



Prifysgol Cymru
Y Drindod Dewi Sant
University of Wales
Trinity Saint David

**The Research of Location and Control Technology
in Automatic Vehicle**

by

Wang Cheng

Supervisor: Dr. Michael Dacey

Project submitted as part of the requirements for the award of
MSc Software Engineering and Artificial Intelligence

November 2022

DECLARATION

I, ...WANG CHENG..... declare that I am the sole author of this Project; that all references cited have been consulted; that I have conducted all work of which this is a record, and that the finished work lies within the prescribed word limits.

This work has not previously been accepted as part of any other degree submission.

Signed : WANG..CHENG.....

Date : 6/9/2022.....

FORM OF CONSENT

I WANG CHENG..... hereby consent that my Project, submitted in candidature for the M.Sc. Applied Computing degree, if successful, may be made available for inter-library loan or photocopying (subject to the law of copyright), and that the title and abstract may be made available to outside organizations.

Signed : WANG..CHENG.....

Date : 6/9/2022.....

ABSTRACT

As an important application of artificial intelligence technology in automobile industry, automatic driving technology is in a high-speed development period in China. Autonomous driving technology covers navigation and positioning, image processing, data communication, mechanical design and other multidisciplinary knowledge in different fields. Its core functions can be divided into five modules: perception, mapping and positioning, decision-making, planning and control. Positioning technology is a basic function of self-driving cars, and has huge requirements on the requirement of location. Only when the vehicle knows its precise location on the map can it drive correctly according to the planned path. As the last actuator of the whole automatic driving system, the control module determines the moving direction of the vehicle through lateral control, and realizes the acceleration and deceleration of the vehicle through longitudinal control, which is directly related to the realization effect of the whole set of automatic driving technology.

This design is a set of automatic driving tracking control system based on an electric vehicle with a drive-by-wire chassis. It uses the combination of real-time kinematic (RTK) GPS and the technology of simultaneous location and mapping (SLAM) to achieve vehicle positioning. The nonlinear model predictive control (NMPC) algorithm and the PID algorithm are used to carry out the lateral and vertical motion of the drive-by-wire chassis vehicle, respectively.

Automatic cars need to build an approximate model of their environment and have the ability to determine their own location. However, the traditional ways rely on the Global Navigation Satellite System (GNSS) and Global Positioning System (GPS), the function of position will work bad when the devices are disturbed by the weather or building. While, RTK-GPS can change these errors by the internal calibration signal of the detailed base station, but the signal of RTK-GPS will also be blocked in the basement or when there are tall buildings around the vehicle. Furthermore, SLAM technology is a good alternative when GPS signal is not available, it can provide real-time accurate positioning information for autonomous vehicles.

Model predictive control can achieve lateral control of the vehicle at different speeds through safety constraints. At the same time, it tends to further improve the safety of

the lateral control algorithm, the adaptive limit of the constraint value of the NMPC algorithm (the maximum steering angle of the vehicle) at different vehicle speeds is realized by combining with fuzzy control. In order to realize that the reference coordinates of the tracking trajectory can be input to the control algorithm synchronously with the movement of the vehicle, a trajectory input synchronization algorithm is designed. Finally, it tends to verify the feasibility of the designed control system, the control algorithm program was written and the hardware-in-the-loop (HIL) bench and real vehicle experiment were carried out. The results show that the reference trajectory coordinates can be updated with the change of the vehicle position, which improves the practicability of the control system; the maximum steering angle constraint threshold can be adaptively changed with the vehicle speed, which improves the control stability of the vehicle; For accurate tracking under working conditions, the overall maximum lateral tracking error is controlled within 10cm.

Key Words: autonomous driving; SLAM; NMPC; PID; Fuzzy control

TABLE OF CONTENTS

I. INTRODUCTION	1
II. LITERATURE REVIEW	3
A. THE CORE FUNCTION OF AUTOMATIC VEHICLES	3
B. THE TECHNICAL ARCHITECTURE OF SLAM.....	7
C. THE TECHNICAL ARCHITECTURE OF CONTROL.....	12
III. DESIGN AND IMPLEMENTATION	16
A. THE EXPERIMENTAL PLATFORM.....	16
B. POSITIONING SOLUTIONS AND TECHNOLOGY.....	17
C. LATERAL CONTROL SOLUTION.....	21
D. THE PRINCIPLE OF ADAPTIVE FUZZY CONTROL OF FRONT WHEEL ANGLE CONSTRAINT.....	23
E. VERTICAL CONTROL SOLUTION.....	24
F. CONTROL SYSTEM WORKING PRINCIPLE AND CIRCUIT DIAGRAM.....	26
G. SOFTWARE DESIGN AND PROCESS.....	27
IV. RESULTS AND DISCUSSION	32
A. STRAIGHT AND CURVED TEST RESULT.....	32
B. RESULTS OF DOUBLE LINE CHANGE TEST CONDITIONS.....	34
C. OPTIMIZATION RESULTS OF FRONT WHEEL ANGLE ADAPTIVE CONSTRAINTS.....	35
V. CONCLUSIONS AND RECOMMENDATIONS	38
A. CONCLUSION ABOUT THIS PROJECT.....	38
B. OUTLOOK FOR FUTURE WORK.....	38
VI. REFERENCES	40
VII. APPENDICES.....	47

I. INTRODUCTION

In 2021, vehicles production and sales in China have reached 26.082 million and 26.275 million, with an increase of 3.4% and 3.8% year-on-year respectively. The total automobile production and sales in China has ranked first in the world for 13 consecutive years. The market shares of new passenger cars equipped with combined driving assistance systems reaches 20% and more and more companies are investing a lot of R&D personnel and funds in the field of autonomous driving.

Society of Automotive Engineers (SAE) has divided the level of intelligent driving into six stages from L0 to L5. Among them, L0 refers to manual driving, L1 corresponds to the driving assistance stage, L2 and L3 are partial automatic driving and limited automatic driving, L4 is highly automatic driving, and L5 stage is fully automatic driving. Compared with Europe and the United States, China's autonomous driving technology started a step later. In the L2 and L3 stages, it partially lags behind the European and American regions. However, the domestic market has gradually surpassed the European and American regions and moved towards a leading position in the L4 stage. In recent years, domestic and foreign capital investment in autonomous driving has been increasing year by year. As shown in Figure 1, the amount of investment in the autonomous driving industry has soared to 65 billion from 2019 to 2021, and the number of investment in the market has also increased to more than 50 whose financing transactions worth more than 100 million. In 2020, the domestic autonomous driving market will reach 170.2 billion yuan, and in 2021 it will be as high as 235.8 billion yuan. It is estimated that by 2026, the market size will reach 773.5 billion yuan.

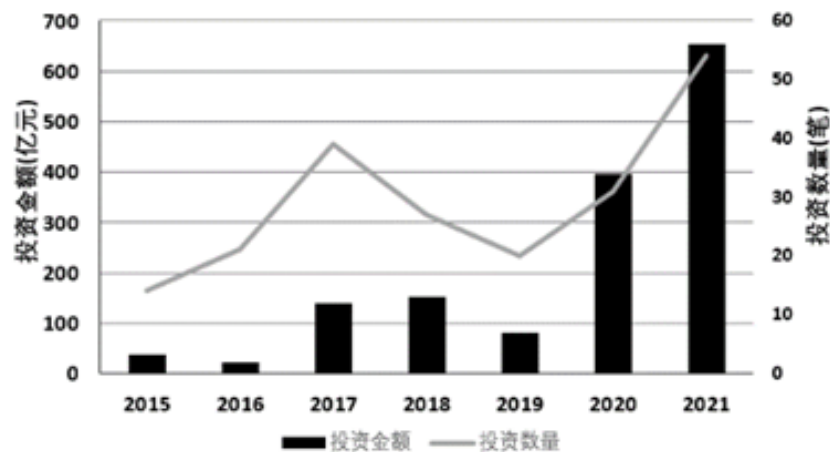


Figure 1 The changes of Autonomous driving financial investment market

The key technologies of autonomous driving involve many disciplines such as automobile, transportation, electronics, control, Internet of Vehicles, information security, machine vision, etc. The core functions can be divided into perception, positioning, decision-making and control. The perception function is mainly realized by multi-sensor fusion technology. Currently, the road environment is recognized by the combination of sensors such as Lidar, camera, millimeter wave and ultrasonic radar. High-precision maps and high-precision positioning technology are the basic key supporting technologies in the driving process, paving the way for the development of planning and decision-making functions. At the same time, the position information of the vehicle will be used by multiple modules in the automatic driving function at the same time. In a complex road environment, the perception module needs to locate the position of the vehicle body and the position of the obstacle in order to make sure the position of obstacles on the map. In the path planning process, the planning module also needs to know the current position and the next moving position, then planning a reasonable path between these two positions. If the position of the vehicle is not accurate enough, the planned driving trajectory is also inaccurate. The positioning function is like the human eye, which always needs to know where it is. However, the position function requires higher accuracy.



Figure 2 The receiver of GPS

Self-driving cars require an accurate model of their environment and the ability to determine their own location. Traditional positioning methods rely on Global Positioning Systems and Global Navigation Satellite System, However, the measurement error of GPS and GNSS usually reaches about 10 m because of factors such as signal reflection, clock deviation and atmospheric obstruction, especially when the vehicle is driving in

tunnels and mountain areas, the positioning signal will be lost which may lead to a serious accident. RTK-GPS can correct these errors through the internal calibration signal of the fixed base station, but the RTK-GPS signal will also be blocked in the basement or there are tall buildings around the vehicle. However, SLAM technology is a good alternative when GPS signal is not available, it is usually equipped with Lidar and repeatedly observes the landmarks in the environment through sensor information at different times estimating the state of the vehicle at these times based on the constraints of the observation, which includes pose and speed about the vehicle. Then, a map consistent with the environment is incrementally constructed according to the subject's own position, so as to achieve the purpose of simultaneous positioning and mapping, which is the main idea of SLAM. In this process, mapping and localization are mutually reinforcing. The real-time map provides environmental information for localization, and localization provides the coordinate reference for incremental mapping. The point cloud of SLAM is shown in figure 3.

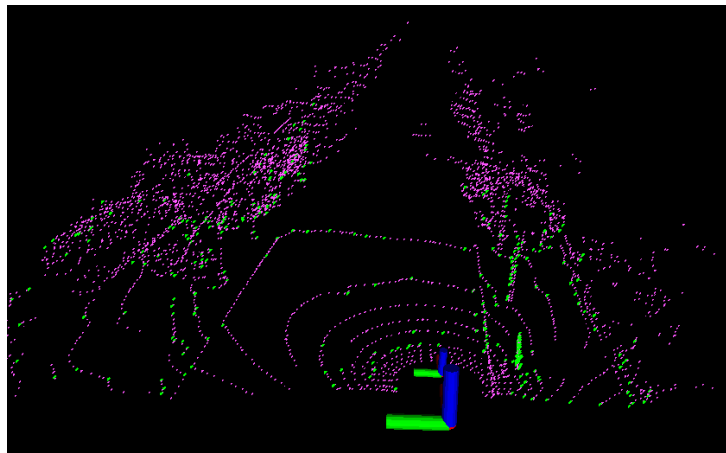


Figure 3 The point cloud of SLAM

As one of the most important functions in the automatic driving control module, the function of control mainly realizes the movement of the vehicle as close to the planned trajectory as possible, while ensuring the tracking accuracy and steering smoothness. If the vehicle motion is decoupled, it can be subdivided into lateral motion and longitudinal control. The lateral motion control is to realize the tracking of the desired path by the smart car, and the longitudinal motion control is to realize the follow-up of the smart car to the desired speed. Generally speaking, motion control should not only ensure the tracking accuracy, but also consider the driving safety and riding comfort of the smart car.

II.LITERATURE REVIEW

With the rapid economic and social development, the material life of the people has been greatly enriched, and the number of car ownership has increased year by year. As of 2020, the number of motor vehicles in China has exceeded 302 million. The vehicle can bring convenience to people in daily life, but also make many adverse effects, such as traffic accidents, traffic congestion and environmental pollution. In 2020, the number of people who died in traffic accidents in the world is as high as 1.27 million. At the same time, the number in China has also reached more than 200,000 casualties. Traffic accidents have become a major factor that threatens people's lives [1]. At present, autonomous driving technology is developing rapidly. Since the driver is partially or completely eliminated, autonomous vehicles have become a research and development hotspot to improve traffic efficiency, ensure road safety and realize the implementation of artificial intelligence in the automotive field. As the core of autonomous driving, autonomous driving technology involves multiple disciplines such as vehicles, computers, communications, and information. However, a complete set of autonomous driving systems includes four aspects: environmental perception, positioning, decision planning, and motion control [2].

