ISSN 2959-6157

Research on unmanned vehicle trajectory planning problem based on genetic algorithm

Haokun Xie

School of Leeds, Southwest Jiaotong University, Chengdu 610100, China

Abstract:

In this study, a method based on Genetic Algorithm and optimal control for path planning of an unmanned vehicle is proposed. This work discretizes the optimal control problem into a nonlinear programming problem and resolves it to enhance the initial answer. All physical quantities are incorporated as variables into the problem-solving process and limited by an objective function to derive the optimum trajectory.

Keywords: unmanned vehicle system; Genetic Algorithm; Optimal Control Problem; path planning

1. Introduction

Research on unmanned vehicle systems is vital not only for driving forward technological innovation but also for its profound influence on future transportation frameworks. Guided by initiatives like China's Intelligent Connected Vehicle Development Action Plan and the Outline for the Construction of a Strong Transportation Nation, the integration of intelligent transportation has been firmly established as a cornerstone in China's strategic planning. These policies foster the development of autonomous driving technologies, encouraging advancements in areas such as vehicle-to-infrastructure communication and smart infrastructure integration. The overall goal is to cultivate a supportive environment that accelerates the progress of unmanned vehicle systems, aligning with the objectives of modernization and sustainable growth.

By 2025, we expect to see some highly autonomous vehicles enter commercial service, showcasing China's ambition in the global technology race. This marks a significant step toward integrating unmanned vehicles into everyday life. At the same time, similar policies that are presented in the United States and Europe emphasize that the development of autonomous vehicle technologies should be prioritized. These policies promote innovation in technology while laying the groundwork for transportation systems in the future. Researches on unmanned vehicle systems enable us to follow the advancement of future transportation. This research on unmanned vehicle systems could address the congestion to some extent since unmanned vehicles could make the best decision under complex circumstances. Besides, the systems would consider the cost of fuels and this would decrease the carbon emission. This implies that research into unmanned systems not only contributes to addressing global challenges like resource depletion but also achieves the aim of sustainability. Ultimately, research in this field could promote technological innovation and create new opportunities for future living and the sustainable growth of our planet.

Research on unmanned vehicles will undoubtedly help mankind in the future. There are three primary modules of the unmanned vehicle system which are perception, planning, and control. These modules collaborate to ensure the safety and efficiency of vehicles.

Various sensors equipped in the perception module would collect and interpret data from the surrounding environment by monitoring road conditions and tracking the movements of other road users. These collected data enable the system to avoid potential risks by analyzing the information within it. The planning module plays a central role by integrating sensor inputs with mission objectives, traffic rules, and road conditions. It produced a safe and effective driving route that takes the vehicle dynamics into account. This module connects the perception and control modules by receiving information from sensors and then sending the ideal path to the control module. It is the vital factor that affects the consistency of the system. A good planning module could help unmanned vehicles to have a good performance when auto driving. Ultimately, the control module will precisely regulate the vehicle's movements according to the route outlined by the planning module. The control module is responsible for adjusting the steering, braking, and acceleration of the vehicle to maintain its position on the predefined path. Concurrently, the module modifies the control according to the surrounding dynamic environment, thereby ensuring the secure operation of the vehicle. This is because the planning module that stands between sense and control modules connects these two. It not only processes the data but also turns it into actionable commands. The planning module determines how well the vehicle drives.

There are two types of planning systems. Single Vehicle Planning systems, Multi-Vehicle Planning systems. Single-vehicle planning systems allow individual cars to navigate autonomously using sophisticated decision-making algorithms that guarantee safe and efficient movement in more complex scenarios. Further, a single vehicle planning system can bypass dynamic obstacles on the way. The multi-vehicle planning system extends this to be multi-agent-capable, planning an entire vehicle trajectory on individual vehicles and ensuring collision avoidance among all autonomous vehicles It is the single-vehicle planning system fundamental to the multi-vehicle planning system. However, outages in the single-vehicle scheduling system are leaking energy for proper operation of a multi-vehicle scheduling system. The reason is that the multi-vehicle planning system requires unmanned vehicles to independently achieve accurate judgment in making plans. A single-vehicle planning system is needed to provide the safety of the vehicle in autonomous operation (it is a necessary condition for collaborative multi-vehicle planning). Hence, it requires a detailed study. Five main techniques have been used in trajectory planning for unmanned vehicles cycle using the literature review graph theory, optimization, sampling, reinforcement learning, and heuristics are the main solutions currently adopted. One of them, Vehicle Trajectory Prediction Based on Graph Convolutional Networks in Connected Vehicle Environment [1], describes a trajectory prediction model using graph convolutional networks (GCNs) in a connected vehicle environment. The method constructs spatial and temporal graphs and extracts interaction features using GCNs. The method improves the prediction accuracy by 8% compared to existing models and proves its effectiveness in complex traffic scenarios (MDPI). Intelligent Vehicle Path Planning Based on Optimized A* Algorithm [2] focuses on the improvement of the A* algorithm for dynamic environments for real-time path planning in dynamic environments. The improved A* can effectively handle obstacles and traffic flow, and achieve faster computation speed and smoother trajectories.

