

Received 3 September 2023, accepted 13 October 2023, date of publication 23 October 2023, date of current version 1 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3326754

RESEARCH ARTICLE

An Intelligent Navigation Control Approach for Autonomous Unmanned Vehicles via Deep Learning-Enhanced Visual SLAM Framework

LU CHEN¹, YAPENG LIU¹, PANPAN DONG³, JIANWEI LIANG¹, AND AIBING WANG³

¹Department of Mechanical and Electrical Engineering, Shijiazhuang College of Applied Technology, Shijiazhuang 050081, China

²Department of Information Science and Engineering, Hebei Vocational College of Labour Relations, Shijiazhuang 050091, China

³Department of Automotive Engineering, Hebei Jiaotong Vocational and Technical College, Shijiazhuang 050035, China

Corresponding author: Yapeng Liu (ljw800723@mail.sdufe.edu.cn)

ABSTRACT For autonomous unmanned vehicles (AUVs), navigation control is an essential concern in academia. Conventionally, existing research works dealt with this issue by resorting to the “Simultaneous Localization and Mapping” (SLAM) technology. Although many SLAM-based approaches had been proposed in recent years, they could not mostly perceive semantic characteristics from dynamic visual scenarios. To deal with this challenge, this paper proposes a navigation control approach for AUVs via a deep learning-enhanced visual SLAM framework. Firstly, the commonly used coordinate system in the motion of AUVs is analyzed, and a mathematical description of the circular arc motion of AUVs is formulated. Then, an up-convex curve model is adopted to realize high-accuracy detection of the bilateral lane lines. On the foundation, a yaw angle guidance-based imitation learning framework is utilized to realize navigation control. This well facilitates the analyzing the causal relationship between scenarios and decisions. Some experiments are conducted by simulating real-world urban road scenes, to testify the efficiency of the navigation control for AUVs. The results show that navigation accuracy can be improved by about 5%.


INDEX TERMS Autonomous unmanned vehicles, navigation control, deep learning, visual SLAM.

I. INTRODUCTION

Traffic accidents, road congestion, and environmental pollution caused by vehicle exhaust have become urgent social problems [1]. In this context, intelligent vehicle technology with autonomous driving as the core provides an effective solution to the above problems and has become the most important application area for artificial intelligence technology to be implemented. The driverless vehicle is a combination of image processing, artificial intelligence, automatic control, and many other technologies with one to achieve autonomous navigation. The current mainstream autonomous driving solutions consist of global planning, decision-making, local planning, and control modules [2]. The modular autonomous driving solution based on high-precision maps consists of a perception module that transmits environmental information to a decision module,

which outputs behavioral strategies, and a path planning module that plans the autonomous vehicle trajectory according to the strategies [3]. And finally, a control module that outputs control values for the steering wheel, throttle, and brake by fitting the current position of the vehicle to the target trajectory position [4].

However, in large-scale complex road conditions, the engineering cost of this solution increases dramatically. And it cannot take into account extreme situations [31]. Artificial intelligence technologies are expected to support the achievement of these goals [32]. One of the long-term goals in the field of artificial intelligence is that intelligence can learn effective strategies by interacting with their environment and optimize them as they gain practical experience [33]. The data-driven autonomous learning paradigm based on reinforcement learning provides the mathematical framework to achieve this goal, from the field of robotics to recommendation systems [34]. From quantitative trading systems for financial markets to natural language processing [35].

The associate editor coordinating the review of this manuscript and approving it for publication was Laura Celentano .

Reinforcement learning has been successful in many application areas [36]. However, these approaches lack scalability and are limited to dealing with low-dimensional problems. In recent years, the rise of deep learning, with its powerful function approximation and the representational learning capabilities of neural networks, has made it possible to use reinforcement learning to deal with complex problems of high dimensionality [37].