A. The Core Function of Automatic Vehicles

The environmental perception system relies on in-vehicle sensors such as cameras and radars or Internet of Vehicles devices to detect the traffic environment information around the vehicle, and obtain the drivable area, traffic signs and obstacle information, etc., to provide the basis for subsequent decision-making and planning layers. The task of the decision-making planning system is to decide the next action of the autonomous driving vehicle according to the information of the perception layer and the current traffic environment, and plan the optimal trajectory for driving. The core is how to deal with it like a driver. complex and changeable traffic environment, and make safe and reasonable decisions. The main task of the motion control system is to enable the vehicle to follow the upper-level planned trajectory and speed by controlling the vehicle's drive train, brake train, and steering train, similar to how the driver manipulates the car to make the car drive as desired. At present, the selection of environmental perception sensor solutions includes pure vision sensors represented by

Tesla and vision sensors combined with Lidar, millimeter-wave radar, etc., represented by Waymo [3]. Target recognition and detection is an important part of the environment perception system of unmanned vehicles [4], and deep learning algorithms have attracted the attention of many researchers because of their stronger discriminative ability. Girshick [5] proposed the R-CNN algorithm, which creatively applies deep learning methods to target recognition based on candidate regions, but the algorithm has the shortcomings of not sharing features and long recognition time. In order to overcome the limitations of the R-CNN algorithm, the Faster R-CNN algorithm [6] was proposed, which truly realizes end-to-end detection and recognition, but has the shortcomings of low accuracy of small target recognition and complex model.

In terms of decision-making and planning, Xu [7] proposed a collaborative method for traffic signal optimization and vehicle speed control, which can simultaneously optimize traffic signal timing and vehicle speed trajectory, thereby improving traffic efficiency and reducing fuel consumption. The paper [8] proposed a fleet-based trajectory optimization method to help vehicles pass through intersections with maximum speed. The method transforms the trajectory optimization of the fleet into the trajectory optimization problem of the leading vehicle of the fleet through the gap feedback control system, thereby smoothing the vehicle trajectory to reduce fleet parking, saving fuel and alleviating traffic congestion. Xu Guoyan [9] used an improved deterministic policy gradient algorithm to define the input and output states of unmanned vehicles, discuss the constraints inside and outside the vehicle, and generate optimal decision-making for obstacle avoidance by designing static and dynamic obstacle environments for simulation training.

As early as the ICRA conference in 1986, the probability perspective of SLAM was first proposed, and the use of probability theory to study SLAM technology became the mainstream. During the same period, as researchers discussed at some academic conferences, environment-consistent mapping was established as a fundamental problem in robotics. Since then, some important work has gradually promoted the development of SLAM theory. Papers [10] and [11] apply statistical Bayes to describe the relationship between landmarks and geometric uncertainty. A core of this work is to show that there is a high degree of correlation, and this correlation increases with observations. In 1990, Smith and Cheeseman [12] pointed out that when a robot

moving in an unfamiliar environment makes relative observations of landmarks, since these observations have constraints on the vehicle position estimation, the estimation of these landmarks must be interrelated. This study reveals a profound truth that a complete solution to the SLAM problem requires a pair of combined quantities, the pose of the robot and the position of each landmark. And this means that we need to estimate a very long state vector (about as many as there are landmarks in the map), and also perform operations on the order of the square of the state vector. Due to the lack of analysis of the convergence of the map and the limitation of the computational complexity of the mapping process, the researchers tried to avoid filtering the entire system during the research process [13], [14]. As the understanding of the SLAM problem gradually deepens, people gradually realize that if they treat the SLAM problem as an estimation problem, the error will be convergent [15], [16]. In addition, the probability theory derivation method has been introduced into the research of SLAM, and the use of probability theory tools to solve the SLAM problem has achieved very good results, including extended Kalman filter, particle filter and maximum likelihood estimation, etc.

During the "Analysis of Algorithms" era (2004-2015), researchers conducted many studies on the fundamental properties of SLAM, including observability, convergence, and consistency. During this period, researchers understood the important role of sparse features in efficient SLAM solutions and developed major open source SLAM libraries. In the era of "integrated artificial intelligence" (2015-present), with the development of artificial intelligence technology, the application of artificial intelligence technology in the field of SLAM has achieved very significant results, which are extremely effective in dealing with dynamic objects in the environment, relocation and semantic maps. It has greatly promoted the development of SLAM technology.

At present, the development of SLAM has gradually formed two categories, namely Lidar SLAM and visual SLAM. Among them, compared with visual SLAM, Lidar SLAM can show better robustness to external factors such as illumination, texture and distance. In the field of automatic driving with very high safety requirements, Lidar SLAM is more robust than visual SLAM [17]. Therefore, this paper is mainly oriented to the research direction of Lidar SLAM.

B. The Technical Architecture of Slam

There are four important parts about the process of SLAM and Front-end odometer is the basic one. At first, the odometer can process sensor data to obtain measurement data that is effective for the environment. Secondly, it is responsible for associating sensor measurement data with specific landmarks in the environment, which is called data association. Then, the motion state of the carrier is estimated by observing data of multiple frames of road marking points in the running process. In the end, the front end will provide the initial value of nonlinear optimization for the operating state variables of the carrier. The Lidar odometer estimates the motion state of the carrier by matching the point cloud of the current frame to the point cloud of the reference frame by means of Scan-Matching.

The most classic method based on point correspondence is the iterative closest point(ICP) method, as for any point in the current frame, the ICP algorithm mainly finds the closest point in the reference frame for matching according to the Euclidean distance, establishes the error function and calculates the Jacobian matrix according to the distance of the matching point, and then uses nonlinear optimization to calculate the Pose transform that aligns the reference frame with the current frame. Trimmed-ICP proposed in [18] is an extension of ICP that combines the Least Trimmed Squares (LTS) method, enabling the algorithm to operate in point clouds with overlapping degrees below 50%. The "point-to-surface" matching proposed in the paper [19] is also an improved method of ICP, which improves the performance of matching optimization by detecting planes and utilizing the normal information of plane surfaces. The GICP algorithm proposed in the article [20] adds "point-to-point" matching on top of the "point-to-point" matching method of ICP, and combines the matching errors of "point-to-point" and "point-to-point" in an optimization framework to estimate the position. In fact, GICP is more robust to mismatches. The IMLS-SLAM proposed in [21] preprocesses dynamic objects and employs a sampling strategy on Lidar frames, using surface features of implicit moving least squares (IMLS) to define the model from previous Lidar frames. The Semantic ICP proposed in [22] combines geometric and semantic inference to improve the registration task by incorporating pixelated semantic measures into estimating the relative transformation between two point clouds, where point associations are treated as potentially random variable, which leads to an Expectation-

Maximization (EM) solution. Semantic ICP outperforms GICP due to the combined use of semantic labels and EM data association. Compared with all point matching of ICP and its derived methods, the matching method based on feature points uses the extracted feature points to complete the matching. Under the condition of ensuring the matching accuracy, the calculation efficiency of point cloud matching is greatly improved, so that Lidar SLAM is able to run on embedded devices, so feature matching based methods are gradually becoming a more popular alternative. The paper [23] proposes a plane-based real-time Lidar odometer registration method. Assuming operating in a structured environment, it extracts planes from point clouds and matches them by solving a least squares problem. The article [24] proposes a lead-line-based mileage estimation method. This method randomly generates line segments from the original point cloud for registration. The point cloud feature extraction and matching algorithm applied in the LOAM framework proposed by the article [25] is the most classic and most commonly used method at present. It first calculates the curvature of each point, and the calculation of the curvature is obtained by calculating the depth values of the point and its nearby points. Arrange the point clouds in the same column according to the size of the curvature, extract the maximum value in the arrangement as the edge point, and extract the minimum value as the plane point. After that, the sum of the distances from all edge points to the line and the distances from the plane points to the surface is minimized to form an optimization problem, which is solved using a nonlinear optimization method. The Lego-loam proposed in the article [26] is optimized on the basis of LOAM. Firstly, point cloud segmentation is used to identify the ground and filter out abnormal points. Then, a two-step matching optimization is performed using the LM to estimate the 6-DOF transformation between consecutive frames: the first step is to use the ground point cloud for matching; the second step is to match the edge points and plane points in the segmented point cloud. Since the segmentation step of Lego-loam reduces the difficulty of feature point matching, Lego-loam can achieve a 35% reduction in running time with the same accuracy as LOAM. The paper [27] proposes a novel feature extraction method for Lidar data that requires no prior knowledge and is widely applicable to various environments to solve the feature extraction problem in Simultaneous Localization and Map Creation (SLAM). This method uses the classic image feature extraction method - Harris corner detector, specifically,

multi-scale Kanade-Tomasi corner detector to extract features. This method can extract stable and accurate feature points from measurement data of various scales, and can obtain feature point descriptors and uncertainty information at the same time. The article [28] proposes a multi-modal feature hybrid association algorithm based on a multi-hypothesis framework, and analyzes the constraints on the robot pose solution for multiple modal features. Robot pose estimation algorithm. The MULLS algorithm proposed by the article [29] divides the point cloud into six categories by more detailed division, and then corresponds to three types of residuals. Different residuals have different weights according to the performance of different point clouds, forming a comparison. Efficient optimization of multiple residual scales for estimating the front end.

The method based on point matching usually relies on providing a good initial value, and at the same time, due to the defects of the algorithm itself, the final iterative result is very easy to fall into the local optimum. Therefore, the article [30] proposes another matching algorithm, a matching algorithm based on The Normal Distributions Transform (NDT). The normal distribution is then compared with other normal distributions for other point clouds, and the assigned score is calculated, and finally the rotation and translation at which the score is maximal are found by a nonlinear optimization algorithm. Because it does not use the feature calculation and matching of corresponding points in the registration process, the calculation time is relatively fast, and the dependence on good initial values is small, and large initial value errors can be corrected. The transformation method based on the normal distribution has also become a common method to solve the registration problem in Lidar odometer applications, and in the past two decades, a variety of improved versions of NDT have been proposed [31] – [34].

The second part of SLAM is backend optimization which belongs to the higher level compared with the front-end odometer. The calculation of the front-end odometer is derived from sensor data which will inevitably have noise, and since the front-end is an incremental odometer, the influence of noise will not accumulate. Therefore, the main task of the back-end is to use the probabilistic model to uniformly filter the front-end odometer information and estimate the state of the entire system from the noisy data. The back-end is the core part of SLAM technology and its development is closely related

to the development of the back-end. The back-end mainly goes through three stages, namely Gaussian filtering, non-parametric filtering and methods based on nonlinear optimization.

As for the Gaussian filtering, the backend was mainly modeled with a non-linear Gaussian model, with early work by Ayache and Faugeras [35] and Crowley [36], Chatila and Laumond [37] using Kalman filter-like algorithms for mobile robot navigation. In addition, EKF requires explicit motion and observation models and linear assumptions for the models. Therefore, S.J.Julier and J.K.Uhlmann [38] investigated the performance impact of nonlinear models on the application of EKF. The article [39] builds two laser odometer based on PL-ICP and NDT point cloud matching algorithms to correct and fuse the two odometer information based on the idea of Kalman filtering, it can construct a two-dimensional and single-line laser odometer suitable for small scenes. Experiments show that the odometer has higher location correctness and better robustness in small scenes with relatively complete point cloud shapes. Facing to the non-parametric filtering, the papers [40] – [43] has made effort to explore and [44] proposed a SLAM algorithm based on “Rao-Blackwellised” particle filter to solve the SLAM problem of robots in large-scale environments. By selecting stable and easily distinguishable feature points, a data association method based on global constraints is developed, which effectively reduces the probability of mismatching; the improved particle distribution prediction function is used to improve the properties of the particle filter. The experimental results indicate that the algorithm has low computational complexity and high precision, and can effectively solve the robot SLAM problem in large-scale environments. The paper [45] proposes a simultaneous localization and mapping method for mobile robots based on an improved particle filter. This method organically combines conventional particle filter and particle swarm optimization algorithm, and introduces the latest robot observation information to adjust the proposed distribution of particles, so as to ensure the accuracy of the algorithm, reduce the number of particles required for positioning and mapping, and effectively alleviate the Particle Degradation Phenomenon. In addition, considering that the conventional resampling process is easy to cause sample depletion, a probability operator is introduced to increase the diversity of particles. The results show the advantage of this method.

At present, the back-end based on nonlinear optimization generally adopts the method of factor graph optimization [46]. Unlike graph optimization and pose graph optimization, factor graph optimization is a probabilistic graphical model based on the theory of Maximum a posteriori (MAP) estimation. Many works [47–52] discuss factor graph SLAM systems, focusing on improving the effectiveness and robustness of optimization under SLAM problems.