Sampling methods would generate paths by randomly sampling spatial points. These methods such as RRT are often used in complex planning issues. The article ‘HPO-RRT: A Sampling-Based Algorithm for UAV Real-Time Path Planning in Dynamic Environments’ [3] points out a time-based fast exploration random tree (time-based RRT *) algorithm is proposed, which solves the difficulty of trajectory planning when facing the moving threat, and enables it to quickly identify the moving object. Another article Dynamic path planning based on the fusion of improved RRT and DWA algorithms[4]Proposed to combine the advantages of both by combining RRT and DWA algorithms. It enables efficient trajectory planning for unmanned vehicles and can avoid unknown obstacles.

The Reinforcement learning approach is a method for learning and reinforcement trajectory planning strategies by interacting with the surroundings. This method means that driverless cars can make more suitable trajectory planning in a complex and dynamic environment. The article Reinforcement-Learning-Based Trajectory Learning in Frenet Frame for Autonomous Driving [5] proposes a trajectory learning (RLTF) based on reinforcement learning in the Frenet framework, which involves learning the trajectory in the Frenet framework. This enhances the interpretability of the planned trajectory. Finally, the experiment proves the feasibility of the method in the face of complex routes such as continuous detours. Another article, Deep Reinforcement Learning Lane-Changing Decision Algorithm for Intelligent Vehicles Combining LSTM Trajectory Prediction[6], proposed an algorithm, The algorithm utilizes deep deterministic policy gradient (DDPG) reinforcement learning, It is also integrated with long-term and short-term memory (LSTM) trajectory prediction model (called LSTM-DDPG). This greatly solves the problem of driverless cars changing lanes when there are cars around them.

The way to generate paths quickly through empirical rules and heuristic functions is the heuristic method. This method can ensure safe and effective trajectory planning in a simple environment. Intersection management for autonomous vehicles: a heuristic approach[7] proposes a heuristic method to conduct the management of unmanned vehicle traffic intersections, which can achieve efficient traffic at traffic intersections and avoid traffic jams. At the same time, the improved heuristic Bi-RRT algorithm is proposed to accelerate the path planning and minimize the redundant path points, which makes the path more relaxed.

Optimization-based methods using mathematical optimization techniques. This method can satisfy both safety and dynamic constraints while ensuring the planned trajectory optimization objective functions such as time or energy consumption. Spatio-Temporal Joint Optimization-Based Trajectory Planning Method for Autonomous Vehicles in Complex Urban Environments[8] introduces multi-constraint-based co-optimized trajectory planning (SJOTP) for complex urban environments. This method comprehensively considers obstacles and automobile motion models, jointly optimizes multi-constraints and multi-targets, and finally realizes fast and accurate path planning in complex urban environments.

In addressing an Optimal Control Problem (OCP), particularly in complex nonlinear systems, the choice of an initial solution is a determining factor in the success of the optimization process. Choices of initial solutions properly can help to improve the computational efficiency of solutions and lead to quicker convergence. At the same time, an ideal step-0 solution will also give you some proxy of how good is your end solution going to be. If you pick a bad initial solution, the algorithm may be trapped in a local minimum. This can slow down the eventual convergence, and you may never even find a proper solution. Through the study on how more optimal to formulate the initial solutions, this chapter considers here how different metrics of performance may eventually resolve in OCP situations.

To apply the result of this research in practice, an unmanned vehicle is modeled by the Kinematic Bicycle Model (KBM). We use KBM because it is a well-known vehicle dynamic model for lower speeds and this makes it easier to replace these tools with those in autonomous vehicle analysis. Furthermore, a simple shaped model (KBM) is widely applied in trajectory planning under collision avoidance requirements. To inform that initial estimate, the VA must consider key physical factors —velocity, steering angles, and acceleration. Although this makes it a perfect base for optimization, also means faster convergence to globally optimal solutions in terms of computing efficiency.