Deep learning has solved the dimensional catastrophe problem of machine learning and has achieved great success in areas such as image recognition and natural language processing. It has also greatly contributed to the development of reinforcement learning [38]. Deep reinforcement learning is a method that combines reinforcement learning with deep learning. Deep learning extends reinforcement learning to previously intractable decision problems, i.e., high-dimensional (continuous) state and action spaces. Deep reinforcement learning combines the powerful representational capabilities of deep learning with the powerful inference capabilities of reinforcement learning to determine the behavior of an intelligent body. It provides an alternative solution for the implementation of autonomous driving technology. Currently, the application of deep reinforcement learning methods to autonomous driving technology is still in its infancy and faces many problems. Imitation learning is another autonomous learning method. By introducing expert experience to make up for the deficiency of slow learning speed and poor stability of deep reinforcement learning.

However, there are still problems that need to be solved, such as there is no unified answer for how to design the state space to describe the environment completely and accurately, and how to design the reward function to help the vehicle learn a better policy. In this paper, we mainly use deep reinforcement learning as an autonomous learning method to implement autonomous driving decisions and control. When facing complex urban scenes, another autonomous learning method, imitation learning, is introduced for the shortage of slow convergence of deep reinforcement learning. The main research of this paper is about SLAM technology and path planning algorithm for unmanned vehicles, choosing adaptive Monte Carlo localization as a means of localization, studying the classical EKF-SLAM technology based on extended Kalman filter and RBPF-SLAM technology based on particle filter, synthesizing the advantages and disadvantages of both algorithms, choosing the particle filter-based algorithm for further research, and combining the improved particle filter-based The Gmapping algorithm based on improved particle filtering and Cartographer algorithm based on graph optimization are tested for indoor environment mapping, and a preliminary framework of mapping algorithm is established for autonomous navigation of unmanned vehicles.

II. RELATED WORK

The initial solution to the SLAM problem was designed based on the Kalman filter algorithm. The Kalman filter algorithm research idea initially solves the SLAM problem in specific

cases, but due to the limitations of the algorithm itself, the algorithm can only be applied to systems with linear Gaussian properties. In the literature [5], a Gmapping algorithm based on the Fast SLAM scheme was proposed. Gmapping is a particle filtering-based 2D SLAM algorithm using LiDAR sensors, which is now commonly used for the localization and construction of 2D maps of indoor environments. After years of development, the Gmapping algorithm is simple, easy to implement, and low-cost. Basiri et al. [6] used an indoor robotics platform based on a Raspberry Pi board for indoor construction using the Gmapping algorithm, and the results demonstrated the robustness of navigation. Zhao et al. [7] proposed a gradient correction refinement algorithm, a new method for localization based on improved particle filtering, which extends the traditional particle filtering algorithm and is more general.

In the literature [8], a fast SLAM system based on 3D attitude is proposed, where the vehicle trajectory can be better estimated by carrying a 360° LIDAR. Zhu and Yang [9] proposed Foot SLAM is an algorithm based on the principle of simultaneous localization and composition (SLAM), which relies on human mileage (i.e., the measurement of pedestrian's footsteps) to construct probabilistic maps of human motion for such environments. The literature [10] used two RTK-GPS receivers mounted on a vehicle to achieve position and direction feedback of the robot, designed a backstepping sliding mode control (SMC) navigation controller, and verified the capability and robustness of high-degree-of-freedom incomplete system control. However, satellite positioning navigation is susceptible to environmental constraints such as floors, trees, and tunnels, especially in indoor scenarios making the reception of satellite signals subject to interference, resulting in reduced accuracy and even positioning failure. Zhou et al. [11] used a multi-sensor fusion algorithm to further improve localization and navigation accuracy by making full use of the advantages of various sensors in the hardware architecture to compensate for individual sensor deficiencies.