The third part of SLAM technology is loopback detection, which provides loopback information for SLAM system for global consistency optimization in the backend. SLAM systems are still inevitably affected by accumulated drift in long-term operation because it lacks of global information. The purpose of loop closure detection is to eliminate the accumulated error of the odometer. In practice and research, descriptor based detection methods are usually used. It extracts descriptors from sensor data, and determines the loop closure detection relationship based on the descriptors of the two frames. This approach gets rid of accumulated errors, making the loopback detection module a relatively independent module in the SLAM system.



Figure 4 The global map

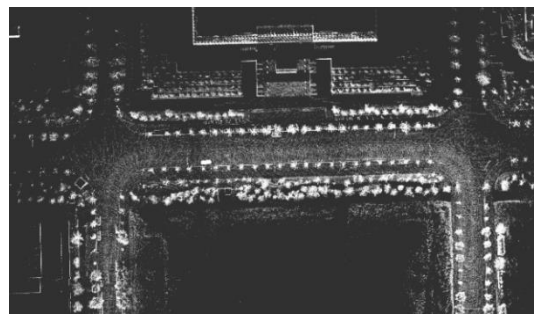


Figure 5 The local map

Mapping which is one of the two major goals of is the last part. In the classic SLAM model, the map is a collection of all feature points. Once the location of the landmark points is determined, the mapping is completed. The result of SLAM is shown in figure 4 and the local map is shown in figure 5. With the enrichment of upper-layer applications, the requirements for mapping continue to expand. At present, there are roughly three types of maps in the SLAM field: Metric Maps, Topological Maps and Semantic Maps. In dense models, primitive representations describe 3D geometries through a large collection of unstructured points (i.e. point clouds) or polygons [53]. Point cloud information from binoculars, RGB-D cameras, and 3D Lidar are widely used in robotics [54]. Dense maps have recently gained popularity in monocular SLAM and are combined

with the use of direct methods [55] to estimate robot trajectories and 3D models directly from the intensity values of all image pixels.

C. The Technical Architecture of Control

As one of the most important functions in the automatic driving control module, path tracking mainly realizes the control of the vehicle as close to the planned trajectory route as possible, and also ensures the tracking accuracy and steering smoothness [56]. At present, the commonly used algorithms for path tracking control include pure tracking algorithm, PID control algorithm, LQR control algorithm, Stanley algorithm and MPC algorithm.

The pure tracking algorithm has been researched for many years in the path tracking of autonomous vehicles, and it has been widely used because of its simplicity and ease of use. The pure tracking algorithm is a calculation method based on geometric principles. It can be used to calculate the arc-shaped trajectory that the vehicle travels from the current position to the target point in motion. There is a one-to-one correspondence between the vehicle turning radius and the steering wheel angle, and does not require Considering the vehicle dynamics model, the selection of the forward looking distance has a great influence on the path tracking accuracy, so it is widely used in low-speed scenarios [57]. Stanley control, also known as front wheel feedback control, its core idea is to calculate the steering angle based on the path tracking deviation [58], which is a pure geometric tracking method based on kinematics model. Compared with pure tracking algorithm, Stanley control also considers the influence of heading angle deviation by adjusting the steering angle of the front wheel to eliminate the influence of lateral deviation and heading angle deviation. PID control has the advantages of simple control principle and easy implementation. It is often used in the path following control of mobile equipment. [59] adopts the method of combining PID control and neural network to dynamically adjust the control parameters which can improve the stability of the motion of the control system. [60] proposed an integral-separated PID control algorithm in which α and β are multiplied before the proportional term and the integral term, which improves the response speed of the system and enhances the robustness of the system. [61] analyzed the dynamic characteristics of the mobile robot and designed a PID controller method of direct feedback linearization. The method is

simple and easy to implement, with high tracking accuracy and good anti-interference ability. [62] put forward a way to code the individual by genetic algorithm, using the method of changing the fitness function of the individual in the genetic algorithm to the PID control accuracy of the mobile robot to adjust the PID parameters, which improves the robot's self-adjustment ability and reduces the tracking error. In high-precision servo control, the tracking performance of the system can be improved by means of feedforward control. [63] established a robot kinematic model and designed a double closed-loop proportion & feedforward controller. The simulation result shows that this method can reduce the average tracking error compared with the path following ability of the classical PID control.

The Linear Quadratic Regular (LQR) controller is a modern control method that uses state space equations for control. It constructs a quadratic minimum cost function through a fixed weighting matrix to solve the linear system based on the mathematical model of the vehicle in one control cycle and works out the output of the steering angle. In the literature [64], the preview strategy and the linear quadratic optimal control (LQR) theory are used to design the lateral motion controller of the autonomous vehicle, the selection of the preview distance and the optimal controller weighting matrix were analyzed in detail. Finally, the effectiveness of the controller was verified through simulation and tests. [65] studies the lateral motion control problem for four-wheel steering vehicles which first establishes vehicle dynamics and a path tracking error model based on a preview strategy. After combining this two models, a LQR-based four-wheel vehicle is designed. In [66], the application of optimal control in the lateral control of intelligent vehicles has been studied. The LQR controllers in the above studies all use a fixed weighted matrix, which greatly limits the adaptability and flexibility of the controller. Therefore, the literature [67] designed an improved LQR controller with self-adjustable weighting matrix. Combined with feedforward control, it realized the lateral motion control of smart cars. In this paper, the path tracking error mechanics model of the basic non-aiming strategy is first established, and on this basis, the lateral feedforward and the LQR feedback controller are designed. Furthermore, a weighted matrix adaptive adjustment strategy which includes the path tracking deviation and vehicle-road position is designed to improve the control adaptability and accuracy.

Model predictive control originated from predictive algorithms in industrial control, such as Cutler's dynamic matrix control (DMC) based on linear superposition conditions [68] and Richlet's model predictive heuristic control (MPHC) [69]. These prediction algorithms do not require precise model parameters and it can estimate the state of the system by simply collecting the system impulse response or step response. Using the optimization idea of modern control theory, the optimal value is obtained at each sampling time, then the model is fed back and compensated according to the real-time state information of the system. Reference [70] pointed out that these characteristics make it avoid the difficulty of identifying and minimizing the parameter model, which also reduce the real-time calculation amount of online optimization and improve the robustness of the control system. It is suitable for the actual needs of the industrial control process.

The MPC algorithm is more and more widely used in the field of path tracking. Leman designed a nonlinear MPC path tracking controller to avoid obstacles while tracking a reference path [71]. Katriniok proposed an MPC path tracking algorithm based on a linear time-varying model, and designed a state estimator to ensure that the control behavior would not violate the vehicle dynamics constraints [72]. Yang Kaiming proposed an LTV-MPC path tracking algorithm based on steering torque input, which can combine the electric power steering system with the vehicle model, so as to realize automatic driving by controlling the motor torque, which ensures the tracking accuracy and improves the stability [73]. Sun Chuanyang proposed a MPC path tracking controller that switches the tracking error, which is based on the switching between steady-state sideslip and transient sideslip, it will calculate the heading deviation and the vehicle can have better tracking performance [74].

The Model Predictive Control (MPC) algorithm includes three key steps: predictive model, rolling optimization and feedback correction. It refers to predicting the future output of the system according to the current state of the system and future control quantities by using the model of the target control object, and Combined with the actual road and the constraints of the control amount, it is optimized to solve the control amount at the future time. Using the MPC algorithm to solve the trajectory tracking control problem of intelligent cars should select the modes firstly. The vehicle model is mainly composed of the chassis model and the tire model. This model design

must follow the specific requirements and basic rules. As for this algorithm, the vehicle model tends to predict the state of the vehicle over a period of time in the future which is based on current and historical state. The MPC algorithm belongs to a kind of optimal control whose goal is to find the optimal strategy under the given conditions. The basis for judging the performance is called the performance index function or objective function. The objective function should include the future vehicle state and control input. The trajectory tracking model prediction control is at any sampling time, and the finite time period of the objective function is within the prediction time domain. The solution process is an open-loop optimization problem within a finite time length, and then the optimal control sequence in the prediction time domain is solved. Different from the traditional optimization problem, the MPC algorithm does not execute the optimal control sequence one by one, but only enables one control variable at the current time point, that is, the first control variable. When the next sampling time is reached, the prediction time domain is moved forward by one step, and then the optimal control sequence is re-calculated according to the current time domain. The objective function of the MPC algorithm is for the whole in form, but the specific content is for different moments.

Establishing an accurate vehicle model is very difficult and has no practical significance. The vehicle models used in engineering applications are often simplified, and it is inevitable to calculate the model errors. However, this error can be eliminated or its influence can be reduced by rolling optimization and the generation of error can also be reduced by feedback correction. On the basis of the traditional MPC control algorithm, there are a lot of more detailed studies. In the literature [75], P. Falcone designed the vehicle active steering control system based on the model predictive control method, in which the prediction model of the controller selected the vehicle three-degree-of-freedom model. In order to reduce the amount of calculation, the author uses the continuous online linearization method to process the vehicle model, calculates the ideal wheel angle by solving the linear optimization problem. Finally, it compares and analyzes the experimental results of linear and nonlinear control. Paper [76] studied the influence of the reference path deviation and the path curvature weight on the path tracking accuracy of the MPC algorithm. When the actual path changes the weight coefficient can be adaptively changed to improve the tracking accuracy.

III. DESIGN AND IMPLEMENTATION

A. The experimental platform

This project takes the wire-controlled electric car as the control object. At first, we need to determine the basic physical parameters of the vehicle, and determine the feasible lateral control scheme and longitudinal control scheme according to the CAN protocol of the trolley-by-wire control chassis, combined with the test function and objective. After verifying the first control scheme in the simulation software, determining the feasibility of the scheme, combined with the software operation environment of the actual vehicle, the control algorithm is transplanted to the HIL bench to further verify the reliability of the control scheme, and then the test verification of the actual vehicle is carried out.



Figure 6 The electric vehicle of wire-controlled chassis

On the prescribed test section, the trajectory point to be followed is first collected with GPS equipment, and the X, Y coordinates and heading angles under the corresponding geodetic coordinate system are recorded as data. Then, we should place the vehicle anywhere on the trail trajectory, ensuring that the front direction of the vehicle is consistent with the direction in which the vehicle is recording the map point. After the automatic driving track function is enabled, The GPS receiver can receive the positioning information from the RTK-GPS. If the positioning signal is not good, the positioning system will automatically switch to the SLAM module and receive the positioning data from it. At the moment, the lateral control algorithm will adjust the front steering wheel angle according to the lateral deviation of the vehicle position and the trajectory point; The longitudinal control algorithm controls the acceleration and deceleration of the vehicle to the target speed and continues to drive at a uniform speed.

B. Positioning solutions and technology

The RTK GPS positioning technology is used to dynamically locate the position of the autonomous vehicle on the road. The system composition usually includes three parts: a reference station, a mobile station (vehicle) and a communication unit. The reference station is usually set on a building near the road or on a roadside pole, the mobile station is installed on the vehicle, and the communication unit is used to realize data communication between the reference station and the mobile station, as shown in Figure 1. The base station receives GPS satellite signals in real time, and its pseudo-range carrier phase observations are shown in formula (1), and the observations are sent in the form of broadcast; the real vehicle receives GPS satellite signals in real time, and its pseudo-range carrier phase observations are shown in Formula (2). At the same time, the carrier phase data sent by the base station is received, and the group difference is calculated.

$$\rho_r^i = \lambda(N_{r,o}^i + N_r^i) + \phi_r^i \quad (1)$$

$$\rho_p^i = \lambda(N_{p,o}^i + N_p^i) + \phi_p^i \quad (2)$$

In the formula above, $N_{r,o}^i, N_{p,o}^i$ are the fuzzy number of the whole cycle; ρ_r^i, ρ_p^i are the cumulative number of the whole cycle from the initial epoch to the observation epoch; ϕ_r^i, ϕ_p^i are the fractional part of the measured phase; λ is the carrier wavelength, L_1 is the carrier wavelength is 19 cm, L_2 The carrier wavelength is 24 cm.

When using RTK GPS positioning technology to accurately measure the location of the test vehicle, the correct solution of the fuzzy number in the whole circle is a necessary condition to realize the high-precision location of the test car. The integer least-squares theory is used to solve the integer fuzzy numbers, that is, the floating-point solution is first obtained by the least squares method, and then the integer-cycle fuzzy numbers are obtained by the fast search algorithm under the integer constraints, and finally the fixed solution is obtained. In the development of this road test system, it tends to realize the real-time positioning about this test vehicle during the driving process, the combined wave-width lane processing technology is used to solve the fuzzy number of the whole cycle, and the dual-frequency GPS receiver is used during operation, and the phase difference measurement of the conventional code carrier is used to obtain an approximate estimate of the ambiguity of the whole cycle, then smooth the pseudo-range noise and multipath effect, reduce the number of searches for the number of

whole cycles, and finally pass the linear combination of the two wavelengths (lane width) according to the dual-frequency signal and the lane width of a wide lane can be obtained.

