within the speed limit. Autonomous vehicles equipped with this technology can achieve accurate and smooth self-driving based on high-precision maps provided in advance. In addition, Guidolini et al. [17] proposed an FSM behavior selection technique considering pedestrian posture and speed. It first accepted pedestrian data collected by lidar and processed by CNN and then established the attitude and speed of the object in the path to guide the vehicle to drive on a safe path. The state machine mainly has two states including free running and stop, which are mainly switched according to pedestrian information and driver's manipulation of crosswalks. According to the experiment results, the new method performs well when autonomous vehicles shuttle from busy crosswalks. However, the method based on FSM usually finds it difficult to analyze the external uncertain information and model the complex road scene.

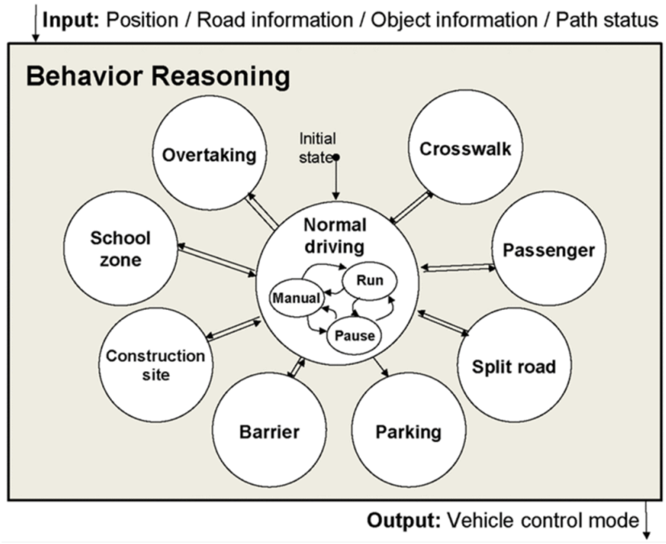

Figure 4. Schematic diagram of decision process of FSM-based behavior selection system [17]

1.2.2 Behavioral Decision-Making Technology Based on Partially Observable Markov Decision Process (POMDP)

The main advantage of the method based on the partially observable Markov decision process (POMDP) is that it can reason and model the uncertainty of action transition between states and the uncertainty of perception. Given the possible defects of sensor information transmission such as sensor noise and physical delay, Ulbrich et al. [18] developed a lane change model based on real-time POMDP. Its POMDP was mainly divided into two stages, namely, calculating and evaluating whether the lane change is possible and beneficial. To simplify the POMDP, a signal processing network considering the relative distance, relative velocity, and time of collision with objects around the vehicle is also developed. Galceran et al. [19] proposed an integrated reasoning and decision closed-loop system based on POMDP, which could be staggered according to behavior prediction and strategy selection. The system first used the method of Bayesian change point detection to predict the surrounding vehicles' future actions by using their action history. Then, a decision algorithm approximate to POMDP was used to evaluate various possibilities through forward simulation. Finally, the strategy with the maximum reward value was implemented. This behavioral decision-making system combines prediction and analysis systems, which effectively avoids the shortcoming that the traditional POMDP-based decision-making algorithm needs to collect abundant training data, and ensures that the provided strategy is not over-conservative but effective.

1.3 Local Motion Planning

Local motion planning aims to align the actual motion trajectory of the vehicle with the optimal path, while ensuring that the kinematic constraints of the vehicle, driving safety, and comfort are met as much as possible. Unlike global path planning, local motion planning needs to consider additional constraints, such as lanes and surrounding obstacles. The methods of local motion planning can be divided into three categories: the graph search-based method, the sampling-based method, and the interpolation curve-based method.

1.3.1 Method Based on Graph Search

Trajectory planning technology based on graph search can specify the evolution of a vehicle's state over time. The state lattice and A* algorithm are the two most common trajectory planning technologies. The technology based on state lattice can discretize the decision space. One of its advantages is its strong ability to handle high-dimensional and high-latitude information, making it suitable for dynamic environments. However, its computational cost is high, and the discrete output path may limit the planning ability. McNaughton et al. [20] proposed a state lattice planner based on a spatiotemporal search graph. This approach effectively addresses the limitation of the traditional state lattice, which fails to accurately represent real-world driving scenarios. By adapting the state lattice to the structured environment, the proposed planner provides a more accurate depiction of the road driving scene. The spatiotemporal search technology can assign state vectors to each node and achieve a more precise discretization with a wider range of motion strategies. The A* algorithm is a heuristic implementation of Dijkstra's algorithm. It typically prioritizes searching for the path from the starting point to the endpoint, but it does not take into account vehicle motion. Fassbender et al. [21] introduced two new node expansion methods based on the classical A* algorithm to obtain more accurate and smoother path curves. It is proven that both methods are effective, which is expected to extend their performance in unstructured environments under unreliable lane network data in future research.