The literature [12] proposed an inter-frame matching scheme for point cloud information collected by LiDAR, which combines correction and optimization with nonlinear optimization based on an improved bit-pose algorithm for laser point cloud data to accomplish localization of mobile robots in unknown environments. The literature [13] improved the ant colony algorithm to solve this aspect. The fast random expansion tree method algorithm (RRT) is a tree-like graph-based algorithm, which is structured as a randomly bifurcated tree structure from inside to outside, with the tree extension direction determined by random adoption in the re-space. Tong et al. [14] proposed a 3D vision robot path planning method based on the data model, which focuses on the accuracy of vector testing of the robot's distance to obstacles in the case of more complex obstacles, and allows the mobile robot to complete a high-precision obstacle avoidance function. Zhu and Yang [15] proposed a machine vision-based navigation method for wheeled robots mainly

using a fuzzy sliding mode control model to complete the path tracking control of mobile robots. In the literature [16], a simple and inexpensive visual navigation system is studied, which is based on monocular visual navigation techniques to localize and complete navigation tasks for a series of road signs. In recent years, with the widespread use of deep learning in the field of image detection, research scholars have started to use deep learning methods to learn navigation information directly from raw perceptual data.

III. METHODOLOGY

A. ROS-BASED AUTONOMOUS NAVIGATION FRAMEWORK

ROS (Robot Operating System) is one of the most popular and advanced frameworks, which is widely used in the field of autonomous navigation because of its ease of operation, rich functionality, and power. ROS itself is a distributed communication framework, which can help program processes to communicate with each other more easily. The core idea of ROS is to make different software functions into nodes and communicate between them through messages. Figure 1 shows the scene of SLAM processing multiple traffic semantic information.

The system consists of the following main components:

- **Sensor data acquisition:** The system utilizes LIDAR measurements and motion data from the unmanned vehicle's built-in odometer as a data source. The odometer provides data on the movement speed and azimuth changes of the driverless vehicle to control the movement of the driverless vehicle; the data collected by the LiDAR provides information about the environment of the driverless vehicle [17]. By combining data from both sensors, the driverless vehicle is guided through route navigation.
- **Coordinate conversion:** Before the unmanned vehicle can be positioned, the position coordinates of the sensor need to be changed. Since the sensor coordinates used are relative, the sensor measurements must be converted to position coordinates in the global coordinate system.
- **SLAM module:** Unmanned vehicles map their environment, locate their position in a completely unknown environment, and navigate using the environment map and its location at the same time, which is the problem of instant localization and map generation (SLAM) for driverless vehicles, which will be studied in this module.
- **Path planning module:** Path planning is one of the key technologies for unmanned driving. This part focuses on path planning for maps with known global information as well as completing path planning based on local information.

In addition, the sensor data needs to be collected, and the data from the LIDAR combined with the wheel odometer of the cart chassis is used as the input quantity of the whole navigation system. The LIDAR scans the surrounding environment to obtain the obstacle information of the surrounding environment, at this moment the radar scanning point is in the

radar coordinate system; the wheel odometer is measured by the encoder sensor to get the rotational speed of the wheel combined with the kinematic model to get the movement speed (linear and angular velocity) change, and the odometer data (coordinates and heading of the chassis) are obtained by the track projection method, and the odometer coordinate system is established at the position of the chassis. The fusion of RBPF-SLAM and EKF-SLAM adopted in this paper combines the advantages of the two, avoids the defect that small-weight particles may be lost in the simulation process, and optimizes the uncertainty in particle filtering. Compared to the current state-of-the-art cartographer algorithm which needs to obtain the sensor scanning data of each frame, the algorithm proposed in this paper guarantees the accuracy while requiring a lower amount of data.

Since the sensor coordinate system is a relative coordinate system at this moment, it is necessary to convert the sensor data to the global coordinate system and use TF coordinate transformation to ensure the uniformity and consistency of the data. The odometer data are used to obtain the position estimates of the car, and these position estimates are used as the initial values to match with the radar data to update the position estimator values, and frame by frame the radar data are motion filtered and continuously superimposed to form a 2D raster map. this process completes the map building and positioning, at this time the new coordinate system is obtained as the global coordinate system, and then the map server can save and read the map information [18]. In the established global and local cost maps, after specifying the coordinates and orientation of the target points, the global path planner combines the sensor data after the coordinate system conversion to plan the offline global optimal path. The local planner settles the speed combination of the optimal trajectory in the specific motion based on the local cost map information, and in case of emergency failure or target unreachability, the motion state recovery mechanism is activated to help the cart start navigation again.