When the RTK GPS positioning system is working, the base station needs to transmit the observed carrier phase information to the mobile station on the test vehicle in real time. In order to ensure the high precision of the test vehicle positioning information, the bit error rate of data transmission is required to be less than 10 seconds. The update time of differential information is not higher than 10 seconds. This places high demands on the wireless communication unit between the base station and the test vehicle.

When the signal of RTK GPS is disturbed, the positioning system of autonomous driving will automatically switch to the positioning module provided by SLAM, and SLAM will continue to provide position information for the vehicle.

The sensor kit used in this design includes an OUSTER 32-line Lidar and differential GPS as the signal input to the system. And the system is equipped with differential GPS as a control group to provide real values for the verification of system performance. Among them, the data of A is the roads of the campus of Wuhan University of Technology. The detailed information of this dataset is shown in Table 1. Tests are run on paths that contain loopbacks, giving the system a loopback factor. Manually delete GPS data on a large scale to simulate the loss of GPS data. All data are collected on-site and mapped offline.

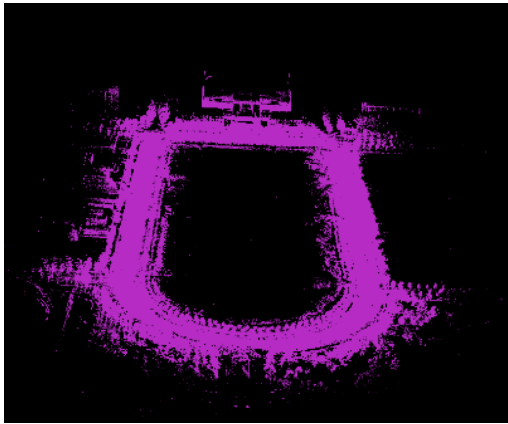
Table 1 Mapping dataset

Name	number of radar frames	Altitude change	total mileage
data	2476	8m	1800m

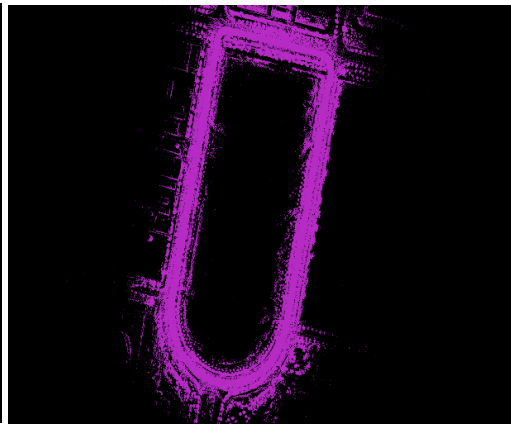
Firstly, we should power on and connect the domain controller, Lidar and GPS, and connect an external display screen for on-site operation and screen display. The specific operation steps are shown in the appendices.

In order to collect the data set, the test drove an electric car on the main road of the campus of Wuhan University of Technology, and returned to the same location after driving a circle. This test includes a loopback path to provide a loopback factor for the system to optimize global consistency. The received GPS data is only used during initialization to solve the vehicle attitude and align the body coordinate system to the

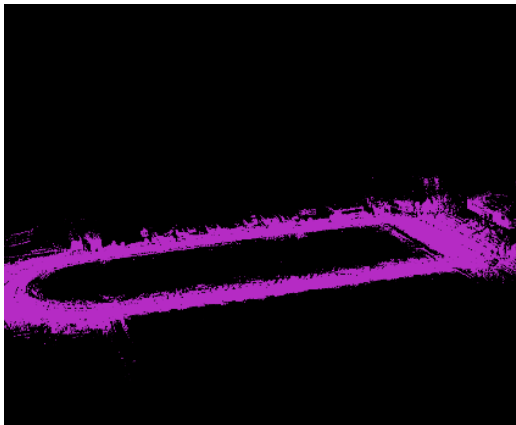
world coordinate system. The result of using this system to build a map is shown in the following figure.



a. front view



b. top view



c. side view

Figure 7 the map of slam

The team uses the NDT positioning algorithm to achieve real-time positioning on the established high-precision map. In order to verify the superiority of the positioning accuracy of this system, in the same scene, the value of the differential GPS is used as the comparison value, and the positioning results are compared with the differential GPS trajectory. The actual mapping scene is shown in Figure 8.

During the positioning process, the RTK GPS data received and the actual position data are drawn separately. It is clear to find that the positioning track is basically the same as the GPS track, which proves that the SLAM technology can replace the GPS to complete the function about location when the GPS signal or GPS signal is poor or missed.



Figure 8 Mapping scene satellite map

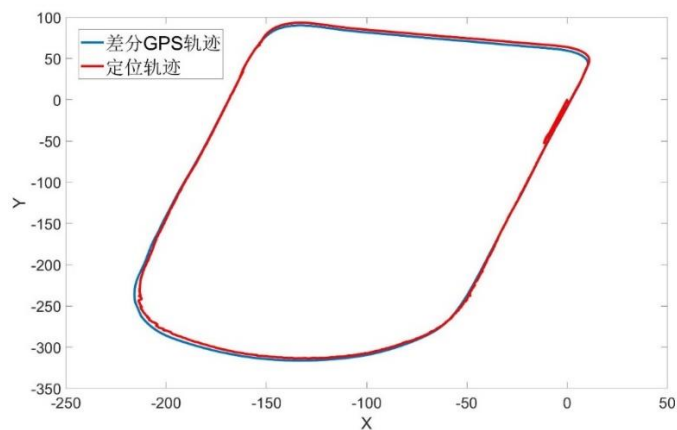


Figure 9 GPS track and positioning track

As can be seen from the above figure, the trajectory of the NDT algorithm is stable and accurate. According to the calculation, the horizontal error and rotation error of the NDT algorithm have been kept within a small range. On the whole, the high-precision pose estimation based on the NDT algorithm shows good performance, and a positioning trajectory with high accuracy can be obtained. It offers the guarantee about the positioning data and the control part can receive it.

C. Lateral Control Solution

Lateral control in automatic driving technology means that the controller turns the steering wheel instead of the driver according to the processing of the algorithm to complete the steering action of the vehicle. At present, the commonly used algorithms for track control are pure tracking algorithm, PID, LQR and NMPC algorithm. Pure tracking algorithm is a calculation method based on geometric principles which does not need to consider the vehicle dynamics model. However, the choice of pre-aim distance has a great impact on the accuracy of path tracing and pure tracking algorithm is widely used in low-speed scenarios. PID control algorithm is mainly used for dealing with the vehicle posture information with the feedback to do deviation processing. It requires a large number of experimental tests to calibrate the parameters and the anti-interference ability is poor.

LQR control algorithm has high requirements on the accuracy of the mathematical model, mainly for the optimization method of linear system without constraints, and the stability is poor in the case of interference, which is not suitable for medium and high-speed scenarios. Model predictive control algorithm includes three key steps: prediction model, rolling optimization and feedback correction, which refers to the use of the target control object model, according to the current state of the system and the future control amount to predict the future output of the system, and combined with the actual road as well as control quantity constraints to optimize the solution of the control amount at the future moment. The process can constantly reduce the lateral error with the reference path.

This scheme considers the movement of three-degrees freedom of longitudinal, transverse and horizontal movement. The target vehicle changes the direction by front wheels and the bicycle model that meets the above conditions is shown in Figure 10. The coordinate system $o\text{-}xyz$ is a vehicle coordinate system fixed to the center of mass of the body, the x -axis direction is along the longitudinal axis of the vehicle, the y -axis is perpendicular to the vertical axis of the vehicle, the z -axis is perpendicular to the oxy upwards, and the coordinate system OXY is the inertial coordinate system fixed to the ground. The vehicle dynamics equation is analyzed as a prediction model, and the forward wheel steering angle is output after processing by inputting parameters such as longitudinal and tangential speed, heading angle and heading angle rate.

In order to design a linear time-varying model predictive controller for tracking driving, the nonlinear vehicle dynamics model needs to be linearized, and the approximate linearization method is relatively simple and has strong applicability. According to the known vehicle dynamics equation, the selected state quantity is $\chi = [\dot{y}, \dot{x}, \varphi, \dot{\varphi}, Y, X]^T$, the control quantity is $u = \delta_f$, and the corresponding state quantity and control quantity satisfy $\dot{\chi} = f(\chi, u)$.

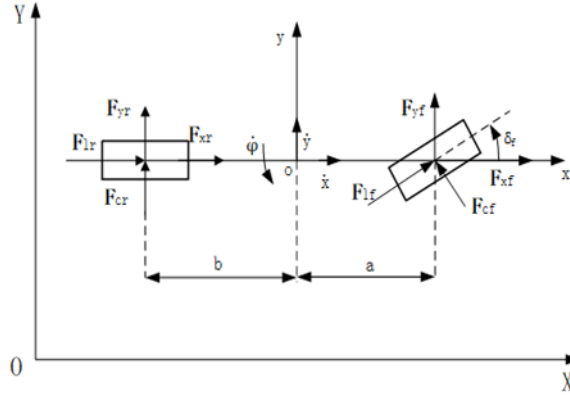


Figure 10 the bicycle model

$$\begin{bmatrix} \ddot{y} \\ \ddot{x} \\ \ddot{\varphi} \\ \dot{\varphi} \\ \dot{Y} \\ \dot{X} \end{bmatrix} = \begin{bmatrix} -\dot{x}\dot{\varphi} + \frac{2}{m} \left[C_{cf} \left(\delta_f - \frac{\dot{y} + a\dot{\varphi}}{\dot{x}} \right) + C_{cr} \frac{b\dot{\varphi} - \dot{y}}{\dot{x}} \right] \\ \dot{y}\dot{\varphi} + \frac{2}{m} \left[C_{lf} s_f + C_{cf} \left(\delta_f - \frac{\dot{y} + a\dot{\varphi}}{\dot{x}} \right) \delta_f + C_{lr} s_r \right] \\ \dot{\varphi} \\ \frac{2}{I_z} \left[aC_{cf} \left(\delta_f - \frac{\dot{y} + a\dot{\varphi}}{\dot{x}} \right) - bC_{cr} \frac{b\dot{\varphi} - \dot{y}}{\dot{x}} \right] \\ \dot{x} \sin \varphi + \dot{y} \cos \varphi \\ \dot{x} \cos \varphi - \dot{y} \sin \varphi \end{bmatrix} \quad (3)$$

In the formula (1), m is the curb mass of the vehicle, a is the distance from the vehicle centroid to the front axle, b is the distance from the centroid to the rear axle, φ is the vehicle heading angle, and I_z is the moment of inertia of the vehicle around the z-axis. F_{xf} and F_{xr} are the forces on the x-axis of the front and rear wheels respectively in the vehicle coordinate system, and F_{yf} and F_{yr} are the forces on the y-axis of the front and rear wheels. F_{lf} and F_{lr} are the portrait forces received by the front and rear wheels respectively, F_{cf} and F_{cr} are the lateral forces received by the front and rear wheels, and δ_f is the deflection angle of the front wheels.

The nonlinear model is Taylor-expanded at any point, and only the first-order term is retained to obtain a linear time-varying model. Finally, in order to complete the control of the vehicle, approximate discretization processing is required:

$$\chi(k+1) = A(k)\chi(k) + B(k)u(k) + d(k) \quad (4)$$

By constructing the objective function based on the lateral deviation, heading angle error and control increment, and converting the optimization objective into a quadratic programming problem, the solution of the control increment Δu is realized under the constraints of constraints:

$$J = [Y - Y_{ref}]^T Q [Y - Y_{ref}] + \Delta u^T R \Delta u + \varepsilon^T \rho \varepsilon \quad (5)$$

In the above formula, Y_{ref} is the reference trajectory containing the longitudinal position and heading angle information, Y is the output based on the target state quantity in the prediction time domain, Q and R are the weight matrix, ρ is the weight coefficient, and ε is the relaxation factor. In the model predictive controller, the constraints of adding the control variable u , the control variable increment Δu and the upper and lower boundaries of the actual portrait position y of the vehicle will be considered at the same time, as follows:

$$\begin{aligned} \Delta u_{min} &\leq \Delta u_t \leq \Delta u_{max} \\ -u_{max} &\leq u_t + A_I \Delta u_t \leq u_{max} \\ y_{min} &\leq y \leq y_{max} \end{aligned} \quad (6)$$

D. The Principle of Adaptive Fuzzy Control of Front Wheel Angle Constraint

Aiming at the relationship between the vehicle front wheel angle σ_f and the speed V in this study, a fuzzy controller can be designed for the vehicle speed and the front wheel angle. According to the speed requirements of different working conditions, the front steering angle that meet the vehicle safety and dynamic comfort can be calculated respectively. The program can turn the angle value firstly and take this angle as the maximum value u_{max} constrained by the model predictive controller to the front wheel turning angle. Then it will constrain the selection of the final control variable by the objective function, and continuously control the vehicle to accurately complete the steering.