2. Construction based on OCP problems

To achieve the study of trajectory, it is particularly attractive to implement optimal control problem (OCP) within single-vehicle trajectory planning due to employing multiple methodologies like global/ off-the-shelf optimization, multi-objective optimization (MOO), and complex constraints formulation. Path planning settingThe incorporation of optimal control problem (OCP) into the path planning framework improves the planning process so that it can search for some kind of optimal solution considering diverse operational and physical constraints. With this approach, we can optimize different objectives in parallel (for example: path length, energy, and smoothness of the path) This compendium of goals represents the best path forward for accelerating autonomous vehicles to efficient and comfortable operation. Moreover, employing OCP for this objective facilitates the integration of a broader spectrum of physical limitations, including speed limits and turning radii, thereby ensuring that the generated trajectories are suitable for practical use. The OCP is an essential instrument for complex trajectory planning of autonomous vehicles, effectively merging theoretical limitations with practical adaptability.

2.1 Simple Bicycle Model

The Simple Bicycle Model is often used to represent the kinematics of unmanned vehicles when researching self-driving vehicles (CAVs). There are many quantities in KBM to be explained and studied.

This simple bicycle model is a model which can simulate the motion of unmanned vehicles in two dimensions. This model could illustrate the vehicle's position, velocity, and steering. Firstly, equations $x(t)=v(t)cos(\theta(t))$ and $y(t)=v(t)sin(\theta(t))$ describe the velocity components of the vehicle in the x- and y-axis directions, respectively. These two components are determined by the velocity of car v(t) and vehicle orientation angle θ(t). Vehicle orientation angle θ(t) Indicates the direction of the vehicle relative to the x-axis. The third equation $\theta(t)=\frac{v(t)}{L\omega}tan(\phi(t))$ describes the steering speed of the vehicle, which is the result of the combination of the vehicle speed v(t), the axle distance $L\omega$ (i.e., the distance between the front wheels and the rear wheels), and the steering angle ϕ(t), where ϕ(t) expresses the degree of steering through its tangent. Together, these equations determine how freely the vehicle moves and steers in a space with no external constraints.

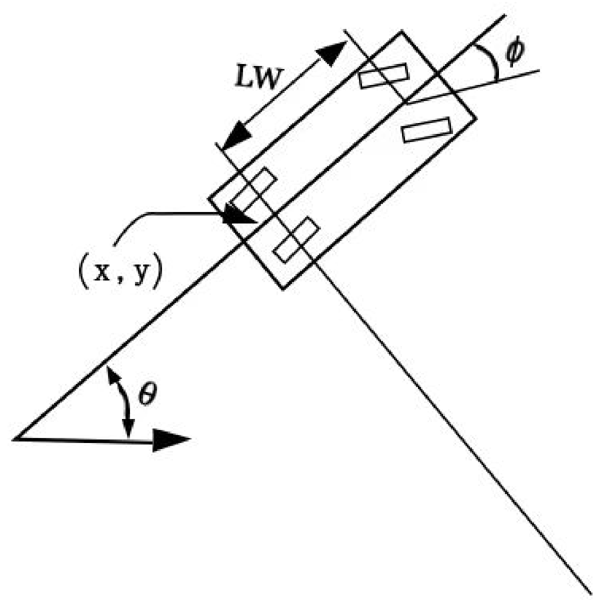

Fig. 1 Figure of simple bicycle model

2.2 Optimal control problems

In optimal control problems (OCPs), minimizing energy consumption is frequently identified as a pivotal performance metric, particularly in the context of dynamic planning for self-driving vehicles (CAVs). In this context, the energy consumption is typically represented by the sum of the squares of the acceleration a(t) and the steering angular velocity ω(t). The acceleration squared reflects the energy consumed by the vehicle due to speed changes. Its squared form ensures a positive energy consumption and imposes a greater limit on high acceleration by squaring. This motivates the algorithm to look for smoother speed trajectories, which improves energy efficiency and ride comfort. The squaring of the steering angular velocity, on the other hand, accounts for the energy expenditure associated with steering maneuvers and reduces the frequency and intensity of steering inputs, thereby enhancing vehicle ride stability and safety. The objective function combines these two factors to achieve an optimal solution that balances the relationship between speed, energy consumption, ride comfort, and safety. This is done by adjusting the corresponding weighting coefficients to ensure that the autonomous driving system achieves efficient use of energy while maintaining responsiveness and smooth operation.

2.3 Constraints

In the context of optimal control problems (OCPs), prevalence constraints are employed to guarantee that the behavior of self-driving vehicles (CAVs) adheres to the prescribed physical laws and operational constraints, including speed limits and steering angle limits. These constraints are the key factors to guarantee the vehicle operates safely and practically when facing complex conditions. The speed constraint 0 ≤ vi(t) ≤ v max implies that the vehicle's speed should not exceed the maximum speed limit. This way the possibility of accidents caused by high speed would be decreased. The steering angle constraint $\phi i(t) \le \phi$max ensures that the vehicle's steering angle will not be too large to lose control while passing obstacles. The above constraints not only help to more accurately simulate the behaviors of real vehicles, but also help to ensure the safety and feasibility of the planned paths of the unmanned vehicle system. This allows the vehicle trajectory to be more practical while adhering to safety guidelines for comfortable and efficient driving.