Figure 2 shows the framework diagram of ROS-based autonomous navigation. As the last part of the autonomous navigation system, the velocity combination calculated by the local planner is passed to the cart chassis controller in the form of linear and angular velocity control quantities to enable the cart to achieve tracking motion for the given path. In this paper, we mainly focus on two aspects of map building and positioning and path planning and designing an autonomous navigation framework based on ROS in combination with the existing hardware base. The SLAM problem is essentially a theoretical probabilistic model based on a probabilistic model. The SLAM problem has two elements of importance: first, the use of sensors to capture environmental information and process the information while constructing a SLAM map that matches the environmental features. Secondly, the position of the unmanned vehicle is estimated based on the environmental feature information. The SLAM problem is whether the robot can navigate in an environment without prior information and build a map of its surroundings

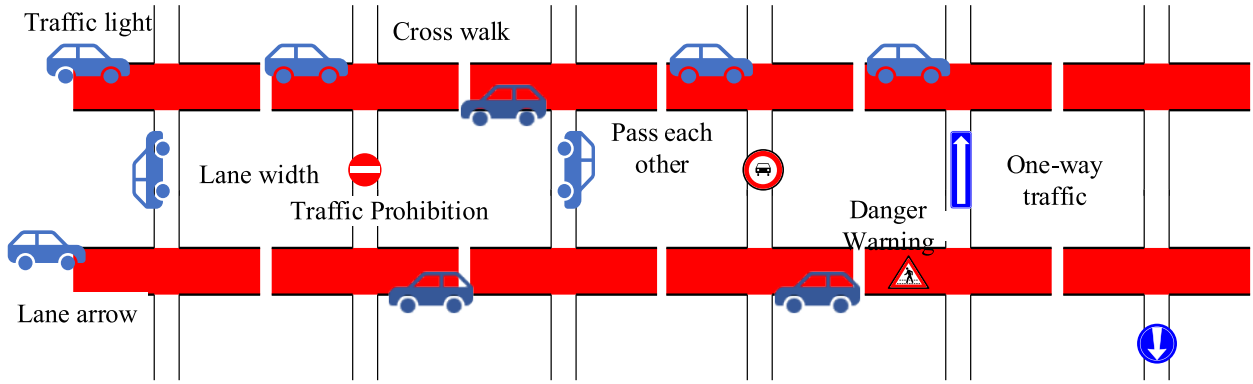


FIGURE 1. The scene of SLAM processing multiple traffic semantic information.

