

Research on Unmanned Driving Trajectory Planning Problem Based on Computational Optimal Control

Chenxi Gao

Wuhan Britain-China School, Wuhan, 430030, China

Abstract. This paper first introduces the background of the research: with the rapid development of technology, autonomous vehicles have gained increasing popularity and significantly improved human living standards. As a result, it has become a growing area of research interest for scientists in the technology field. The unmanned system is the core component of driverless vehicles, and the planning system is a crucial link that connects the previous and subsequent stages of these systems. However, planning a safe and efficient optimal trajectory is challenging due to considerable calculation and practical constraints. Traditional trajectory planning methods have limitations; therefore, a more effective solution is needed. Therefore, it has superiorities beyond other methods and thus has promising application prospects.

Keywords: Computational optimal control; Trajectory planning; Artificial Intelligence; Unmanned systems.

1. Introduction

With the development of artificial intelligence technology, autonomous driving is gaining increasing recognition among the general public. It can reduce casualties in extreme situations because using uncrewed vehicles instead of humans during dangerous tasks can minimize personnel losses in accidents [6,7]. Unmanned systems are divided into three major modules: perception, planning, and control[8-10]. The planning module serves as an intermediate step, with the output of the perception system as input for planning and the planning system as input for the control system, making it an essential part of unmanned driving.

There are two categories of planning systems: path tracking and trajectory tracking. Path tracking establishes a coordinate system on the road surface, while trajectory planning adds variable speed to the research. Since speed directly affects passengers' riding experience and energy consumption, and smooth driving can improve ride comfort and reduce fuel consumption, this paper focuses on the problems of trajectory planning [11, 12].

There are several methods for trajectory planning, primarily categorized into four types: sample-based, graph-based, deep reinforcement learning, and computational optimal control. However, most current trajectory planning techniques cannot meet the requirements of high real-time performance and users' pursuit of optimal trajectories simultaneously, which means that fast and comfortable driving route planning needs to be improved [13].

In contrast, although the computational optimal control technique has limitations in other aspects, it has high real-time performance and can plan the optimal trajectory. Therefore, this article believes that this technique is the best choice for solving trajectory planning problems and will primarily focus on researching trajectory planning problems in light of the rapid technological development. First, the main composition of modules in unmanned systems and the function of trajectory planning will be introduced. Next, this paper will explain the advantages and disadvantages of different methods, introduce some classic cases based on these methods, and analyze the reasons for choosing this optimal control, considering its emphasis on efficiency and speed, which makes it the preferred method in many scientists' minds. Subsequently, this paper provides a detailed introduction to the process based on computational optimal control, outlining its principles, applications, cutting-edge research, and future development directions.

2. Introduction to the Unmanned Driving Trajectory Planning Module

2.1 Function of Trajectory Planning Module

Unmanned systems are divided into three main modules: perception, planning, and control. The perception system is used to perceive the surrounding environment, the planning system is used to plan and change paths in real-time based on the surrounding environment, and the control system is used to manipulate vehicle motion according to the planned path[8-10]. Among them, the planning system conducts a series of algorithm research based on the input from the perception system about drivable and non-drivable areas, then outputs drivable paths. As a result, it plays a bridging role in autonomous driving, thereby becoming the primary research focus of this article.

2.2 Classification of Trajectory Planning Methods

There are four primary methods for trajectory planning in autonomous driving. This chapter introduces the operating principles of each technique, analyzes their advantages and disadvantages, and presents some classic cases that illustrate the application of these methods.

2.2.1 Sample-based method

The technical idea behind this method is based on a qualitative analysis of trial and error to determine whether each step is optimal, rather than relying on algorithms with rigorous mathematical language. Specifically, this method takes the input map as a sample and divides it into numerous equally sized grids, placing the starting and ending points in one grid, respectively [15,16]. In this case, there are eight directions of travel centered around the starting point, which correspond to the eight grids [17]. Similarly, for each subsequent step, eight possible paths can be designed.