2.4 conclusion

All the equations are listed below

$$x(t) = v(t)cos(\theta(t))$$

$$y(t) = v(t)sin(\theta(t))$$

$$\theta(t) = \frac{v(t)}{L\omega}tan(\phi(t))$$

0≤vi(t)≤vmax0

$$\phi i(t) \le \phi \text{max}$$

3. OCP problem-solving and initial solution construction

3.1 Discrete

Most often converting a continuous time dynamic system to a discrete-time model is needed when tackling optimum control problems (OCP). In this work, we employed interpolation to discretize the initial solution into 50 x-y points and reduced a complex trajectory issue to a 1D optimization. By formulating the trajectory planning problem as a finite-dimensional optimization, the original issue with an infinite number of dimensions is greatly simplified and hence the solution accuracy is improved. This approach picks a time step and derives the discretization from it. The original continuous solution is discretized by uniform interpolation, and the obtained values get fed into the physical model to calculate the steering angle together with other needed physical quantities for real-world use. Furthermore, limitations on energy consumption and vehicle kinematics are incorporated to enhance the realism and efficiency of the vehicle trajectory. This discrete method facilitates precise trajectory planning for practical applications and enhances answers to intricate optimal control problems.

3.2 Construction of the initial solution

The transformation of a complex trajectory planning issue into a nonlinear programming (NLP) problem is a crucial step in solving an optimum control problem (OCP). This will convert a dynamical system optimization into a numerical optimization of the objective function, significantly simplifying the task. The initial solution will then be optimized using interior point optimization techniques. An appropriate initial solution can enhance algorithm convergence and reduce the likelihood of local optimality. This approach reduces the amount of time required to produce the solution and provides a faster overall better answer. After you have an initial solution, it is common to use gradient-based methods to improve the solution using gradient descent. This process helps to improve the efficiency and consistency of the solution according to the given logic & requirements. A good initial solution is necessary for optimal path-finding; the better the original solution, the more likely that using this algorithm will find an optimal solution. More importantly, the preliminary answer should respect constraints such as maximum speed, comfort, and energy usage. By doing so, the degrees of freedom are constrained. Also, it allows us to build up a more efficient and practical realization of the first solution under real driving conditions.

3.3 Method for solving the initial solution

The effectiveness of the initial solution is crucial in optimal solution control problem, since it directly influences the speed and quality of the final solution. Several methods, such as A* search, heuristics, and dynamic programming (DP), are commonly utilized to obtain an optimal initial solution.

A* search is a graph search algorithm that determines the ideal path by assessing the function f(n). In this equation, f(n) equals g(n) plus h(n), where g(n) denotes the actual cost function from the initial point to node n, and h(n) signifies the predicted cost function for the remaining distance. This strategy is predominantly efficient, as it integrates actual and predicted expenses. Conversely, heuristics provide an efficient search utilizing accumulated knowledge and experience, thereby enabling the rapid attainment of a solution. It is important to note, however, that heuristics are not guaranteed to perform perfectly in extremely complex systems and may not even identify the global optimal solution. In contrast, dynamic methods represent the most commonly utilized strategy, wherein the problem is decentralized to each stage and solved sequentially before the solutions are integrated. This circumvents the issue of failing to identify the global optimal solution. ALL approaches can be highly effective depending on the complexity of the problem and the computational resources available. However, they can rapidly provide a feasible solution and are frequently employed in the initial exploration of complex problems. Dynamic programming is a method that employs the decomposition of problems into sub-problems to solve complex problems. This is achieved by recursively solving each sub-problem and storing the results of solved sub-problems, thus avoiding repeated computations and effectively finding the global optimal solution. Each of these three methods has its advantages and limitations. The choice of which method to use depends on the nature of the specific problem, the quality requirements of the solution, and the limitations of computational resources. In practice, these methods can be used individually or in combination to improve solution efficiency and solution quality.

4. Initial solution construction based on the genetic algorithm and the solving of the OCP problem

4.1 Solving the steps of the genetic algorithm

In the Genetic Algorithm (GA) based path planning solution, a multitude of pivotal parameters have been established with the objective of to emulate the authentic scenario, thereby rendering the preliminary solution more realistic. In the genetic algorithm (GA), the population size is set to 100. This configuration guarantees the conveyance of diversity, thereby facilitating a more comprehensive examination of potential paths and ultimately leading to a more complete path solution. The algorithm is run for up to 200 generations, which balances the computational resources and ensures that there are sufficient evolutionary processes to optimize the initial solution. Each chromosome in the population encodes a sequence of 40 kinematic variables, which are used to produce coordinate values (x, y) corresponding to the two-dimensional plane.