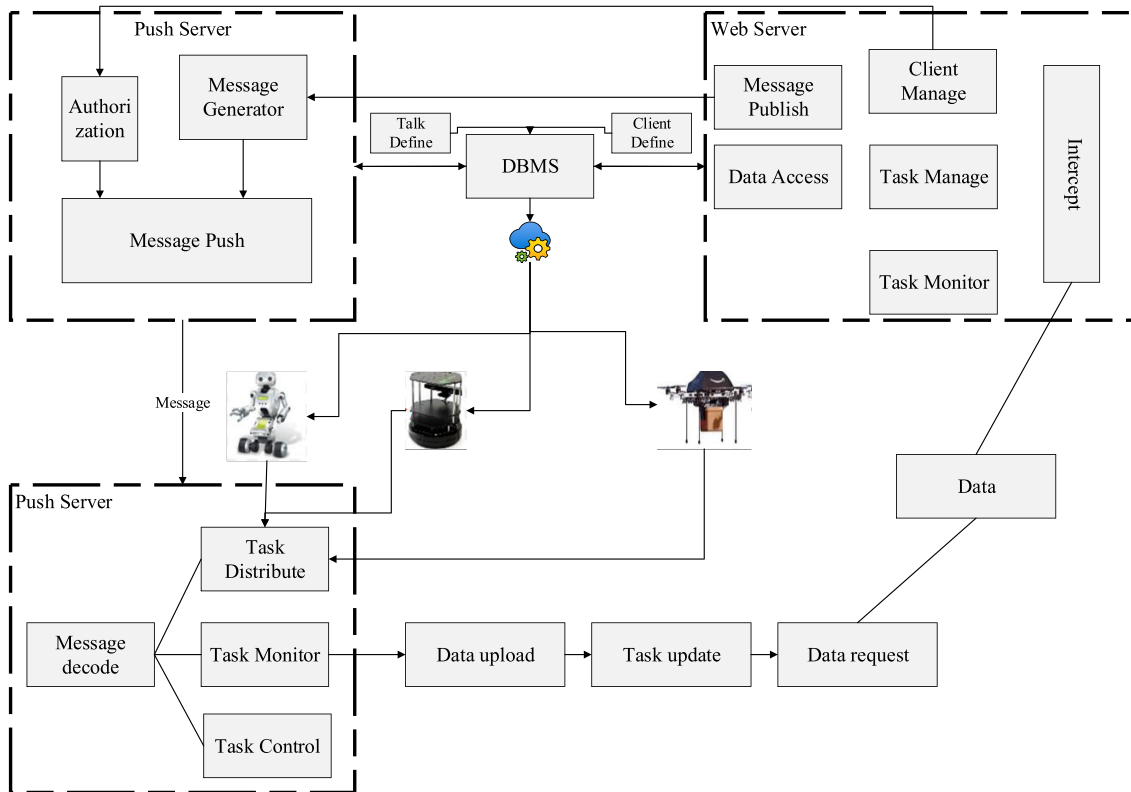


FIGURE 2. ROS computational structure diagram.

while locating the robot’s position in this map. SLAM has different implementations in different domains, including indoor, outdoor, aerial, and underwater robotic systems at the theoretical and conceptual levels.

B. MOTION MODEL OF MOBILE UNMANNED VEHICLES

Motion modeling is an essential part of the field of SLAM for unmanned vehicles and in the field of unmanned vehicle navigation. Modeling the unmanned vehicle according to its structure, motion and other factors is the main role of the motion model. An unmanned vehicle in a rigid traveling state

is generally described by six variables, including its 3D linear coordinates and six terms such as roll angle, yaw angle, and Euler angle. In this paper, the research environment assumes that the unmanned vehicle moves in a two-dimensional plane, and the motion state of the unmanned vehicle moving in a two-dimensional plane can be represented by a vector, which is shown as:

$$x_t = \begin{Bmatrix} x \\ y \\ \theta \end{Bmatrix} \tag{1}$$

Localization is the process of estimating the joint state vector of an unmanned vehicle. The location of the environment in which the unmanned vehicle is located can be represented as a coordinate point (x, y) in an absolute coordinate system. However, in unmanned vehicle simultaneous localization and map creation studies, the unmanned vehicle cannot be abstracted as a physical mass point because the geometry of the unmanned vehicle is no longer negligible compared to the distance between the environmental features observed by the sensors. In other words, if the unmanned vehicle is abstracted as a physical mass, the solution to the SLAM problem will deviate from the actual value and lose any practical value and meaning. The direction of θ is defined as negative counterclockwise and positive clockwise in the absolute coordinate system X-axis [19]. In the solution of a problem such as SLAM, the kinematic model can be expressed as:

$$p(x_t | x_{t-1}, u_t) \quad (2)$$

In the equation, x_t and x_{t-1} represent the position of the unmanned vehicle at time t and the position of the unmanned vehicle at time $t - 1$, respectively, and u_t is the control quantity. The meaning of the equation is: in the actual physical experiment, u_t is the feedback data from the odometer, and the position attitude of the unmanned vehicle at the time $t - 1$. x_{t-1} is converted into a posteriori probability distribution under the action of the control quantity u_1 , which is the posterior distribution probability of the position attitude of the unmanned vehicle at time t .