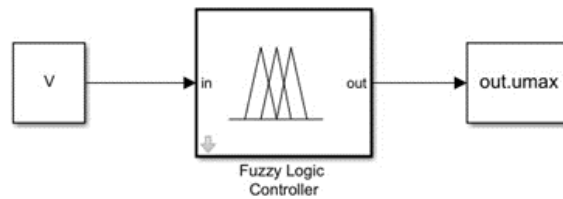


Figure 11 Design of Fuzzy Controller

When studying the turning condition, high speed will directly lead to higher lateral acceleration, which will seriously affect the ride comfort and the dynamic safety of the vehicle. Therefore, this study considers setting the domain of discourse for the input variable under turning conditions as [0 36], limiting the maximum vehicle speed to 36km/h, and selecting three triangular membership functions to study vehicle speeds of 10km/h, 20km/h and 30km/h are allocated to the low, medium and high domains of discourse, respectively. Different turning radius have different requirements for the turning angle threshold of the front wheel of the vehicle. When it is smaller than the turning angle threshold, the physical limit of the turning of the vehicle will directly affect the trajectory tracking accuracy. Therefore, the scope of the output variable u in this study is [6 15], that is, the front The rotation angle ranges from 6 degrees to 15 degrees, and the corresponding relationship is shown in Table 1. The number of membership functions of the output variables is 3, and the fuzzy control rules are shown in Table 2.

Table 2 Fuzzy Subset of Turning Cases

variable	L	M	H
scope	[0 10 12]	[12 20 25]	[25 30 36]
variable	S	MO	B
scope	[6 6.5 6.9]	[6.9 7.6 8.5]	[8.5 10.5 15]

E. Vertical Control Solution

The function of longitudinal control is to replace the driver's control of the acceleration and brake pedals through the control algorithm according to the target speed planned by the automatic driving, so as to realize the control of the vehicle speed by the vehicle. This scheme designs the upper-level controller according to the incremental PID theory. Since the reference trajectory point is composed of 150 discrete data points, the discrete incremental PID is used to calculate the expected acceleration in each cycle. It will calculate the increment of the control amount, the determination of the control increment is only related to the last three sampling values, and the accumulation of control errors is small. The formula is as follows:

$$e(k) = v_{des}(k) - v(k) \quad (7)$$

$$\Delta u(k) = K_P[e(k) - e(k-1)] + K_I e(k) + K_D[e(k) - 2e(k-1) + e(k-2)] \quad (8)$$

In the above formula, V_k is the actual vehicle speed during the V th control; $V_{des}(k)$ is the expected vehicle speed during the v -th control; $e(k)$ is the speed deviation. K_P , K_I and

K_D represent the proportional, integral and differential control coefficients of the PID controller, respectively. The system also saves the last PID output, so the PID output of this control cycle can be expressed as:

$$u(k)=u(k-1)+\Delta u(k) \quad (9)$$

u_k is the expected acceleration output by the upper-level controller.

During the driving process of the vehicle, the vehicle only selects one of the driving mode and the braking mode, so the lower-layer controller needs to select the vehicle mode according to the expected acceleration output by the upper-layer controller. In order to prevent poor comfort caused by frequent switching of driving and braking during vehicle driving, a dead zone range is set for the desired acceleration $[-0.1m/s^2, 0.1m/s^2]$. When the desired acceleration is within the dead-band range, the vehicle does not switch between drive and brake control. When the expected acceleration is greater than 0.1, the vehicle is in a driving state and accelerates, and when the expected acceleration is less than -0.1, the vehicle is in a braking state and decelerates to brake.

After obtaining the desired acceleration, the lower-level controller calculates the actuator control quantities according to the inverse longitudinal vehicle model. According to Newton's second law and the definition of air resistance and rolling resistance, the longitudinal dynamics model can be expressed as the following formula:

$$ma_{des} = F_t - F_b - \frac{1}{2}C_D A \rho v^2 - mgf_i \quad (10)$$

F_t represents driving force; F_b represents braking force, a_{des} represents expected acceleration; C_D represents air resistance coefficient; A represents frontal area; f represents rolling resistance coefficient; ρ represents air density; i represents slope; m represents vehicle mass; g stands for gravitational acceleration; v stands for vehicle speed. When the vehicle is in a driving state, the control amount of the lower controller is the desired accelerator opening. At this time, the braking force and the driving force F_t can be expressed as:

$$F_t = \frac{T_m i \eta}{r} \quad (11)$$

In the above formula: T_m is the motor output torque; i is the transmission ratio; η is the transmission efficiency; r is the tire drum radius. Through this formula, the expected motor output torque can be calculated, and then the expected accelerator pedal opening α can be found according to the calibrated motor torque and accelerator pedal

opening relationship table. When the vehicle is in a braking state, the control amount of the lower controller is the desired accelerator opening. At this time, the relationship between the braking force $F_t = 0$, the braking force F_b and the master cylinder brake pressure P_{des} is linear, which can be expressed as:

$$F_b = K_b P_{des} \quad (12)$$

Finally, the lower-level controller outputs the desired accelerator pedal opening and the desired master cylinder brake pressure to the corresponding actuators to control the driving of the vehicle.

F. Control system working principle and circuit diagram

A set of qualified hardware system is the premise of software operation and the guarantee for controlling the execution of algorithms. The hardware is mainly composed of GPS/IMU/Lidar to offer the positioning function, domain controller and in-wire chassis, which is shown in figure 12 and figure 13. The mechanism diagram of the control system is shown in Figure 12, the control module receives data from the perception and planning module, and after processing by the control algorithm, the steering wheel corner data and acceleration and deceleration signal are output, and the actuator of the CAN communication is transmitted to the actuator of the wire control chassis.



Figure 12 the automatic vehicle



Figure 13 the domain controller

All positioning and control data are stored in the domain controller and calculated in the various modules. Firstly, GPS receivers will get the location data, if the program finds the location data change a lot and the SLAM module will replace the GPS to offer the location data. Then, the control module begin to work and follow the reference path to move.

CAN Bus is the hub to solve the data exchange between the domain controller and the VCU of the intelligent car-by-wire chassis. The CAN communication module model used by the domain controller in this paper is TJA1044GT/3. Figure 14 is a schematic diagram of its internal structure.

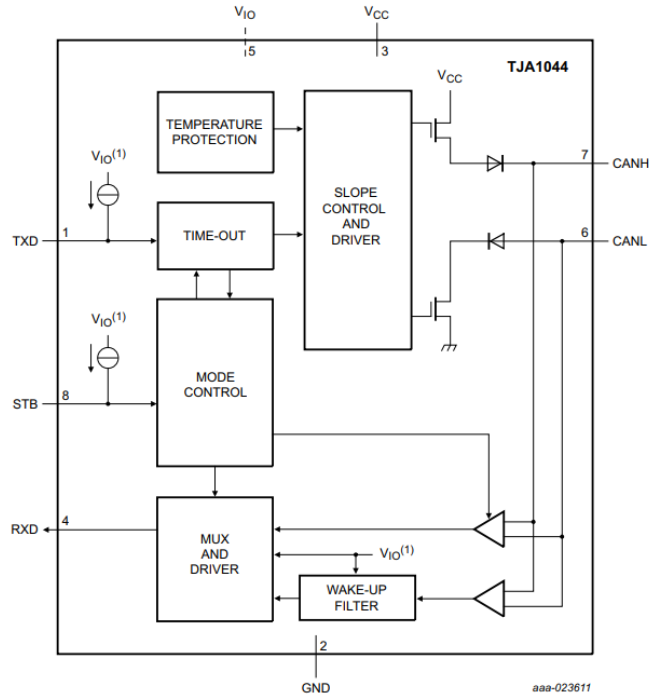


Figure 14 TJA1044GT/3 internal structure diagram

Table 3 is the function description of each pin

Table 3 TJA1044GT/3 Functional description of each pin		
symbolic name	number	function
TXD	1	Send data
GND	2	grounding
V _{CC}	3	power
RXD	4	Receive data
V _{IO}	5	I/O voltage
CANL	6	Low-level CAN
CANH	7	High-level CAN
STB	8	standby mode

G. Software Design and Process

Since the communication of the ROS system is in the form of a distributed architecture, specifically, each function is regarded as a network function node, and the entire system

is run through topic communication. According to this characteristic of the ROS system, we build a distributed network of the experimental platform in the experimental platform. In this way, as long as the above-mentioned required sensors are placed under the same local area network, the information exchange of all sensors can be realized.

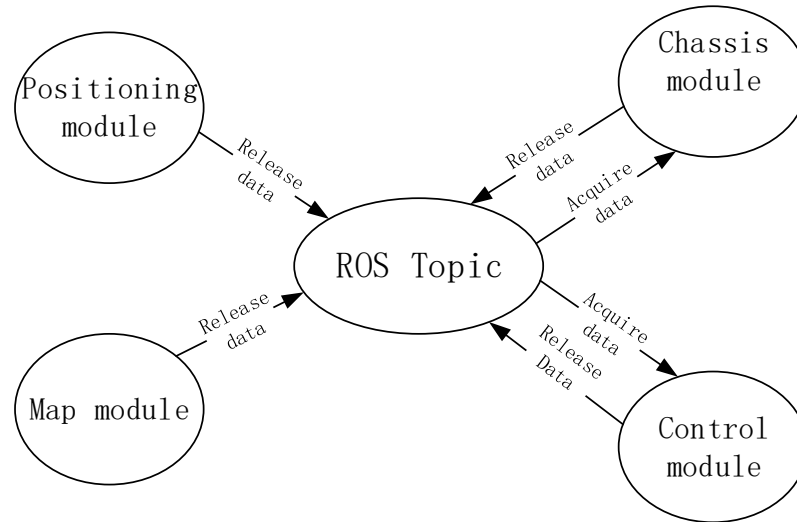


Figure 15 ROS communication mechanism

In this design, we will obtain all the chassis information of the vehicle through CAN in the chassis module, and then share the chassis data of the vehicle through ROS communication.

Controller Area Network(CAN) is the serial communication protocol of ISO International Organization for Standardization. CAN communication has the characteristics of fast speed and long distance. The speed is up to 1Mbps (distance <40m), and the maximum distance is 10KM (rate <5Kbps). CAN bus can connect multiple units at the same time, and has high reliability and good error detection ability. It is widely used in automotive computer control systems and industrial environments with harsh ambient temperature, strong electromagnetic radiation and large vibration.

The corresponding data information can be obtained by parsing the CAN message. There are two kinds of byte arrangement order of the message, one is the Intel arrangement order, and the other is the Motorola arrangement order. This solution will carry out the Motorola arrangement order.

The overall control system flowchart is shown in figure 16. As an important part of the control system, the horizontal control module needs to acquire the positioning data during the operation of the ROS system to acquire the current position of the test object

under the geodetic coordinate system; It also needs to contact with the map module to obtain the coordinates of the reference point in the direction of vehicle travel; Then control function should get the chassis module node information, like vehicle heading angle, yaw rate and other information.

The subscribed above node data will be used as the input of the lateral control algorithm, after the analysis of the vehicle dynamic characteristics in the predicted time domain, combined with the comparison of the difference of the reference trajectory, the lateral error and the heading angle error of the vehicle's current position and the reference point are selected as the cost, and the constraint value of the control

amount is changed adaptively by the fuzzy control module according to the range of vehicle speed, and finally the control amount is calculated by continuously reducing the generation value through the secondary planning algorithm, that is, the front wheel angle of the vehicle. Because the chassis by wire is controlled by the steering wheel angle, and through the conversion of the gear ratio to achieve the control of the vehicle front wheel angle.