The fitness function is the catalyst for the evolutionary algorithm, aiming to minimize the Euclidean distance between the final path point and the target destination. This promotes the solution's proximity to the target endpoint, aligning with the algorithm's goal of accurate pathfinding. The concluding phase of the process involves incorporating a constraint into the evolutionary algorithm, so fulfilling the goal of obstacle avoidance. The constraint verifies if the path's points intersect with the circular obstacle set. In the case of a junction, the obstruction is navigated by imposing a more severe penalty for such an occurrence.

4.2 Initial solution results of the genetic algorithm

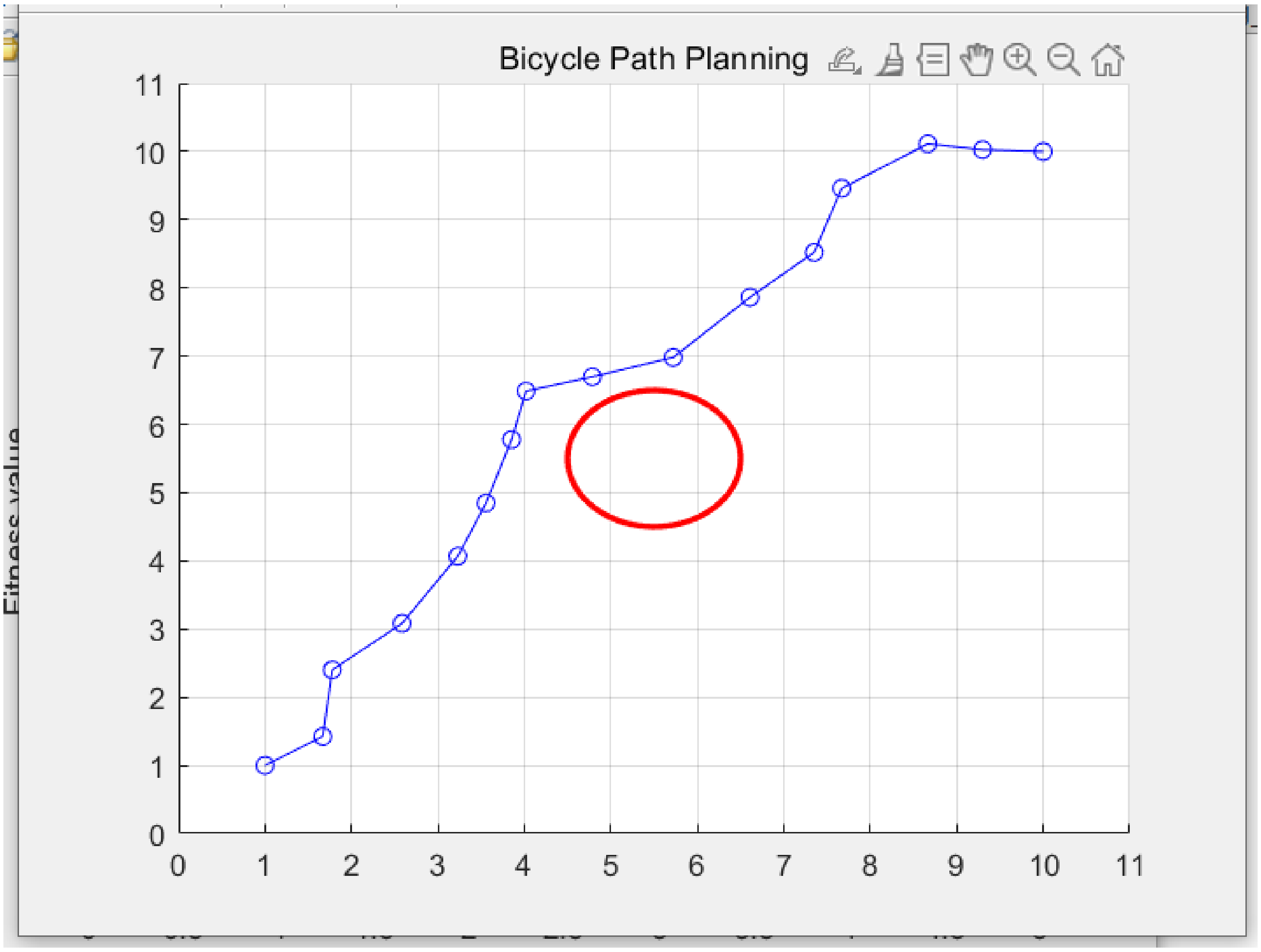

Fig. 2 The path generated by Genetic Algorithm

The image illustrates the outcome of a path-planning exercise that employed a genetic algorithm. The image depicts a grid with a blue path that begins at the bottom left and terminates at the top right. This path effectively circumvents a substantial red circular obstacle. The path successfully circumvents the obstacle by deviating significantly from its trajectory, thereby demonstrating that the algorithm is capable of respecting the imposed spatial constraints.