The disadvantages of this method are: firstly, it requires a large amount of computation, and the accuracy of the trajectory depends to a large extent on the thickness of the grid division. If the grid division is too coarse, the planned trajectory may appear as a broken line, which does not conform to the actual route of car travel in real life. However, if the grid is divided too finely, the computational workload increases exponentially [9]. Secondly, this method relies on the quality of the sampling. If the sampling fails, it may not be possible to obtain data that reaches the endpoint and may be hindered by obstacles[10]. Thirdly, due to the enormous computational complexity of this method, trajectory planning is likely to take a long time and does not meet the requirements of real-time performance[11].

RRT (Rapidly-Exploring Random Tree) is a classic example of a sample-based method. The RRT algorithm, also known as the Rapidly Exploring Random Tree algorithm, was first proposed by La Valle in 1998 as an efficient path-planning algorithm. The RRT algorithm starts with an initial root node and searches the space through random sampling. Then, it adds one leaf node after another to continuously expand the random tree[12]. This algorithm randomly selects points from the input map samples, connects these points one by one to form a tree shape, and thus is named a rapidly exploring random tree.

When the target point enters the random tree, the expansion of the random tree stops immediately. A path from the starting point to the target point can be found at this point.

2.2.2 Graph-based methods

This method shares operating principles with the sample-based method and complements it. We divide the road surface into numerous grids and establish a search method using the search algorithms DFS (depth-first search) or BFS (breadth-first search) to determine the optimal solution for each step, thereby obtaining the optimal path [15, 16].

However, algorithms based on graph theory are greedy algorithms that can only obtain the optimal solution for each step based on the current selection and cannot access global search data, thus being unable to find the overall optimal path [18]. This method's other advantages and disadvantages are the same as those of the sample-based method and will not be elaborated on here.

Dijkstra is a classic example of the graph-based method, forming the foundation of many other planning approaches. The core idea of the Dijkstra algorithm is to start from the starting point, adopt

a greedy strategy, and traverse the adjacent node of the vertex that is the closest to the starting point and has not been visited [18]. The algorithm continues until it reaches the destination. The algorithm uses an array to store the shortest distances from the origin to each vertex. Initially, the path weight of the starting point is set to 0, and the weights of other vertices are set to infinity. Then, the algorithm continuously selects the unvisited vertex closest to the starting point and updates the distances of its adjacent vertices. This process continues until all vertices have been visited[31].

The algorithm steps are as follows: First, initialize the starting point, set the distance of the starting point to 0, and set the distance of all other vertices to infinity [20]. Then, select the unvisited vertex that is closest to the starting point and mark it as visited. Next, update the distances of the adjacent vertices of this vertex. If the distance through this vertex to the adjoining vertex is shorter, update the distance value. Repeat the above two steps continuously until all vertices have been visited[19].

A* is another graph-based classic algorithm. It virtualizes the map, dividing it into several small squares and marking them with different colors. Thus, the map can be represented by a two-dimensional array. For example, yellow squares represent the starting point, green represents the destination, black represents obstacles, and red represents search squares. An open-close list is established during the search, and the small squares are placed in the open or closed list according to their colors. The open list records all the cells considered for finding the shortest path, while the closed list records the cells that will not be considered again. By continuously searching for surrounding points, a new point is selected as the starting point to continue the search until the destination is reached[25].

The A* algorithm searches a two-dimensional grid. Essentially, it simplifies the vehicle to a particle, performs state sampling, and fixes the movement direction to eight directions (or four directions), while also determining the movement distance. However, the trajectory calculated by the A* algorithm may be infeasible because it does not consider the kinematics of the vehicle body in actual situations, such as the existence of right-angle bends in the trajectory. Therefore, scientists have made improvements on this basis and invented the hybrid A* algorithm, which introduces the heading angle, conducts control sampling, and transforms the search into a three-dimensional space search. It conforms to the vehicle kinematics model[26].

2.2.3 Methods based on deep reinforcement learning

This method combines two learning methods of artificial intelligence, deep learning and reinforcement learning, enabling computers to train functions through reinforcement learning and replace complex functions with those learned through deep learning. The essence of this method is to establish a reward mechanism that enables machines to learn step by step and ultimately obtain the optimal trajectory [2,27].