There are two general representations of motion modeling in the plane. The kinematic models can be divided into velocity-based kinematic models and odometer-based kinematic models according to the different motion control tu . These two models have different application scenarios, and the two can compensate for each other's shortcomings. Most commercial unmanned vehicles are driven by independent translational and rotational velocities, such as differential and synchronous drives. The odometer model, on the other hand, refers to the ability of the unmanned vehicle to give information about the distance the unmanned vehicle is moving and its rotation in motion through its range sensors. Overall, the motion model tends to have high accuracy with the addition of odometry, and the commands of the unmanned vehicle may be incorrect at a certain time, for example, there will be cases where the car slides sideways, and mismatches between the speed commands and motor control commands [20].

After the unmanned vehicle executes the control command, the odometer gets the appropriate data back, so there is the disadvantage of lag in path planning. On the contrary, the speed model, which is mainly calculated by speed, can be well applied in the real-time motion calculation of the unmanned vehicle. The vehicle model can be simplified to a two-degree-of-freedom model: degrees of freedom for the vehicle to move along the longitudinal direction and degrees of freedom for the front wheels to rotate along an axis perpendicular to the vehicle's plane of motion. Thus, the two front

wheels and the two rear wheels can each be equated to a single wheel to obtain a bicycle model consisting of only two rigidly connected wheels moving in the plane. Since the assumption of no sideslip is proposed, the velocity of the vehicle in the orthogonal direction of the front and rear wheels should be 0. For the rear wheels of the vehicle, the following constraint equation is obtained by this assumption:

$$\eta = \frac{S_k}{x} + c \quad (3)$$

where η is the rear wheel position, S_k is the equation of motion, and c is a constant.

The following relationship exists between the velocity of the front wheel and the velocity of the rear wheel:

$$v_f = v_r \cdot \sum_{i=1}^n \frac{(x_i - y_i)}{t} \quad (4)$$

where v_f denotes the front wheel velocity and v_r denotes the rear wheel velocity. Unlike a general mobile robot, the vehicle has three degrees of freedom (x, y, ϕ) in the plane of motion, i.e., transverse motion, longitudinal motion, and transverse pendulum motion, but the vehicle has only two controllable degrees of freedom, i.e., the longitudinal velocity v in $m - s^{-1}$ and the front wheel rotation angle δ in rad, so the vehicle is a non-integrity constrained system. Since it cannot move in the orthogonal direction along the vehicle axis direction, it cannot achieve arbitrary velocity components in the x -direction as well as the y -direction with a fixed attitude, and therefore cannot travel in any direction. The bicycle model is the most widely used in the vehicle non-integrity constraint. The vehicle dynamics model has many complex variables affecting factors and complexity can be divided into vehicle driving dynamics, vehicle driving dynamics, and vehicle maneuvering dynamics according to the research content and different assumptions. Since the tire direction is not the same as the vehicle's forward direction, it will cause the tire to generate a small slip, and the slip angle is defined as follows:

$$a_f = \delta - \theta_{vf} \quad (5)$$

where θ_{vf} is the directional angle of the front wheel velocity in rad, and δ is the front wheel turning angle in rad. Thus the lateral forces on the front and rear wheels are as follows:

$$F_{yr} = 2C_{af}(-\theta_{vf}) \quad (6)$$

$$F_{yf} = 2C_{af}(-\theta_{vf}) \quad (7)$$

where C_{vf} is the front wheel lateral deflection stiffness and C_{af} is the rear wheel lateral deflection stiffness, both in $N - rad^{-1}$, as both the front and rear wheels in the model are simplified from two wheels. The ordinary Kalman filter (KF) is used to obtain a dynamic estimate of the target in the linear Gaussian case using the minimum mean square error, and the method is suitable for systems where both the state and measurement models are linear and the error model conforms to a Gaussian distribution. For systems with a nonlinear relationship between the state equation or the measurement and the state, the nonlinear relationship is often

solved by transforming it into a linear problem using a local linear approximation.