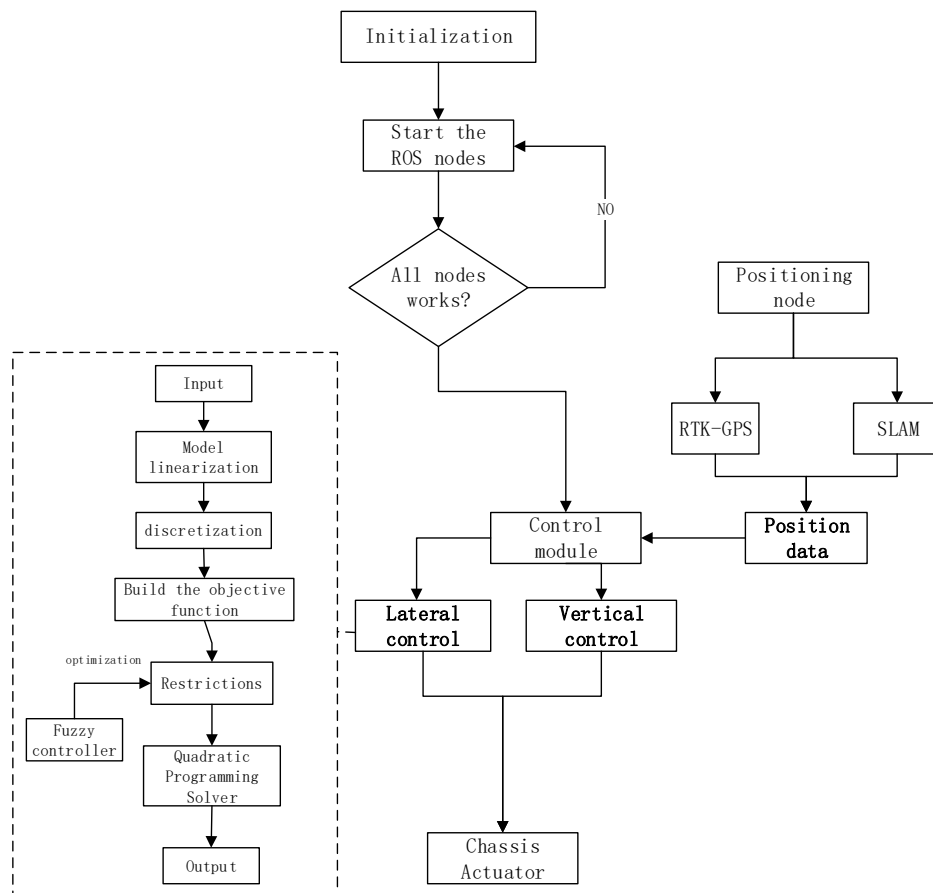


Figure 16 The overall control system flowchart

Therefore, the conversion from the front wheel angle to the steering wheel angle is realized by the lookup table method, and finally the steering wheel rotation angle is released in real time through the node information of the control module.

The longitudinal control module will realize the control of the acceleration and braking actions of the vehicle. First, it subscribes to the target value of the vehicle speed under different working conditions in the planning module. For example: the target tracking speed should reach 60km/h in the straight-line condition, decelerate to 30km/h in the turning condition, keep the vehicle speed at 40km/h in the "double-lane-shifting condition" and when the vehicle changes lanes, and need to slow down and stop when encountering a parking spot, and then perform PID control adjustment according to the current vehicle speed and the desired vehicle speed until the target vehicle speed value is reached.

Then this design will choose a reference path to follow and we need to collect this path data firstly. We can use the automatic driving platform that can realize high-precision positioning to collect the GPS information of the reference track point. The collected GPS track is actually a series of ordered latitude and longitude coordinates, which need to be converted into the WGS-84 coordinate system. After converting to local geodetic coordinate system, the coordinate value of our reference track points in the geodetic coordinate system can be obtained, which is map data required for tracking, which is saved in csv file format. When the vehicle is tracking, it is necessary to roll update the reference trajectory points in each cycle according to the vehicle pose information to ensure the continuity of the trajectory, which is conducive to the smooth output of the control amount and avoids the vehicle from shaking.

In order to reduce the computational difficulty in each cycle and ensure the real-time control, the reference trajectory in each cycle consists of 150 data points. When the reference track point information is collected, the start point and end point of the reference track will coincide, and the reference track is closed at this time. In order to ensure that the vehicle can travel along the closed reference trajectory without interruption, a method of adaptively updating the reference trajectory points is proposed. In each cycle, the reference trajectory points can follow the changes of the vehicle pose.

At first, the program will load the map file, according to the vehicle heading angle and the data point information in the map file, traverse and calculate the angle between all points in the map and the vehicle x-axis (vehicle forward direction). If the data points are satisfied, it means that these data points are in vehicle direction. Then, according to the vehicle position, the program will find the point closest to the vehicle in the data points of the vehicle's forward direction, which is the starting point of the reference trajectory, and record the line index of the point in the map file. For the next step, 150 loops are to be performed according to the index, and the value of temp index at each loop is recorded.

When the reference track is closed, the value of temp index must be compared with the size of map data in each cycle. If the number of temp index is larger than the map size, it means that the track point corresponding to temp index is in the next circle, and the size of map data needs to be subtracted to ensure that Continuity of reference track points. After the 150 cycles are over, select the corresponding point in the map for the temp index value of each cycle, and these 150 sequence points are the reference trajectory points in the cycle. In each cycle, the process will continue to generate reference trajectory points that can be dynamically updated with changes in the vehicle's pose, ensuring the continuity of the trajectory and avoiding vehicle jitter.

IV. RESULTS AND DISCUSSION

After completing the selection of the scheme and the transplantation of the algorithm, it is necessary to flash the control algorithm model into the real controller to complete the simulation platform test work that is close to the real vehicle test environment. According to actual needs, this scheme builds a hardware in the loop (HIL) simulation test system. The test bench consists of a 32-bit SPC5748GHK0AMKU6 domain controller, a host equipped with PreScan/Matlab simulation software, and a USB-to-CAN interface card, as shown in Figure 17. The operating interface of the PreScan simulation software used is shown in Figure 18.

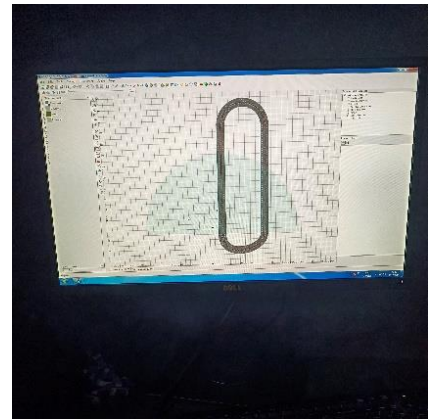


Figure 17 Controller and USB to CAN interface card Figure 18 Simulation operation interface

In the PreScan simulation software, the scene map can be built according to the actual tracking track coordinates; At the same time, the CAN protocol of the controlled vehicle can be analyzed in Simulink with the Matlab, and the transmission and reception of the vehicle control data can be completed in real time. Communication and processing between data are completed. Among them, the USB to CAN interface card supporting software can realize the loading of vehicle DBC files, CAN-open protocol, J1939 protocol and real-time saving of communication messages.

According to the function and performance requirements of the tracking control system, design and install the hardware platform and write the control program. Firstly, the test conditions are simulated on the HIL platform, and then the design can carry out the real vehicle test and debugging.

A. Straight and Curved Test Results

According to the previous research and scheme design, this design first performs hardware installation and software program transplantation, and then builds a 1:1 scene

map in the simulation software (Prescan) based on the recorded field tracking coordinate data on the HIL bench, as shown in the figure 19 and figure 20. The target speed of 10km/h is set for tracking control, and the maximum lateral error of the simulated vehicle tracking control in the straight section is controlled within 5cm, and the maximum lateral error in the curve is controlled within 10cm.

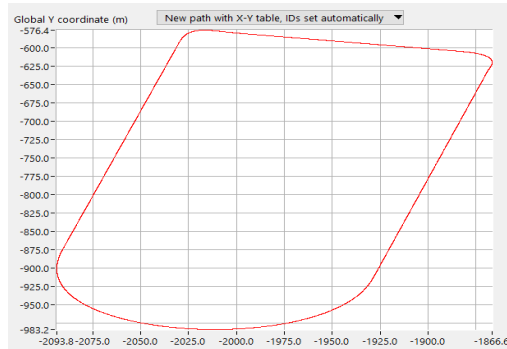


Figure 19 the real record trajectory



Figure 20 the simulation scenes

The actual trajectory to be tracked by the autonomous vehicle is shown in figure 19. In this design, the trajectory data will be collected in advance, and then according to the location of the vehicle at each moment, the map module will provide 150 reference trajectory points to the control module. The module will process these data and output the steering wheel angle at the next moment. For longitudinal speed control, when the target speed is 10Km/h, the smart car starts from a standstill. The control module will send acceleration information to the vehicle chassis to accelerate the vehicle. When the vehicle accelerates to the target speed, the speed will stabilize at the current 10 Km/h.

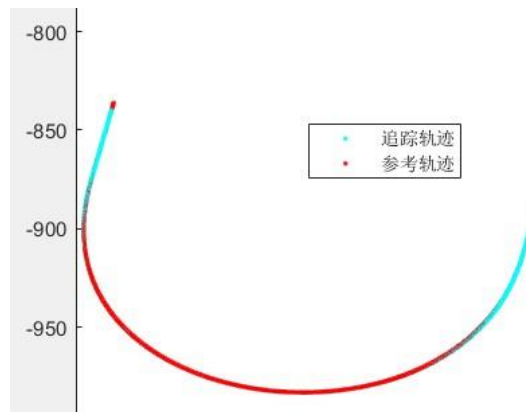


Figure 21 the result of tracking trajectory

As shown in the figure above, the red track is the reference track to be tracked by the vehicle, and the blue track points are the track the vehicle actually traveled. It can be seen that the actual driving trajectory of the vehicle and the reference trajectory almost

seen completely coincide, indicating that the actual control effect is very good.

B. Results of Double-line Change Test Conditions

This test refers to ISO/3888 technical report and GB/T 6323-2014 standard to set the target path of Double Lane Change (DLC) as shown in Figure 22.

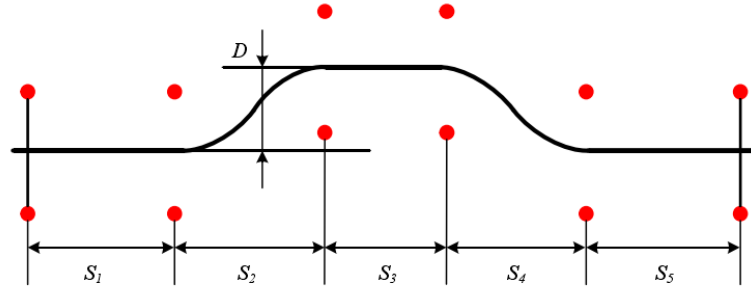


Figure 22 DLC path

The parameters are shown in table 4.

Table 4 Parameters of DLC

S1	S2	S3	S4	S5
65m	30m	25m	25m	65m

The DLC test condition is usually used to test the handling performance of the car, and it is also a severe test for the accuracy of the tracking control of the autonomous vehicle. The simulation conditions are shown in Figure 23. On a double lane with a width of 3.5m, the target vehicle will complete a standard "double shift line" trajectory at a speed of 20km/h. Figure 24 shows the test results of the real vehicle. During the whole tracking process, the actual path of the vehicle is basically consistent with the reference trajectory, and the maximum tracking error in the whole process is controlled within 10cm.



Figure 23 the simulation scenes

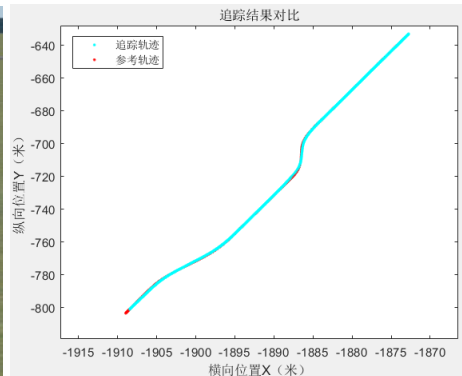


Figure 24 the real trajectory

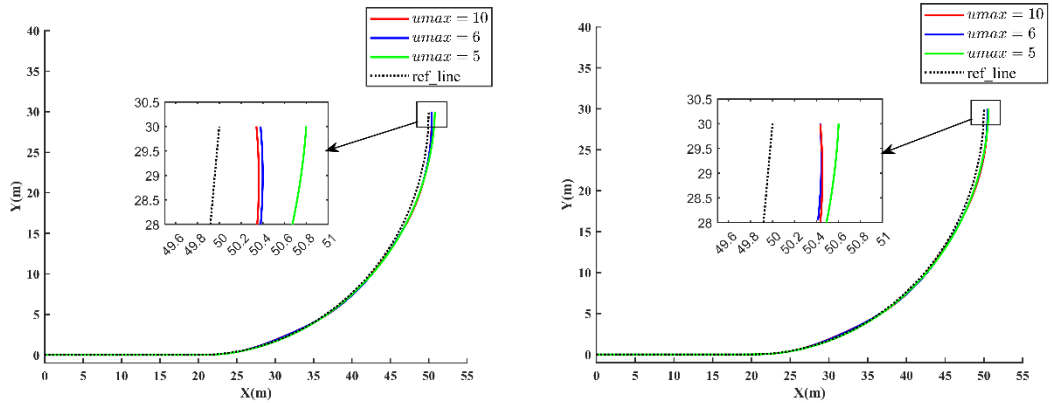
As shown in the figure 24, the red track is the reference track to be tracked by the vehicle, and the blue track points are the track the vehicle actually traveled. It can be seen that the actual driving trajectory of the vehicle and the reference trajectory almost seen completely coincide, indicating that the actual control effect is very good.