In the established simulation environment, the machine can continuously trial and error to find the optimal function solution and achieve the goal. However, reward functions are exceptionally complex and often difficult to express directly; therefore, deep learning is needed to replace these complex functions [27, 29]. It is used to fit functional relationships by providing data for computers to calculate functional relationships. In trajectory planning, deep learning is used to fit a reward function, which is an action function that determines the output driving action based on variables such as x , y , and v [2].

The advantage of this method is that it has a strong generalization ability and can perform reinforcement learning computations by fitting different reward functions in any environment [28].

The disadvantages of this method are that, firstly, the reward function is challenging to set, and it is still difficult to determine the exact functional relationship based on the provided data. Secondly, during the training process, deep learning requires high computational power from the machine, which necessitates a large amount of data to support the achievement of the goal [29]. Thirdly, the training process takes a considerable amount of time to generate the optimal trajectory for different road conditions. Fourthly, it isn't easy to judge the quality of training results, as machines may not be able to learn the optimal trajectory [30, 31].

These crucial problems are particularly evident in recent advances that combine MCTS with deep neural networks for function approximation.[43]. MCTS optimizes the search tree by iteratively selecting, expanding, simulating, and updating nodes, and then selects the optimal action policy [44].

2.2.4 Computational optimal control

For the optimal control problem of trajectory planning, we need to establish x and y axes on the road surface, and add the variable of velocity v to find the functions of x, y, and v with respect to time t. For the optimal trajectory, the essence of optimization also transforms from optimizing the function to optimizing a set of points. For example, for the function of x with respect to t, the values of x1 to x100 on this function can be found, and these values must meet specific requirements in practical situations, such as speed limits, vehicle dynamics, reaching the target point, and collision avoidance.

The advantage of this method is that it can find a function solution that approximates the optimal solution, which is better than the optimal paths found by the above three methods. The disadvantages of this method are: firstly, it takes a long time to collect a large number of numerical values and solve the function[34,35]; Secondly, even if enough values are collected, it may not be possible to obtain the function of the optimal trajectory, as the differences between discrete points and continuous functions still exist.

In conclusion, the method of computational optimal control is the most effective approach for achieving optimal trajectory planning. Theoretically, this method can ensure the computation of the optimal solution. However, the superiority of the algorithm needs to be considered in practice, which is also one of the future Development directions in this field.

This chapter primarily introduces the roles of planning modules and the implementations of trajectory planning in autonomous driving, and then analyzes the advantages and limitations of each method. Each one has its own benefits, but the process of Computational optimal control can balance the quality and time of planning, which is considered the best method introduced in this article. Therefore, this article will mainly focus on a further detailed analysis and review of this method.

Table 1 Comparative Analysis of Unmanned Driving Trajectory Planning Methods

	Principles	Advantages	Disadvantages	Classic cases
Sample-based	Divide the sample map into numerous grids, design eight directions of path from the starting points, and repeat the step until reaching the end point. Use a standard function to qualify the quality of paths. Divide the map sample into numerous grids and use search algorithms such as DFS or BFS to plan the trajectory.	Untrained problem, the function can reach maximum/minimum values without encountering any unsolvable issues.	Requires a significant amount of computation; determining the thickness of the grids is a challenging task. The quality of the planned trajectory depends on the quality of the sample, and the calculation time is considerable. The greedy algorithm can only obtain the optimal path for each step, but not the global data. Other disadvantages are the same as those of the	RRT (Rapidly-exploring Random Tree)
Graph-based		As with the advantages of the sample-based method.		A* algorithm Dijkstra

			sample-based method.	
Deep reinforcement learning	Let the machine learn a reward function through deep learning, establish a reward mechanism that enables machines to learn step by step, and ultimately obtain the optimal trajectory. By using the discrete method, transform the problem from finding a set of optimal functions between variables into finding numerous points in the functions. Optimize the points to ensure they also satisfy specific constraints.	Strong generalization ability	Difficult to set a reward function; requires strong computational power and a large amount of data, and it is difficult to judge the quality of learning.	MCTS-RL
Optimal control		It takes a long time to collect data and establish a function, and it may not achieve an optimal function.	Trajectory optimality and generalization.	Model Predictive Control (MPC)

3. Ideas and Methods for Computational Optimal Control

3.1 Concept of Computational Optimal Control

3.1.1 Technical Modeling

It is necessary to determine the relationship between acceleration and time and establish functions for translational acceleration (a), rotational acceleration (w), and time (t) separately [17]. A (t) represents the movement of the accelerator, while w (t) represents the movement of the steering wheel. Among them, the value of time t has an interval, such as $0 < t < 5$ [17].