It seems reasonable to posit that the start point is situated in the lower left quadrant of the grid, while the endpoint is located in the upper right quadrant.

The coordinates of the obstacle center are approximately (5.5, 5.5) on the grid, with an obstacle radius of approximately 2 units. This positioning necessitates a detour around the obstacle to reach the destination.

The genetic algorithm employs a population size of 100, which indicates a robust search through the path space. The algorithm operates over a maximum of 200 generations, allowing the solution to refine and converge towards an optimal path.

Each member of the population encodes a sequence of steps in the path, represented by 40 genes (20 pairs of x and y increments), providing the necessary granularity to navigate complex routes, such as those involving the avoidance of obstacles. The plot visually confirms that the algorithm's configuration—specifically the balance of exploration (via population size and mutation) and exploitation (through selection and crossover)—is effectively tailored to solve the problem of navigating through constrained environments without violating defined boundaries.

4.3 Discrete of initial solutions of the GA

The process of interpolation and discretization is applied to a set of initial coordinate data extracted from a Comma-Separated Values file. The process begins with the loading of the raw x and y coordinates using NumPy's `genfromtxt` function, which reads data from the file while skipping the header and focusing on the first two columns. Subsequently, cubic spline interpolation is employed through SciPy's `interp1d` function, which is defined for both x and y coordinates over a normalized domain from 0 to 1. This interpolation is based on the original data points, which are resampled to create a denser and smoother sequence of 50 new data points spanning the same range. This densification permits a more exact calculation of derivatives and angular changes in subsequent stages of the analysis, effectively transforming sparse coordinate data into a more continuous and usable form for further motion analysis and physical quantity calculations.

4.4 Construction of the OCP problem

One hundred pairs of corresponding x and y values were derived after discretizing the initial solution produced by the genetic algorithm. The vehicle's steering angle, together with variable parameters such as acceleration and angular velocity, can be derived from the equations of the simple bicycle model with one hundred discretized coordinate sets. The discretized data for each physical quantity can be acquired and incorporated as variables into the solution. Acceleration (a) and rotational velocity (w), indicative of comfort and energy consumption, serve as objective functions, while constraints, including the relevant physical vehicle model, enhance the realism of the resultant path. By aggregating all physical quantities as variables within the solver and subsequently constraining them through the objective function and limitations, the approach can provide a more refined trajectory based on the initial solution. This more even route conserves greater energy.