1.3.2 Sampling-Based Method

Sampling-based technology randomly selects nodes in space to sample to build data connectivity, so as to find the relationship between the current state of the car and the next object state. To solve the poor reliability of RRT, for example, many useless samples appear and unnatural paths are generated, Du et al. [22] proposed an RRT framework based on drivers' visual search behavior for simultaneous motion planning. It mainly uses some characteristics of drivers' visual search behavior on the road to quickly and effectively guide the sampling strategy of the RRT framework, with its complete structure of the DV-RRT framework seen in Figure 5. Karaman et al. [23] developed an improved RRT* algorithm based on the RRT algorithm, which achieves asymptotic optimality that RRT does not have. RRT* algorithm is based on simultaneous motion dynamics planning technology, which still searches the best path of real-time constraints when the algorithm enters the iterative rule stage, thus realizing the online convergence of the plan during execution. It is manifested that its convergence probability to the optimal solution is greatly improved compared with the classical RRT and the trajectory quality is also promoted.

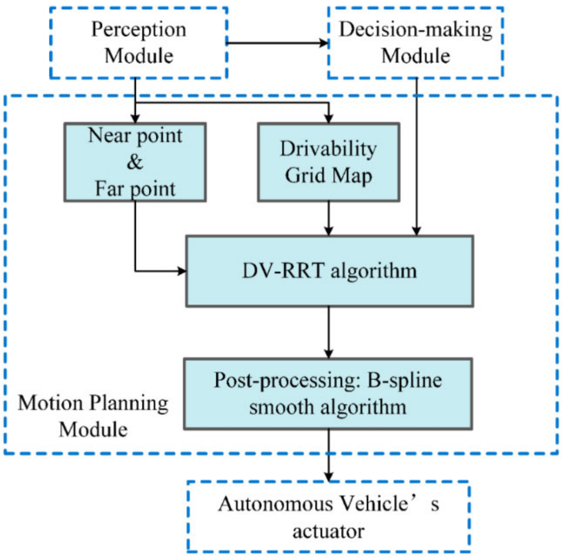

Figure 5. Complete structure of DV-RRT local motion planner [23]

1.3.3 Method Based on Interpolation Curve

Based on the interpolation curve technology, the paths generated by the previous global planner are interpolated. A new set of collision-free paths is inserted, and these new paths are fitted with the original paths to create a smoother trajectory. Alia et al. [24] proposed a method called Tentacles, which is based on a cyclotron curve for local trajectory planning and tracking. This method imitates the behavior of insects by using Tentacles to detect and avoid obstacles. The Tentacles occupy grids with special geometric shapes that simulate the surrounding environment and possible trajectories of vehicles. Compared to traditional circular tentacles, this new method focuses on the influence of vehicle steering angle by utilizing a cyclotron curve. This allows for the selection of the best tentacles from a wide range of navigation options. Mouhagir et al. [25] proposed a method for interpolating cyclotron curves based on POMDP. Under the trajectory planning of gyratory tentacles to avoid static obstacles, we construct modified occupancy grids that take into account horizontal and vertical safe distances. These modified occupancy grids, combined with POMDP, are used to select the best tentacles. Its main advantage is that it focuses on the dynamic obstacles and characteristics of both transverse and longitudinal vehicles.

2. Control

The control system primarily accepts and adjusts the planning trajectory generated by the motion planning system. Then, the instructions for controlling the specific speed change and steering angle of the autonomous vehicle are transmitted to the execution hardware, including the steering wheel, throttle, brake pedal, and other actuators. The goal is to generate commands that enable vehicles to follow the optimal planning path as closely as possible in real-world scenarios. With various control methods available for different levels and purposes, this chapter categorizes control methods into two types: direct hardware drive control and path tracking control method.

2.1 Direct Hardware Drive Control

The method of direct hardware drive control involves calculating the physical input to the vehicle components directly based on the optimal path calculated by the motion planner. Generally, after accepting the global and local optimal paths, this method can directly calculate the torque input of the automobile's steering system, throttle, or brake actuator. Meanwhile, it can reduce inaccuracies in automobile speed and steering wheel angle, which are primarily affected by mechanical factors such as feedback and other adjustments. This method mainly includes feedback control and fuzzy control.