To plan the optimal trajectory, we need to find a set of optimal functions for a and w with respect to t, where a and w have boundary constraints, such as maximum and minimum values. In addition, the setting of the function should also be combined with the actual kinematics of the vehicle body, such as the vehicle's inability to turn through right angles [36-40]. For goal setting, it is necessary to establish functions for process/state variables that satisfy the function constraints of a and w.

The essence of the optimal function is to aim at variables in the process and reach the goal of obtaining minimum values of the square of acceleration plus the square of time and the square of angular acceleration plus the square of time [41]. The minimum goal of square of time is also the second constraint set by the function.

3.1.2 Advantages

Compared to other trajectory planning methods, Computational optimal control can establish the best standards and limit the trajectory through boundary constraints of maximum and minimum

values, thereby meeting the requirements. This method aligns with the principles of process control and is therefore applied in the field of autonomous driving.

3.2 The development of Computational optimal control

The exploration stage occurred from the 1980s to the 1990s. People have developed mathematical models based on the dynamics and kinematics of vehicles, enabling a more accurate description of vehicle trajectories in real-world scenarios. Some early classic optimal control algorithms, such as Linear Quadratic Regulator (LQR), began to be applied to solve the trajectory planning problem in some simple situations, achieving straight-line path and speed control.[40]

After that, the method experienced a high-speed development stage. From the early 21st century to the present, with the rapid improvement of computer science technology, algorithms such as Model Predictive Control (MPC) have been able to achieve trajectory planning goals in more complex scenarios [41]. The technology of computing optimal control applies to the control quantity that meets the requirements of the optimal process.

3.3 Numerical Solution Method

Scientists explore the essence of functions and transform the problem of solving continuous functions into the issue of finding discrete points [46-48]. They use the discrete method to analyze the relationship between control quantities, such as translational acceleration, rotational acceleration, position, and time, of the vehicle on the road surface.

Among them, the discrete method is currently divided into two types: partial discretization and complete discretization. The partial discretization method only discretizes a portion of the variable, such as translational acceleration, while allowing other variables, like speed, to satisfy certain constraints [48]. The complete discretization method discretizes all variables that have many interrelated components, and is suitable for large-scale problems [49].

In conclusion, Chapter Three introduces the concept of Computational optimal control, its development, and its applications in various fields. Additionally, a more specific introduction to numerical solutions is provided in this chapter, where discrete solutions are discussed.

4. Research on Trajectory Planning Based on Computational Optimal Control

4.1 Trajectory Planning Principle Based on Optimal Control

To achieve the goal of solving the problem through technology, the optimal control function needs to be established first, such as the function relating acceleration, steering angle, and time. Secondly, the index needs to be established to clarify the standard of the optimal trajectory, for example, minimizing the acceleration and angular velocity as much as possible, which can be expressed as square of $a(t)$ plus square of $w(t)$ approaching zero on the function, while satisfying certain constraints about the vehicles themselves and in different situations[41].

Overall, we need to establish a set of indicators that can evaluate the quality of trajectory planning for autonomous driving. These indicators should be related to the control quantity and satisfy certain kinematic and motion condition constraints, such as the turning angles and maximum acceleration[36-40].

4.2 Advantages of Solving Trajectory Planning Problems Based on Optimal Control

For methods, sample-based and graph-based theories, firstly, these two methods can only find local optimal trajectories, but are difficult to find global optimal trajectories. Secondly, if sampling fails, it may not be possible to find a drivable trajectory. Thirdly, these two methods are qualitative analyses based on trial and error, but do not have function indicators to ensure that the planned path is the optimal path. For methods based on deep reinforcement learning, generalization is difficult to guarantee, and it may not apply to complex road conditions because the reward function is challenging

to establish. Moreover, it requires a large amount of data and high computational power, making the computation difficult.

In contrast, the optimal control technique, by establishing indicators and utilizing mathematical methods to solve, can ensure the optimality and generalizability of trajectories and is sure to be able to solve [17, 50, 51].