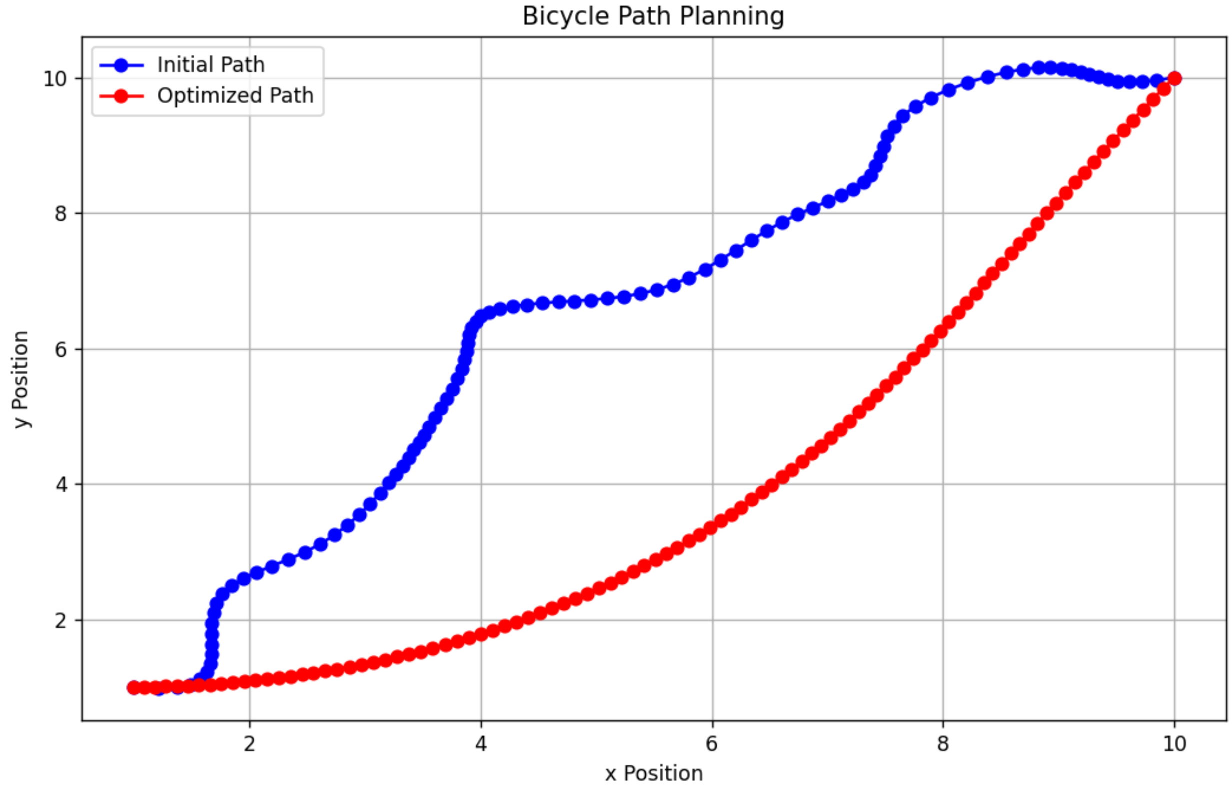

Fig. 3 Figure of the initial path and optimized path

This figure illustrates the preliminary solution and the refined trajectory. The preliminary solution, depicted by the blue segment, exhibits rapid steering throughout the motion, evidenced by pronounced fluctuations and steering along the x- and y-axes. These situations indicate that the path is devoid of constraints or limitations. Conversely, the optimal path depicted by the red line exhibits a remarkably smooth trajectory, devoid of abrupt variations and steering maneuvers. This results from the vehicle's operational limitations, which are designed to maximize path optimization by utilizing energy consumption and comfort as target functions.