C. Optimization Results of Front Wheel Angle Adaptive Constraints

In the simulation of turning conditions at the intersection, the vehicle runs a 20m straight line at uniform speed to simulate the tracking effect of the path tracking controller on the straight line conditions, and then turns left according to the radius of 30m quarter circle to simulate the left turning process at the intersection. For safety reasons, the turning speed should not exceed 40Km/h and the front wheel angle σ_f was estimated based on the actual turning radius:

$$\sigma_f = \tan^{-1} \left(\frac{L}{R} \right) \quad (13)$$

Where L is the wheelbase of the vehicle, R is the turning radius of real-world, and the estimated threshold of front wheel rotation of the vehicle, $\sigma_f = 5.14^\circ$. Selecting the model predictive control parameters $Np=6$, $Nc=2$ and weight matrix $Q = [9000 \ 0; 0 \ 2000]$. The test speeds of the three groups of vehicles were 20Km/h, 30Km/h and 40Km/h respectively, and the constraints of σ_f were set to ± 5 degrees, ± 6 degrees and ± 10 degrees for each group of tests. Tracking path of the testing process and lateral acceleration at 30Km/h are shown in Figure 25.



(a) 20Km/h tracking trajectory

(b) 30Km/h tracking trajectory

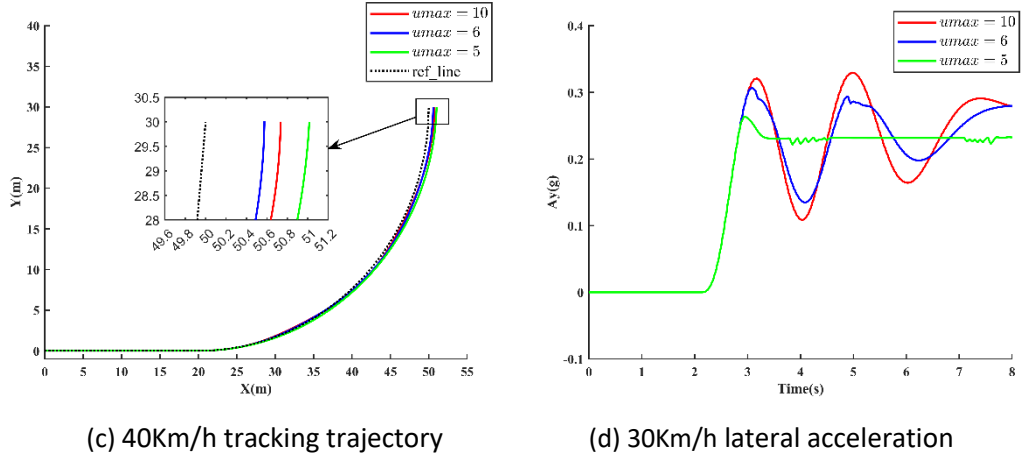


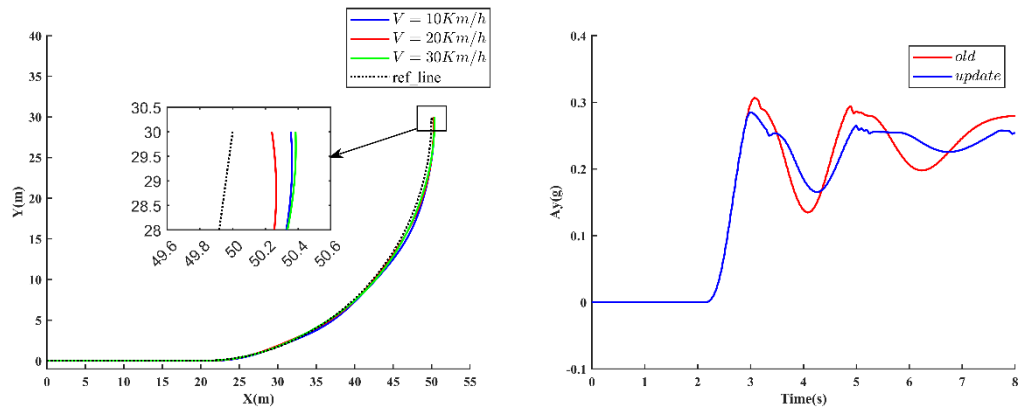
Fig 25. Vehicle state at different speed turning conditions

When the vehicle completes the turning action, the tracking state and reference track of the vehicle controlled at different speeds have certain errors. Compared with the threshold of σ_f corresponding to the 30m turning radius, the tracking error of the front wheel Angle less than the threshold is larger; When the speed is greater than the threshold, the tracking error increases gradually with the acceleration of σ_f . According to Figure 3 (d), when turning at 30Km/h, the lateral acceleration of the three groups of test vehicles was controlled below 0.4g, and the lateral acceleration of the vehicles gradually decreased with the decrease of the constraint value of σ_f .

Then, we perform fuzzy control on the front wheel turning angle constraint, and adaptively change the constraint value. In the simulation of turning conditions, three sets of speeds of 10Km/h, 20Km/h and 30Km/h were used to verify the improved model predictive controller. As shown in Figure 26 (a), the final state positions of the two groups with speeds of 10Km/h and 30Km/h were overlapped closely, and the maximum lateral error from the reference path was no more than 40cm. Compared with the best result in Figure 25 (b) of the original controller simulation results, the position deviation was reduced by 5cm and 25cm compared with the maximum deviation. The transverse error between the final state tracking position and the reference path at the speed of 20Km/h was 24cm, which was reduced by 10cm compared with the minimum transverse error of 34cm in FIG. 25 (a).

When the vehicle speed was less than 30Km/h, the lateral acceleration of the vehicle was small and the impact on vehicle comfort was not obvious. Therefore, the experimental group of 30Km/h was used as the optimized control test.

It can be seen from FIG. 26 (b) that the optimized lateral acceleration of the vehicle decreased significantly, and the optimized lateral acceleration was less than 0.3g.



(a) Actual tracking trajectory comparison

(b) 30Km/h lateral acceleration

Fig 26. Vehicle state at different speed turning conditions

According to the experimental comparison results of the two groups under different working conditions above, the model predictive controller optimized by fuzzy control has better performance in handling and stability of vehicle control, and the lateral error of path tracking was smaller.

V. CONCLUSIONS AND RECOMMENDATIONS

A. Conclusion about this project

In this project, a set of automatic driving tracking control system is designed, which realizes the real-time positioning of the smart car, and completes the accurate tracking of the reference trajectory by designing the control algorithm. This design conducts in-depth research on the key technologies in the positioning function and control function in the automatic driving technology.

In autonomous driving, SLAM can provide the vehicle with location information and surrounding environment information, so that the vehicle can complete the task of path planning and navigation. On the basis of reading and studying the previous researchers' research on SLAM algorithm and the practical application of SLAM, this paper conducts single-sensor SLAM related experiments based on 32-line Lidar, and on the basis of the first two experiments, conducted an exploratory SLAM experiment of two types of sensor fusion. Through related theoretical research, building a matching experimental platform, and related experimental verification and result analysis, it provides more solid theoretical data support for the application of SLAM in future autonomous driving, which can effectively promote autonomous driving.

In the second part, the model predictive control algorithm and the PID control algorithm are analyzed respectively, and the software program is designed and the real vehicle experiment is carried out. Among them, the constraint on the control quantity in the model predictive control algorithm is improved, so that the corner constraint value can adapt to the speed to control the maximum steering angle of the vehicle, the steering precision and the stability of the vehicle are improved. Design and verify the function that the reference trajectory can be updated with the vehicle in real time, which provides a real-time reference trajectory point for the lateral control algorithm and improves the practicability of the control system.

B. Outlook for the Future Work

Although this paper has made some attempts for the research of intelligent vehicle road positioning and path tracking, due to the limitations of time allocation, personal energy and ability, it can be further expanded in the following aspects:

- a. In the research process, it is assumed that the information of each variable of the system can be obtained in real time, and the influence of the system time delay effect on the tracking performance is not considered, while the adverse effects caused by the time delay effect of the actual intelligent vehicle system are common. The time-delay effect of the path-following system can be included in the research.
- b. In the 3D-SLAM experiment based on Lidar, the final imaged point cloud effect is general, and the mapping accuracy is low. Therefore, in the future SLAM problem based on multi-line Lidar, 64-line or even 128-line Lidar can be used for 3D-SLAM.
- c. Intelligent decision-making is a key technology to realize the unmanned driving of intelligent vehicles. This paper does not give too much consideration to it. In the future, based on the comprehensive control of the vertical and horizontal directions of intelligent vehicles, path tracking control can be applied to more intelligent assisted driving technologies.
- d. The actual vehicle test of the control algorithm is relatively simple, and the working conditions are relatively simple. In the future research, more sets of expected speeds should be used to test for more types of road conditions to further verify the actual vehicle test process of the control algorithm. adaptability to road conditions and vehicle speed.

VI. REFERENCES

- [1] Toroyan T. WHO Global Status Report on Road Safety: Time for Action.[J]. Inj Prev, 2021,1515(4):286.
- [2] . Introduction to Self-Driving Cars [M]. Beijing Institute of Technology Press, 2014.
- [3] Gao M. Research on Target Tracking and Trajectory Prediction in Complex Traffic Environment Based on Deep Learning [D]. Jilin University Press, 2020.
- [4] Zhu H, Yuen K V, Mihaylova L, et al. Overview of environment perception for intelligent vehicles[J]. IEEE Transactions on Intelligent Transportation Systems, 2017, 18(10): 2584-2601
- [5] Girshick R, Donahue J, Darrell T, et al. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation[C]//Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition. Columbia: IEEE, 2014: 580-587.
- [6] Ren S, He K, Girshick R, et al. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(6): 1137-1149.
- [7] Xu B , Jeff B X G , Bian Y G , et a1 . Cooperative method of traffic signal optimization and speed control of connected vehicles at isolated intersections[J]. IEEE Transactions on Intelligent Transportation Systems, 2019, 20(4): 1390-1403.
- [8] Han X, Ma R, H. Zhang M. Energy aware trajectory optimization of CAV platoons through a signalized intersection[J]. Transportation Research Part C, 2020 , 118 : 102652.
- [9] Xu Guoyan, Zong Xiaopeng, Yu Guizhen. Research on intelligent obstacle avoidance method for unmanned vehicles based on DDPG[J]. Automotive Engineering, 2019,41(2): 206-212.
- [10] Smith R C, Cheeseman P. On the representation and estimation of spatial uncertainty[J]. The international journal of Robotics Research, 1986, 5(4): 56-68.
- [11] Durrant-Whyte H F. Uncertain geometry in robotics[J]. IEEE Journal on Robotics and Automation, 1988, 4(1): 23-31.

- [12] Smith R, Self M, Cheeseman P: Estimating uncertain spatial relationships in robotics, Autonomous robot vehicles: Springer, 1990: 167-193.
- [13] Leonard J J, Durrant-Whyte H F. Directed sonar sensing for mobile robot navigation[M]. 175. Springer Science & Business Media, 2012.
- [14] Rencken W D. Concurrent localisation and map building for mobile robots using ultrasonic sensors[C]. Proceedings of 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'93), 1993: 2192-2197.
- [15] Csorba M. Simultaneous localisation and map building[D]. University of Oxford Oxford, 1997.
- [16] Csorba M, Durrant-Whyte H, Uhlmann J. A new approach to simultaneous localisation and map building[C]. Proc. SPIE Aerosense, 1996.
- [17] Shan T, Englot B, Meyers D, et al. Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping[C]. 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2020: 5135-5142.
- [18] Chetverikov D, Svirko D, Stepanov D, et al. The trimmed iterative closest point algorithm[C]. Object recognition supported by user interaction for service robots, 2002: 545-548.
- [19] Grant D, Bethel J, Crawford M. Point-to-plane registration of terrestrial laser scans[J]. ISPRS Journal of Photogrammetry and Remote Sensing, 2012, 72: 16-26.
- [20] Segal A, Haehnel D, Thrun S. Generalized-icp[C]. Robotics: science and systems, 2009: 435.
- [21] Deschaud J-E. IMLS-SLAM: Scan-to-model matching based on 3D data[C]. 2018 IEEE International Conference on Robotics and Automation (ICRA), 2018: 2480-2485.
- [22] Parkison S A, Gan L, Jadidi M G, et al. Semantic Iterative Closest Point through Expectation-Maximization[C]. BMVC, 2018: 280.
- [23] Grant W S, Voorhies R C, Itti L. Finding planes in LiDAR point clouds for real-time registration[C]. 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2013: 4347-4354.