4.3 Latest Research Progress

For this technology, the unresolved issues at present are as follows: Firstly, the computation time of this method is still too long, and the problem-solving process is relatively slow. For instance, in an uncrewed vehicle traveling at a speed of 50m/s, it takes 0.5s to plan an optimal trajectory. During the computation period, the car has already traveled 25 meters, and there may be potential safety hazards in the absence of proper planning. Therefore, the long computation time is a current weakness of this technology, which also makes it one of the areas for future research improvement [52].

Secondly, the optimal control function may not be fully solved. When the function needs to set constraints for surrounding other vehicles and complex working conditions, conflicts may arise among the constraints. Therefore, the more constraints there are, the greater the possibility that the function has no solution, and the more strategies are needed to solve it [53,54].

Thirdly, in terms of computation, some problems under specific working conditions cannot be expressed by the optimal control function. Additionally, the solvers in the computer cannot solve conditional constraints, which are instructions on how to execute under specific circumstances [55].

In response to the existing problems, the proposed solutions at present are as follows: Firstly, for the problem of long computation time, the time should be reduced to enable the computer to generate the planning trajectory more quickly[52]. Secondly, to address the restrictions among constraints, the problem should be simplified [54,56]. Two approaches can be employed: either establishing a simpler problem or transforming a complex one into a simpler one. Thirdly, for computing problems, the selective constraints need to be converted into general constraints, or the solver needs to be modified to enable it to solve conditional limitations [55].

5. Conclusion

This paper first introduces the operating principles of four trajectory planning methods for unmanned systems, analyzes their respective advantages and disadvantages, and summarizes the superiority of optimal control computational methods. Secondly, this thesis introduces optimal control computation as a mathematical concept, its development history, and applications in aerospace, chemical control, and artificial intelligence, among others. It elaborates on the core method, the discrete method, its classification, and the underlying principle. Finally, the paper explains how to use optimal control computation to solve trajectory planning problems in unmanned systems. By establishing the optimal control function, two significant issues—establishing the indicators and satisfying the constraints — can be addressed, and a collision-free, safe, and efficient driving trajectory can be generated in real-time. By reviewing traditional trajectory planning methods, this paper analyzes the unique superiority of optimal control technology in solving the trajectory planning problem, specifically in terms of trajectory optimality and generalization. Therefore, computational optimal control has a promising application prospect and practical implementation in the field of trajectory planning, and can address the limitations of other algorithms.

As a research hotspot in planning methods, overcoming the shortcomings of this method is the future direction of effort for physicists. Firstly, due to the large amount of computation and the required computing time, this method must be further optimized to reduce computational time. Secondly, the constraints of the optimal control function are numerous. When the environment or the target becomes more complex, a situation may arise in which the constraints mutually restrict each other, resulting in no feasible solution. Therefore, the problem needs to be simplified or transformed.

Ultimately, for solving conditional constraints, it is necessary to either convert the constraints or design and develop solvers to address them.

Unmanned driving is a witness to the development of science and technology and a crystallization of human wisdom. Among them, the current research hotspot is computational optimal control to solve the trajectory planning problem of unmanned driving. Although the current technical conditions are limited and cannot universally achieve the goal of an optimal trajectory, with the joint efforts of scientists, uncrewed driving cars can carry people's dreams of science and technology, driving towards a brighter future.