2.1.1 PID Control

Feedback control is one of the common methods used for direct hardware drive control in autonomous vehicles. It is capable of estimating and correcting bias by considering factors such as the distance of trajectory deviation, the speed of the vehicle, and the angle of the steering wheel. Funke et al. [26] proposed a framework for feedback control based on the combination of sliding modes and transverse and longitudinal directions. In this framework, the longitudinal feedback controller calculates the transverse and longitudinal slip separately based on the sensor data. It then controls the tire peak force and braking momentum by utilizing the normalized transverse and longitudinal slip unit circle. The author applied this method in the hardware control system of the Audi TTS and successfully achieved efficient hard real-time control at a frequency of 200 Hz. Ziegler et al. [27] proposed a feedback control method that combines feedforward control and offset compensation (Figure 6). This method first converts the yaw rate and steering angle into the actuator input using the static inverse monorail model. It then corrects the input in real time through offset compensation. The desired vehicle steering angle and speed were subsequently calculated based on the vehicle's desired yaw rate and the curvature of the trajectory. The experiment proves that, due to the emphasis on offset compensation, this method is easier to achieve steady-state accuracy based on the deviation of the steering angle. It also has higher accuracy and a wider operation range.

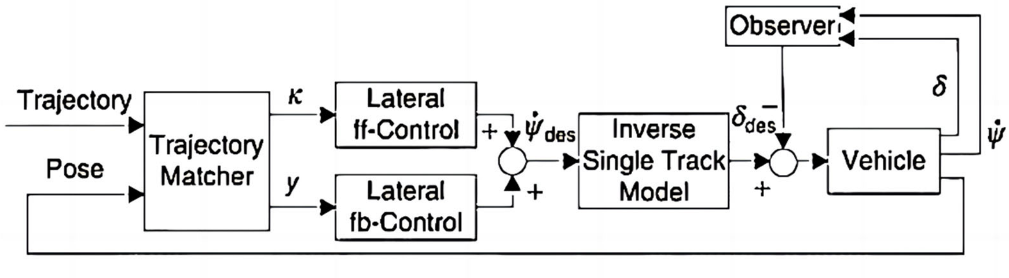

Figure 6. Complete controller loop combining feed-forward and feedback control

2.1.2 Fuzzy Control

Fuzzy control is a control based on fuzzy logic, which is a mathematical reasoning method that tolerates imprecision and uncertainty. As one of the direct hardware drive control methods, it can mainly deal with the uncertain and complex environment when the vehicle is running, and tackle the complex, uncertain, and subjective information of vehicle operation. Naranjo et al. [28] based their fuzzy control method on the combination of artificial vision and transverse and longitudinal directions. The method is mainly divided into two fuzzy logic based controllers for steering (lateral) control and speed (longitudinal) control to drive the vehicle. The fuzzy logic of this method consists of two fuzzy rules while the inputs mainly consist of GPS data and instantaneous values of digital environment map. The experimental results show that this method produces results close to real human driving situations while ensuring the simplicity of the fuzzy rules, and is especially capable of acting quickly in case of trajectory deviations. Fuzzy control is well implemented for the majority of road types controlled by automated driving, but performs poorly at roundabouts, especially roundabout driving, due to the fact that fuzzy control performs poorly in terms of minor lateral control with shorter reaction times. To address this problem, Rastelli et al. [29] designed a fuzzy logic steering control system based on a cascade architecture with parametric trajectory generation. The system first sets predefined points and generates vehicle trajectories around the roundabout by combining geometrical features, followed by outputting fuzzy data through a control system containing two controllers, and finally de-fuzzying to obtain a clear value after optimising the parameter's affiliation function through several iterations. The system is experimentally proven to be robust and stable. Future research on this system may focus on situations such as overtaking, obstacle avoidance and cooperative manoeuvring at roundabouts.

2.2 Path Tracking Control

The path tracking method is to minimize the lateral error between the current position of the vehicle and the reference trajectory. Meanwhile, it is to ensure the correct direction (heading angle) of the vehicle. It features to stabilize the execution of the motion plan calculated by the motion planner subsystem, so as to reduce the inaccuracy mainly caused by the automobile motion model. This control method usually requires the perception data of the sensor and compares these actual values with the expected ones from the waypoint. Based on this comparison, it generates control commands for the throttle, brake, and steering of the vehicle, which mainly includes pure path tracking methods and model predictive control.

2.2.1 Pure Path Tracking Method

The pure path tracking method usually finds a point in the path with a certain forward-looking distance from the current path, and then rotates the front wheel to make the wheel trajectory approach an ideal circular arc geometry. It is usually simple to implement, but the diversity of adjustment variables is low. Paden et al. [30] proposed a pure path tracking method based on a feedback controller and differential equation. Although this technology is easy to implement with good control effect, its stability depends on the smoothness of the reference path, and the extremely uneven reference trajectory may cause excessive computational burden and unstable output results. Thrun et al [31] designed a pure path tracking method based on the collaborative work of multiple control modules.