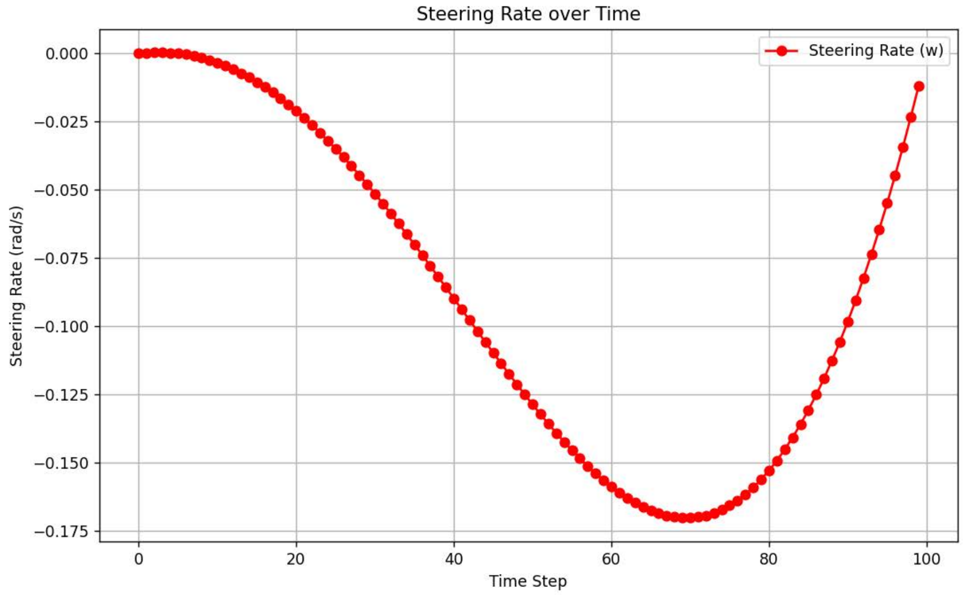

Fig. 4 Figure of the steering rate over time

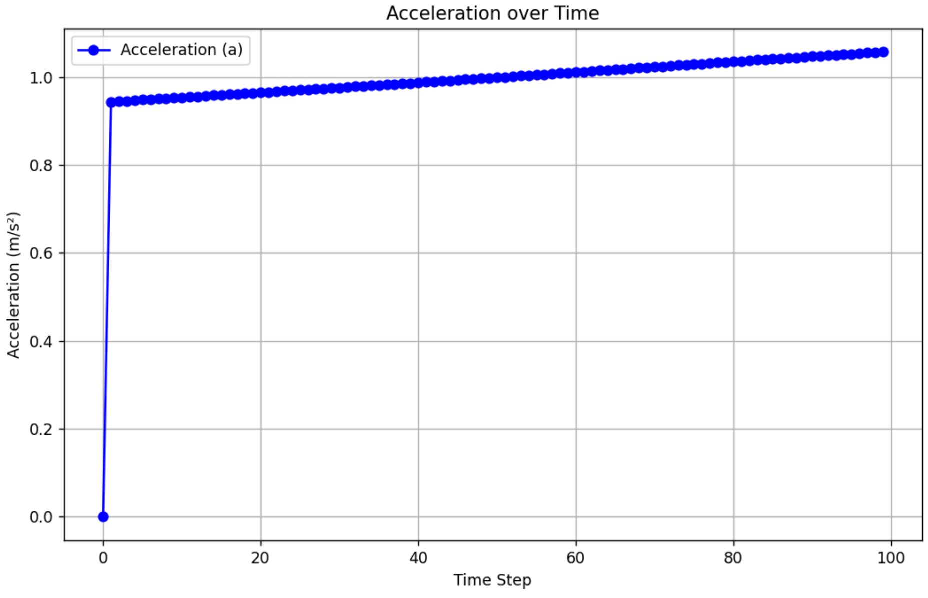

Fig. 5 Figure of the acceleration over time

The two plots illustrate the variations in acceleration and steering rate over time, respectively. The initial graph indicates that the vehicle experiences quick acceleration, increasing from approximately 0 to almost 1 m/s², due to the initial speed being established at 1 m/s². Subsequently, the analysis of the images indicates that the vehicle gradually accelerates with a very uniform increase in speed. This technique entails that during optimization, the system regulates the vehicle's speed by gradual acceleration, so preventing abrupt accelerations that could discomfort the occupants. Simultaneously, energy consumption, as a function of the square of the acceleration, is minimized while maintaining a constant acceleration. The second graph illustrates the variation in steering rate, which decreases from nearly 0 to -0.175 rad/s and thereafter ascends back to 0. This indicates that the vehicle reverts to a nearly straight trajectory following a gentle turn. This minimizes the pain experienced by passengers during sharp turns, significantly enhancing overall comfort. The optimization method efficiently regulates variations in acceleration and steering rate, guaranteeing a fluid and agile vehicle trajectory.

5. Conclusion

This work proposes a path-planning strategy for unmanned vehicles utilizing optimal control problem modeling. The computational optimal control problem is transformed into a nonlinear planning problem and resolved. The findings indicate that the selection of the first solution significantly affects computing efficiency and path smoothness. This strategy optimizes the course within the restrictions of the objective function, greatly diminishing rapid acceleration and sharp curves, enhancing energy efficiency, and improving ride comfort. Moreover, the implementation of physical constraints (e.g., velocity and steering angle limitations) guarantees that the produced trajectories are practical and secure for real-world applications.

References list:

[1]Shi, J., Sun, D., & Guo, B. (2023). Vehicle Trajectory Prediction Based on Graph Convolutional Networks in Connected Vehicle Environment. Applied Sciences, 13(24), 13192.

[2]Chu, L., Wang, Y., Li, S., Guo, Z., Du, W., Li, J., & Jiang, Z. (2024). Intelligent Vehicle Path Planning Based on Optimized A* Algorithm. Sensors, 24(10), 3149.

[3]Guo, Y., Liu, X., Jia, Q., Liu, X., & Zhang, W. (2023). HPO-RRT*: a sampling-based algorithm for UAV real-time path planning in a dynamic environment. Complex & Intelligent Systems, 9(6), 7133–7153.

[4]Dynamic path planning based on the fusion of improved RRT and DWA algorithms. (2023, May 26). IEEE Conference Publication | IEEE Xplore.

[5]Yoon, S., Kwon, Y., Ryu, J., Kim, S., Choi, S., & Lee, K. (2024). Reinforcement-Learning-Based Trajectory Learning in Frenet Frame for Autonomous Driving. Applied Sciences, 14(16), 6977.

[6]Yang, Z., Wu, Z., Wang, Y., & Wu, H. (2024). Deep Reinforcement Learning Lane-Changing Decision Algorithm for Intelligent Vehicles Combining LSTM Trajectory Prediction. World Electric Vehicle Journal, 15(4), 173.

[7]Silva, V., Siebra, C., & Subramanian, A. (2022). Intersections management for autonomous vehicles: a heuristic approach. Journal of Heuristics, 28(1), 1–21.

[8]Guo, J., Xie, Z., Liu, M., Dai, Z., Jiang, Y., Guo, J., & Xie, D. (2024). Spatio-Temporal Joint Optimization-Based Trajectory Planning Method for Autonomous Vehicles in Complex Urban Environments. Sensors, 24(14), 4685.