References

- [1] Y. Liu and H. Gao, "Stability, scalability, speedability, and string stability of connected vehicle systems," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 52, no. 5, pp. 2819–2832, May 2022.
- [2] X. Xu, L. Zuo, X. Li, L. Qian, J. Ren, and Z. Sun, "A reinforcement approach to autonomous decision making of intelligent vehicles on highway learning systems," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 50, no. 10, pp. 3884–3897, Oct. 2020.
- [3] A. A. T., & B. H. S. M. (2016). Influence of Connected and Autonomous Vehicles on Traffic Flow Stability and Throughput. *Transportation Research Part C: Emerging Technologies*, 71, 143-163.
- [4] Talebpour, A., & Mahmassani, H. (2015). Influence of Autonomous and Connected Vehicles on the Stability of Traffic Flow. *Transportation Research Board Annual Meeting*.
- [5] A. A. T., & B. H. S. M. (2016). Influence of Connected and Autonomous Vehicles on Traffic Flow Stability and Throughput. *Transportation Research Part C: Emerging Technologies*, 71, 143-163.
- [6] F. Y. Wang, R. Qin, J. Li, Y. Yuan, and X. Wang, "Parallel societies: A computing perspective of social digital twins and virtual–real interactions," *IEEE Trans. Comput. Soc. Syst.*, vol. 7, no. 1, pp. 2–7, Feb. 2020.
- [7] X. Qian, F. Althé, J. Grégoire, and A. de La Fortelle, "Autonomous intersection management systems: Criteria, implementation and evaluation," *IET Intell. Transport Syst.*, vol. 11, no. 3, pp. 182–189, 2017.
- [8] Kurtz, J. (2002). DEMO III experimental uncrewed vehicle autonomous mobility system overview. *Intelligent Control*. IEEE.
- [9] A. Gholamhosseinian and J. Seitz, "A comprehensive survey on cooperative intersection management for heterogeneous connected vehicles," *IEEE Access*, vol. 10, pp. 7937–7972, 2022.
- [10] C. Liu, C.-W. Lin, S. Shiraishi, and M. Tomizuka, "Distributed conflict resolution for connected autonomous vehicles," *IEEE Trans. Intell. Veh.*, vol. 3, no. 1, pp. 18–29, Mar. 2018.
- [11] S.-M. Park and Y.-G. Kim, "A metaverse: Taxonomy, components, applications, and open challenges," *IEEE Access*, vol. 10, pp. 4209–4251, 2022.
- [12] Z. Zhou, B. Wang, M. Dong, and K. Ota, "Secure and efficient vehicle-to-grid energy trading in cyber physical systems: Integration of blockchain and edge computing," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 50, no. 1, pp. 43–57, Jan. 2020.
- [13] Wahl, H. G., Holzpfel, M., & Gauterin, F. (2014). Approximate dynamic programming methods applied to far trajectory planning in optimal control. *Intelligent Vehicles Symposium*. IEEE.
- [14] Anderson, S. J., Peters, S. C., Pilutti, T. E., & Iagnemma, K. (2011). An optimal-control-based framework for trajectory planning, threat assessment, and semi-autonomous control of passenger vehicles in hazard avoidance scenarios. *International Journal of Vehicle Autonomous Systems*, 8(2/3/4), 190-216.
- [15] K. Majd, M. Razeghi-Jahromi, and A. Homafar, "A stable analytical solution method for car-like robot trajectory tracking and optimization," *IEEE/CAA J. Automatica Sinica*, vol. 7, no. 1, pp. 39–47, Jan. 2020.
- [16] J. K. Subosits and J. C. Gerdes, "Impacts of model fidelity on trajectory optimization for autonomous vehicles in extreme maneuvers," *IEEE Trans. Intell. Veh.*, vol. 6, no. 3, pp. 546–558, Sep. 2021.
- [17] B. Li et al., "Sharing Traffic Priorities via Cyber–Physical–Social Intelligence: A Lane-Free Autonomous Intersection Management Method in Metaverse," in *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 53, no. 4, pp. 2025-2036, April 2023, doi: 10.1109/TSMC.2022.3225250.