Its main innovation lies in the speed control system based on four control modules: path planner, sports health monitor, speed recommendation device, and low level speed controller. Most of the parameters of the four controllers are learned from human driving experience, so that the output result is more in line with human driving needs. Experiments have proved that this multi-controller coordinated pure path tracking method can be more flexible in the process of speed transformation, and the reaction speed of acceleration and deceleration is much faster than that of human drivers, while ensuring that the recovery speed is linear and the driving comfort is better

2.2.2 Model Predictive Control (MPC)

Model Predictive Control (MPC), as an advanced control strategy, is essentially an optimization-based method. MPC generally allows the vehicle to predict the future vehicle state so that the motion follows a predetermined path while considering constraints. In most cases, MPC can effectively deal with multi-objective optimization problems. However, due to its non-linearity and delay, the traditional MPC algorithm in the vehicle steering control will produce a certain deviation from the predetermined trajectory. To solve this problem, an N-MPC steering control system based on a neural network was proposed by Guidolini et al. [32] Based on the complex response dynamics of the test vehicle to ambient stimuli, the system uses a neural network N-SPM for modeling, which has been able to respond quickly and simultaneously with 5.0×10^{-6} seconds as an average response time by selecting parameters through genetic algorithm and training continuously on data sets. Then N-SPM is incorporated into the N-MPC controller and three cycles are formed, including the control cycle, optimization cycle, and N-SMP prediction cycle. According to experiment results, this method can fully simulate the complexity of the steering device, greatly reduce steering delay, and achieve faster stable speed. An MPC control model that allows for flexible adjustment of control parameters to accommodate various driving styles has been proposed by Koga et al. The model first divided the vehicle dynamics into lateral MPC and longitudinal MPC separation, based on their different performance in the transverse and longitudinal directions. Horizontal MPC is primarily established by considering yaw rate and angle error. Each series of adjustable parameters holds specific physical significance, and the combination of these seven parameters can exhibit various driving styles and characteristics. On the other hand, longitudinal MPC is relatively straightforward, as it is a first-order linear point mass model that can be substituted with a simple PID control. Experiments have shown that human drivers exhibit a variety of driving control styles, with some drivers prioritizing steering stability while others may prioritize tracking accuracy. Through the adjustment of seven different parameters, the control system successfully achieves stable path tracking while also considering the driver's preferred driving style.

3. Conclusion

This paper considers the two modules of autonomous driving, namely planning system and control system. It provides a systematic and comprehensive review of this crucial technology in autonomous driving. Taking autonomous vehicle planning as the main focus, this paper summarizes the fundamental operational concepts and the most recent advancements in global path planning, local path planning, and behavioral decision-making. When explaining the control system, this paper classifies the different levels and purposes of the control system, with a specific focus on direct hardware drive control and path tracking control. These studies are critical for understanding the planning and control technology of autonomous vehicles, although they do not provide comprehensive coverage of all technologies.

Another contribution of this paper lies in providing an overview reference to facilitate the further development of technology. Currently, with the continuous advancement of machine learning technologies, especially artificial intelligence, and the emergence of interdisciplinary fields, the further development of current technologies may require the integration of a new generation of intelligent algorithms with various hardware and software systems. It is hoped that by providing an overview of autonomous driving technology, the public can gain a deeper understanding of it.

References

[1] Buehler, Martin, Karl Iagnemma, and Sanjiv Singh, eds. The 2005 DARPA grand challenge: the great robot race. Vol. 36. springer, 2007.

[2] Thrun, Sebastian, et al. "Stanley: The robot that won the DARPA Grand Challenge." Journal of field Robotics 23.9 (2006): 661-692.

[3] Buehler, Martin, Karl Iagnemma, and Sanjiv Singh, eds. The DARPA urban challenge: autonomous vehicles in city traffic. Vol. 56. springer, 2009.

[4] Urmson, Chris, et al. "Autonomous driving in urban environments: Boss and the urban challenge." Journal of field Robotics 25.8 (2008): 425-466.

[5] LaValle, Steven M. Planning algorithms. Cambridge university press, 2006.

[6] Rosyidi, Lukman, et al. "Timebase dynamic weight for Dijkstra Algorithm implementation in route planning software." 2014 international conference on intelligent green building and smart grid (IGBSG). IEEE, 2014.

[7] Liu, Shiang-Tai, and Chiang Kao. "Network flow problems with fuzzy arc lengths." IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 34.1 (2004): 765-769.