- [24] Velas M, Spanel M, Herout A. Collar line segments for fast odometry estimation from velodyne point clouds[C]. 2016 IEEE International Conference on Robotics and Automation (ICRA), 2016: 4486-4495.
- [25] Zhang J, Singh S. LOAM: Lidar Odometry and Mapping in Real-time[C]. Robotics: Science and Systems, 2014: 1-9.
- [26] Shan T, Englot B. Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain[C]. 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2018: 4758-4765.
- [27] Li Yangming, Meng Qinghu. A widely applicable feature extraction method for lidar data[J]. Robotics, 2010, 32(6): 10.
- [28] Sun H. Simultaneous localization and mapping of mobile robots with seamless fusion of multimodal features in structured environments [D]. Nankai University.
- [29] Pan Y, Xiao P, He Y, et al. MULLS: Versatile LiDAR SLAM via multi-metric linear least square[C]. 2021 IEEE International Conference on Robotics and Automation (ICRA), 2021: 11633-11640.
- [30] Biber P, Straßer W. The normal distributions transform: A new approach to laser scan matching[C]. Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453), 2003: 2743-2748.
- [31] Magnusson M, Nüchter A, Lörken C, et al. 3D mapping the Kvarntorp mine: a field experiment for evaluation of 3D scan matching algorithms[C]. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Workshop" 3D Mapping", Nice, France, September 2008, 2008.
- [32] Hong H, Lee B H. Probabilistic normal distributions transform representation for accurate 3D point cloud registration[C]. 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2017: 3333-3338.
- [33] Akai N, Morales L Y, Takeuchi E, et al. Robust localization using 3D NDT scan matching with experimentally determined uncertainty and road marker matching[C]. 2017 IEEE Intelligent Vehicles Symposium (IV), 2017: 1356-1363.

- [34] Lee S, Kim C, Cho S, et al. Robust 3-Dimension Point Cloud Mapping in Dynamic Environment Using Point-Wise Static Probability-Based NDT Scan-Matching[J]. IEEE Access, 2020, 8: 175563-175575.
- [35] Ayache N, Faugeras O D. Building, registrating, and fusing noisy visual maps[J]. The International Journal of Robotics Research, 1988, 7(6): 45-65.
- [36] Crowley J L. World modeling and position estimation for a mobile robot using ultrasonic ranging[C]. ICRA, 1989: 674-680.
- [37] Chatila R, Laumond J-P. Position referencing and consistent world modeling for mobile robots[C]. Proceedings. 1985 IEEE International Conference on Robotics and Automation, 1985: 138-145.
- [38] Julier S J, Uhlmann J K. A counter example to the theory of simultaneous localization and map building[C]. Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164), 2001: 4238-4243.
- [39] Jiang Zuopeng, Mei Tiancan. A single-line laser odometer based on PL-ICP and NDT point cloud matching[J]. Laser Journal, 2020, 41(3): 4.
- [40] Montemerlo M, Thrun S, Koller D, et al. FastSLAM: A factored solution to the simultaneous localization and mapping problem[J]. Aaai/iaai, 2002, 593598.
- [41] Murphy K, Russell S: Rao-Blackwellised particle filtering for dynamic Bayesian networks, Sequential Monte Carlo methods in practice: Springer, 2001: 499-515.
- [42] Montemerlo M, Thrun S, Koller D, et al. FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges[C]. IJCAI, 2003: 1151-1156.
- [43] Grisetti G, Stachniss C, Burgard W. Improved techniques for grid mapping with rao-blackwellized particle filters[J]. IEEE transactions on Robotics, 2007, 23(1): 34-46.
- [44] Aulinas J, Petillot Y, Salvi J, et al. The SLAM problem: a survey[J]. Artificial Intelligence Research and Development, 2008: 363-371.
- [45] Triggs B, Mclauchlan P F, Hartley R I, et al. Bundle adjustment—a modern synthesis[C]. International workshop on vision algorithms, 1999: 298-372.

- [46] Kschischang F R, Frey B J, Loeliger H-A. Factor graphs and the sum-product algorithm[J]. IEEE Transactions on information theory, 2001, 47(2): 498-519.
- [47] Dellaert F, Kaess M. Square Root SAM: Simultaneous localization and mapping via square root information smoothing[J]. The International Journal of Robotics Research, 2006, 25(12): 1181-1203.
- [48] Folkesson J, Christensen H I. Graphical SLAM for outdoor applications[J]. Journal of Field Robotics, 2007, 24(1 - 2): 51-70.
- [49] Grisetti G, Stachniss C, Grzonka S, et al. A tree parameterization for efficiently computing maximum likelihood maps using gradient descent[C]. Robotics: Science and Systems, 2007: 9.
- [50] Kaess M, Johannsson H, Roberts R, et al. iSAM2: Incremental smoothing and mapping using the Bayes tree[J]. The International Journal of Robotics Research, 2012, 31(2): 216-235.
- [51] Olson E, Leonard J, Teller S. Fast iterative alignment of pose graphs with poor initial estimates[C]. Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006., 2006: 2262-2269.
- [52] Thrun S, Montemerlo M. The graph SLAM algorithm with applications to large-scale mapping of urban structures[J]. The International Journal of Robotics Research, 2006, 25(5-6): 403-429.
- [53] Shen C, O'brien J F, Shewchuk J R: Interpolating and approximating implicit surfaces from polygon soup, ACM SIGGRAPH 2004 Papers, 2004: 896-904.
- [54] Nüchter A. 3D robotic mapping: the simultaneous localization and mapping problem with six degrees of freedom[M]. 52. Springer, 2008.
- [55] Pizzoli M, Forster C, Scaramuzza D. REMODE: Probabilistic, monocular dense reconstruction in real time[C]. 2014 IEEE International Conference on Robotics and Automation (ICRA), 2014: 2609-2616.
- [56] Li C. Adaptive sliding mode path tracking control for agricultural wheeled mobile robot[J]. China Mechanical Engineering , 2018,29(5): 579-584.
- [57] Da Lu. An Improved Path Tracking Algorithm for Intelligent Vehicle in Automatic Parking Conditions[J]. Applied Mechanics and Materials, 2014, 3253: 326-330.

- [58] Jin Xinyu, Zhang Jun, Liu Yuansheng. Research on Adaptive Optimal Preview Model Based on Stanley Algorithm[J]. Computer Engineering, 2018, 44(7):42-46.
- [59] Xu Yangyang, Wang Ying, Xue Dongbin. Research on Motion Error of Mobile Robot Using Improved Fuzzy Neural Network PID Control [J]. Chinese Journal of Construction Machinery, 2019, 17(6):510-514.
- [60] Meng Xiangping, Kou Lei, Yuan Quande, et al. Robot Motion Control Based on Improved Integral Separation PID[J]. Journal of Northeast Electric Power University, 2016, 36(5):86-91.
- [61] Feng Jian, Zhang Wen'an, Ni Hongjie, et al. PID control method for trajectory tracking of wheeled mobile robot[J]. Information and Control, 2017, 46(4):385-393.
- [62] Li Linchen, Jiang Xiaoping. Path tracking of mobile robot based on PID control[J]. Laser Journal, 2016, 37(2):110-112.
- [63] Shen Xianqing, Sun Qizhi, Ma Zhipeng. Path following control of mobile robot based on double closed-loop Proportion & feedforward[J]. Journal of Heilongjiang University of Science and Technology, 2021, 31(2):223-228.
- [64] Wang Rongben, Zhang Ronghui, Chu Jiangwei, et al. Optimal design and quality analysis of regional traffic intelligent vehicle controller[J]. Journal of Agricultural Machinery, 2007, 38(1):22-25, 48.
- [65] Ni Lanqing, Lin Fen, Wang Kaizheng. Research on Intelligent Vehicle Path Tracking Control Based on Preview[J]. Journal of Chongqing University of Technology, 2017, 31(3):27-33.
- [66] Meng Yu, Wang Yu, Gu Qing, et al. LQR-GA path tracking control of articulated vehicles based on predicted pose information[J]. Chinese Journal of Agricultural Machinery, 2018, 49(6):375-384.
- [67] Gao Linlin, Tang Fengmin, Guo Peng, et al. Research on improved LQR method for lateral motion control of autonomous driving[J]. Mechanical Science and Technology, 2021, 40(3):435-441.
- [68] Cutler R, Ramakar B. Dynamic Matrix Control-A Computer Control Algorithm[C]// Joint Automatic Control Conference. 1980.

- [69] Richalet J, Rault A, Testud J, et al. Model predictive heuristic control: Applications to an industrial process[C]// Automatic. 1978:413-428.
- [70] Xi Yugeng, Xu Xiaoyu, Zhang Z. Research Status of Predictive Control and Multilayer Intelligent Predictive Control[J]. Control Theory and Application, 1989(2):1-42.
- [71] Shan E, Dai B, Song J, et al. A Dynamic RRT Path Planning Algorithm Based on B-Spline[C]. 2009 Second International Symposium on Computational Intelligence and Design, Changsha, Hunan, China, 2009(2).
- [72] K. Cao, Q. Cheng, S. Gao, Y. Chen and C. Chen. Improved PRM for Path Planning in Narrow Passages[C]. 2019 IEEE International Conference on Mechatronics and Automation (ICMA), Tianjin, China, 2019: 45-50.
- [73] Yang, KM, Liu, YH, Liu, YL, et al. A Linear Time-varying MPC Method for Vehicle Path-Following Assistance Based on Steering Torque[C]. 43rd annual conference of the IEEE Industrial Electronics Society, 2017:4559-4564.
- [74] Sun C, Zhang X, et al. A Model Predictive Controller with Switched Tracking Error for Autonomous Vehicle Path Tracking[J]. IEEE Access, 2019, (99):1-12.
- [75] Falcone P, Borrelli F, Asgari J, et al. Predictive Active Steering Control for Autonomous Vehicle Systems[J]. IEEE Transactions on Control Systems Technology, 2007, 15(3):566-580.
- [76] Shi Zhen, Jiang Hong, Yu Wenhao, etc. Research on Adaptive MPC Algorithm for Path Tracking Control[J]. Computer Engineering and Applications, 2020, 56(21):266-271.
- [77] Bai G, Meng Y, Liu L. Vehicle path tracking control based on variable prediction time domain and speed[J]. China Mechanical Engineering, 2020, 31(11):1277-1284.
- [78] Du X. Development of a Genetic-Algorithm-Based Nonlinear Model Predictive Control Scheme on Velocity and Steering of Autonomous Vehicles[J]. IEEE Transactions on Industrial Electronics, 2016, 63(11):6970-6977.
- [79] Chen S, Feng Nenglian, Tang Jie, et al. Study on handling stability of active front wheel steering[J]. Journal of Hubei Institute of Automotive Industry, 2014(3):11-15.

VII. APPENDICES

(1) The downloaded lidar driver library contains the folder `ouster_client`, the folder `ouster_ros` and the folder `ouster_viz`. First complete the installation of dependencies in the folder `ouster_viz`:

```
sudo apt install build-essential cmake libglfw3-dev libglew-dev libeigen3-dev \libjsoncpp-dev libtclap-dev
```

```
sudo apt install ros-kinetic-ros-core ros-kinetic-pcl-ros \ros-kinetic-tf2-geometry-msgs ros-kinetic-rviz
```

```
source /opt/ros/kinetic/setup.bash
```

```
mkdir -p ./myworkspace/src
```

```
cd myworkspace
```

```
ln -s <path to ouster_example> ./src/
```

```
catkin_make -DCMAKE_BUILD_TYPE=Release
```

(2) Go to the `ouster_example` -> `ouster_client` folder, open the `README.md`, and follow the steps inside to build the client;

(3) Finally, go to the `ouster_example` -> `ouster_ros` folder, open the `README.md` in it, and follow the steps inside to complete the build.

(4) Connecting to the Lidar:

```
sudo ifconfig enp3s0 10.5.5.1
```

Selecting the ip of Lidar: `sudo dnsmasq -C/dev/null -kd -F 10.5.5.50,10.5.5.100 -i enp3s0 -`

Set ptp time sync terminal input before starting the driver:

```
nc os1-122005000054.local 7501
```

```
set_config_param timestamp_mode TIME_FROM_PTP_1588
```

```
reinitialize
```

```
write_config_txt
```

Open `sudo ptpd -i enp3s0 -M` in another terminal to syncbind-dynamic;

After entering the command, you can plug and unplug the network cable, which can be searched and allocated more quickly.

Start the radar driver : `roslaunch ouster_ros ouster.launch sensor_hostname:=10.5.5.61 udp_dest:=10.5.5.1 viz:=true image:=true timestamp_mode:=TIME_FROM_PTP_1588`

```
roslaunch ouster_ros ouster.launch sensor_hostname:=192.168.1.210 udp_dest:=192.168.1.109 viz:=true image:=true timestamp_mode:=TIME_FROM_PTP_1588
```

Typing in the terminal: `roslaunch parking parking`

[illegible]

Typing in the terminal: `source`

After completing the above operations, the work of the mapping part can be carried out. In this experiment, a closed-loop road in front of the library of campus of our school was selected for the test.