- [18] Caldwell, C. V. (2011). A sampling-based model predictive control approach to motion planning for autonomous underwater vehicles. Dissertations & Theses Gradworks.
- [19] Wang, L. L. , Wang, H. J. , & Pan, L. X. (2014). Autonomous underwater vehicle motion planning via sampling-based model predictive control. *Applied Mechanics & Materials*, 670-671, 1370-1377.
- [20] Caldwell, C. V. , Dunlap, D. D. , & Collins, J. E. (2010). Application of Sampling-Based Model Predictive Control to an Autonomous Underwater Vehicle.
- [21] Bircher, A. , Alexis, K. , Schwesinger, U. , Omari, S. , & Siegwart, R. . (2016). An incremental sampling-based approach to inspection planning: the rapidly exploring random tree of trees. *Robotica*, 35(6), 1-14.
- [22] Loureiro, R., Merzouki, R., & Bouamama, B. O. (2012). Bond graph model based on structural diagnosability and recoverability analysis: application to intelligent autonomous vehicles. *IEEE Transactions on Vehicular Technology*, 61(3), 986-997.
- [23] Hao, Y., Laxton, B., Agrawal, S. K., Lee, E., & Benson, E. (2003). Planning and Control of UGV Formations in a Dynamic Environment: A Practical Framework with Experiments. *Robotics and Automation*, 2003. Proceedings. ICRA '03. IEEE International Conference on. IEEE.
- [24] Tamura, S., Ozaki, T., Noda, T., & Maekawa, T. (2008). 3110 path generation for an autonomous vehicle by B-spline curves. *The Proceedings of Design & Systems Conference*, 411-414.
- [25] Devaragudi, S. R., & Chen, B. (2019). MPC-BASED CONTROL OF AUTONOMOUS VEHICLES WITH LOCALIZED PATH PLANNING FOR OBSTACLE AVOIDANCE UNDER UNCERTAINTIES. *International Design Engineering Technical Conferences & Computers and Information in Engineering*.
- [26] Isele, D. , Rahimi, R. , Cosgun, A. , Subramanian, K. , & Fujimura, K. (2018). Navigating Occluded Intersections with Autonomous Vehicles using Deep Reinforcement Learning. 2018 IEEE International Conference on Robotics and Automation (ICRA). IEEE.
- [27] Conlon, N. , Acharya, A. , Mcginley, J. , Slack, T. Z. , Hirst, C. A. , & D'Alonzo, M. , et al. (2022). Generalizing competency self-assessment for autonomous vehicles using deep reinforcement learning. AIAA SCITECH 2022 Forum.
- [28] Song, L. (2022). Impacts of Connected and Autonomous Vehicles on Deep Reinforcement Learning Controlled Intersection Systems. (Doctoral dissertation, The University of North Carolina at Charlotte.).
- [29] Isele, D. , Rahimi, R. , Cosgun, A. , Subramanian, K. , & Fujimura, K. (2017). Navigating occluded intersections with autonomous vehicles using deep reinforcement learning.
- [30] Zhao, D. (2022). Deep Reinforcement Learning for Perception and Control of Autonomous Vehicles.
- [31] Dam, T., D'Eramo, C., Peters, J., & Pajarinen, J. (2024). A unified perspective on value backup and exploration in Monte-Carlo tree search. *Journal of Artificial Intelligence Research*, 81.
- [32] Tomas, J. P. Q. , Aguas, N. J. R. , Villa, A. N. D. , & Lim, J. R. G. . (2021). Developing an adaptive AI agent using supervised and reinforcement learning with Monte Carlo tree search in Fighting Ice. Proceedings of the 2021 4th International Conference on Computational Intelligence and Intelligent Systems.
- [33] B. Li and Y. Zhang, "Fault-tolerant cooperative motion planning of connected and automated vehicles at a signal-free and lane-free intersection," *IFAC-PapersOnLine*, vol. 51, no. 24, pp. 60–67, 2018.
- [34] B. Li, Y. Zhang, Y. Zhang, N. Jia, and Y. Ge, "Near-optimal online motion planning of connected and automated vehicles at a signal-free and lane-free intersection," in *Proc. IEEE Intell. Veh. Symp. (IV)*, 2018, pp. 1432–1437.
- [35] J. Fu, F. Tian, T. Chai, Y. Jiang, Z. Li, and C.-Y. Su, "Motion tracking control design for a class of nonholonomic mobile robot systems," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 50, no. 6, pp. 2150–2156, Jun. 2020.
- [36] X. Peng, K. Guo, X. Li, and Z. Geng, "Cooperative moving-target enclosing control for multiple nonholonomic vehicles using feedback linearization approach," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 51, no. 8, pp. 4929–4935, Aug. 2021.
- [37] K. Bergman, O. Ljungqvist, and D. Axehill, "Improved path planning by tightly combining lattice-based path planning and optimal control," *IEEE Trans. Intell. Veh.*, vol. 6, no. 1, pp. 57–66, Mar. 2021.