[8] Deng, Yong, et al. "Fuzzy Dijkstra algorithm for shortest path problem under uncertain environment." Applied Soft Computing 12.3 (2012): 1231-1237.

[9] Hart, Peter E., Nils J. Nilsson, and Bertram Raphael. "A formal basis for the heuristic determination of minimum cost paths." IEEE transactions on Systems Science and Cybernetics 4.2 (1968): 100-107.

[10] Sariff, N., and Norlida Buniyamin. "An overview of autonomous mobile robot path planning algorithms." 2006 4th student conference on research and development. IEEE, 2006.

[11] Warren, Charles W. "Fast path planning using modified A* method." [1993] Proceedings IEEE International Conference on Robotics and Automation. IEEE, 1993.

[12] Montemerlo, Michael, et al. "Junior: The stanford entry in the urban challenge." Journal of field Robotics 25.9 (2008): 569-597.

[13] Fan, Kuo-Chin, and Po-Chang Lui. "Solving find path problem in mapped environments using modified A* algorithm." IEEE transactions on systems, man, and cybernetics 24.9 (1994): 1390-1396.

[14] Geisberger, Robert, et al. "Exact routing in large road networks using contraction hierarchies." Transportation Science 46.3 (2012): 388-404.

[15] Gangwu, Jiang, Wang Jing, and Chen Bo. "The shortest path algorithm based on hierarchical road network." 2006 6th International Conference on ITS Telecommunications. IEEE, 2006.

[16] Jo, Kichun, et al. "Development of autonomous car—Part II: A case study on the implementation of an autonomous driving system based on distributed architecture." IEEE Transactions on Industrial Electronics 62.8 (2015): 5119-5132.

[17] Guidolini, R â nik, et al. "Handling pededrians in crosswalks using deep neural networks in the IARA autonomous car." 2018 international joint conference on neural networks (ijcnn). IEEE, 2018.

[18] Ulbrich, Simon, and Markus Maurer. "Probabilistic online POMDP decision making for lane changes in fully automated driving." 16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013). IEEE, 2013.

[19] Galceran, Enric, et al. "Multipolicy decision-making for autonomous driving via changepoint-based behavior prediction: Theory and experiment." Autonomous Robots 41 (2017): 1367-1382.

[20] McNaughton, Matthew, et al. "Motion planning for autonomous driving with a conformal spatiotemporal lattice." 2011 IEEE International Conference on Robotics and Automation. IEEE, 2011.

[21] Fassbender, Dennis, Benjamin C. Heinrich, and Hans-Joachim Wuensche. "Motion planning for autonomous vehicles in highly constrained urban environments." 2016 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE, 2016.

[22] Du, Mingbo, et al. "Drivers' visual behavior-guided RRT motion planner for autonomous on-road driving." Sensors 16.1 (2016): 102.

[23] Karaman, Sertac, et al. "Anytime motion planning using the RRT." 2011 IEEE international conference on robotics and automation. IEEE, 2011.

[24] Alia, Chebly, et al. "Local trajectory planning and tracking of autonomous vehicles, using clothoid tentacles method." 2015 IEEE intelligent vehicles symposium (IV). IEEE, 2015.

[25] Mouhagir, Hafida, et al. "Using evidential occupancy grid for vehicle trajectory planning under uncertainty with tentacles." 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC). IEEE, 2017.

[26] Funke, Joseph, et al. "Up to the limits: Autonomous Audi TTS." 2012 IEEE intelligent vehicles symposium. IEEE, 2012.

[27] Ziegler, Julius, et al. "Making bertha drive—an autonomous journey on a historic route." IEEE Intelligent transportation systems magazine 6.2 (2014): 8-20.

[28] Naranjo, José E., et al. "Using fuzzy logic in automated vehicle control." IEEE intelligent systems 22.1 (2007): 36-45.

[29] Rastelli, Joshué Pérez, and Matilde Santos Peñas. "Fuzzy logic steering control of autonomous vehicles inside roundabouts." Applied Soft Computing 35 (2015): 662-669.

[30] Paden, Brian, et al. "A survey of motion planning and control techniques for self-driving urban vehicles." IEEE Transactions on intelligent vehicles 1.1 (2016): 33-55.

[31] Thrun, Sebastian, et al. "Stanley: The robot that won the DARPA Grand Challenge." Journal of field Robotics 23.9 (2006): 661-692.

[32] Guidolini, R â nik, et al. "Neural-based model predictive control for tackling steering delays of autonomous cars." 2017 international joint conference on neural networks (IJCNN). IEEE, 2017.