- [38] J. Liu, X. Dong, J. Wang, C. Lu, X. Zhao, and X. Wang, "A novel EPT autonomous motion control framework for an off-axle hitching tractor-trailer system with drawbar," *IEEE Trans. Intell. Veh.*, vol. 6, no. 2, pp. 376–385, Jun. 2021.
- [39] Z. Wang, L. Wang, H. Zhang, L. Vlacic, and Q. Chen, "Distributed formation control of nonholonomic wheeled mobile robots subject to longitudinal slippage constraints," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 51, no. 5, pp. 2992–3003, May 2021.
- [40] B. Li, Y. Zhang, Y. Feng, Y. Zhang, Y. Ge, and Z. Shao, "Balancing computation speed and quality: A decentralized motion planning method for cooperative lane changes of connected and automated vehicles," *IEEE Trans. Intell. Veh.*, vol. 3, no. 3, pp. 340–350, Sep. 2018.
- [41] Suganuma, N. (2019). Trends in the Development of Autonomous Vehicles and Challenges for Deployment in Society - ScienceDirect. *IATSS Research*, 43(4), 242-243.
- [42] Rahman, M. (2018). Development and evaluation of connected and autonomous vehicle applications for transportation cyber-physical systems.
- [43] Bryson, A. E., & Yu-Chi Ho. (1975). *Applied optimal control*. Technometrics.
- [44] Turner, J. D., & Chun, H. M. (2012). Optimal distributed control of a flexible spacecraft during a large-angle maneuver. *Journal of Guidance Control & Dynamics*, 7(3), 257-264.
- [45] Dayong, G. . (2002). The application of optimal control theory in the electro-hydraulic servo system. *Machine Tool & Hydraulics*.
- [46] Vassiliadis, V. S., Canto, E. B., & Banga, J. R. (1999). Second-order sensitivities of general dynamic systems with application to optimal control problems. *Chem.eng*, 54(17), 3851-3860.
- [47] Rao, A. K., & Naidu, D. S. (1982). Singular perturbation method applied to the open-loop discrete optimal control problem. *Optimal Control Applications and Methods*.
- [48] Wright, S. J. (1993). Interior Point Methods for Optimal Control of Discrete-Time Systems. *Journal of Optimization Theory & Applications*, 77(1), 161-187.
- [49] Pchelkin, S. S., Shiriaev, A. S., Robertsson, A., & Freidovich, L. B. (2014). Integrated time-optimal trajectory planning and control design for an industrial robot manipulator. *IEEE/RSJ International Conference on Intelligent Robots & Systems*. IEEE.
- [50] Zhang, Z., Yu, X., Jin, Z., Ying, Y., Hua, R., & Lin, X.. (2015). Trajectory planning and optimal lateral stability control under multiple barriers for an intelligent vehicle. *World Journal of Engineering & Technology*, 03(5), 1-7.
- [51] Li, B., & Shao, Z.. (2015). An incremental strategy for tractor-trailer vehicle global trajectory optimization in the presence of obstacles. *IEEE*.
- [52] M. Amouzadi, M. O. Orisatoki, and A. M. Dizqah, "Optimal lane-free crossing of CAVs through intersections," *IEEE Trans. Veh. Technol.*, early access, 15 September 2022, doi: 10.1109/TVT.2022.3207054.
- [53] Shaohua, Meng, Jinwu, Xiang, Zhangping, & Luo, et al. (2014). A novel trajectory planning strategy for aircraft emergency landing using the Gauss pseudospectral method. *Control Theory & Technology*.
- [54] Cowling, I. D., Whidborne, J. F., & Cooke, A. K. (2006). OPTIMAL TRAJECTORY PLANNING AND LQR CONTROL FOR A QUADROTOR UAV. *UKACC Int. Conf. Control 2006 (ICC2006)*.
- [55] B. Li and Y. Zhang, "Fault-tolerant cooperative motion planning of connected and automated vehicles at a signal-free and lane-free intersection," *IFAC-PapersOnLine*, vol. 51, no. 24, pp. 60–67, 2018.
- [56] Li, X., Sun, Z., Cao, D., Liu, D., & He, H. (2017). Development of a new integrated local trajectory planning and tracking control framework for autonomous ground vehicles. *Mechanical Systems and Signal Processing*, 87(Pt.B), 118-137.