Among the solutions to SLAM problems, Kalman filter (EKF)-based estimation methods have been widely used as the most fundamental solution of SLAM problems. EKF is more applied in the field of sensor data fusion, especially fusion localization. EKF generally uses two nonlinear functions to describe the state transfer and measurement processes.

$$f_\phi(p, x) = d_p \left(x - \frac{\lambda_{\theta_1}}{d_p(x)} \right) + d_p \left(x - \frac{\lambda_{\theta_2}}{d_p(x)} \right) \quad (8)$$

$$f_\phi(p, y) = d_p \left(y - \frac{\lambda_{\theta_y}}{d_p(y)} \right) + d_p \left(y - \frac{\lambda_{\theta_y}}{d_p(y)} \right) \quad (9)$$

where $f_\phi(p, x), f_\phi(p, y)$ are the two nonlinear functions used to describe the transient transfer and measurement process, $d_p(y), d_p(x)$ are the state quantity and observation quantity of the vehicle at the moment of p , respectively. Extended Kalman filtering uses the linear Kalman filtering method applied to the nonlinear model with local linearization processing, so extended Kalman filtering is the linear Kalman filtering method to cope with the nonlinear system model processing. The extended Kalman filter, like the Kalman filter, divides the filtering process into two processes: state volume prediction and observation update.

The difference is that the extended Kalman filter replaces the system parameters of the KF with a linear matrix. Although Kalman filtering provides a solution for filtering nonlinear systems, it also has the following shortcomings: Extended Kalman filtering assumes that the noise and state distributions obey Gaussian distributions, an assumption that is poorly approximated in practical applications, unless the nonlinear system is continuously close to linearity, the results of extended Kalman filtering will be close to the true value. In the process of local linearization of the nonlinear system using Taylor's formula after expansion only the first-order term is retained, which will introduce errors [21]. The extended Kalman filter may prevent the filtering results from converging when the error in the given initial state and initial covariance is large. The ratio of the proposed distribution to the posterior probability density function (target distribution) indicates how close the target distribution is to the proposed distribution. As more samples are sampled in the sample space, the proposed distribution will gradually approach the target distribution, and this ratio also affects the likelihood that the next sampled particle will be collected. The weights are derived to the vehicle state sequence and sensor data observation sequence from the initial moment t_1 for analysis, and iteratively find the weight changes of particles at different moments:

$$R^m = \frac{R_i^m}{\sum_{i=1}^N R_i^m} \quad (10)$$

In the process of particle computation, the distribution of weights may be polarized, with a small number of particles receiving larger weights and the rest having weights so small

that they can be approximately ignored. The variance of the particle weights increases with time, and the number of effective particles in the state space is small. The problem of degradation of particle weights in sequential importance sampling methods is generally solved by increasing the number of sample particles, with the consequent increase in the computational effort of the algorithm, making it less real-time. The resampling method replaces the particles with smaller importance weights with particles with larger importance weights to suppress the degradation of weights, which can maintain the real-time performance of the algorithm without increasing the number of particles. The algorithm principle of particle filtering can be represented in Figure 3. First, the particle set is initialized according to the prior distribution to generate the sampled particles. Importance weights are generated for each particle, and the second generation of sampled particles is generated by sampling the importance of the initial particles according to the importance weights, and whether to resample them according to the number of valid particles, and the state estimates are output after iterative computation.

The joint posterior probability distribution problem is solved by first obtaining the vehicle position information based on the vehicle observation and control input information to achieve the vehicle's positioning, and then constructing a global map based on the vehicle's own positioning and sensor observation information. In the process of map construction, the map constructed at each moment of the vehicle position is only related to the vehicle position and observation information independent of time, so the maps built at each moment of the vehicle are conditionally independent. The global map posterior probability estimates can be obtained by calculating the product of the posterior probability estimates at each moment, corresponding to the particle filter where each sample particle set represents the map information at this location:

$$E_{all} = \sum_{i=0}^R (E_{contest} \times x_t + \varepsilon_{free} \times x_t^2 \times d_t^{\frac{1}{2}}) + \sum_{i=0}^K (4E_e \times x_t) \quad (11)$$

Particles with low-importance weights are usually replaced by sample particles with high weights. Since the number of particles is discrete and finite, the resampling process is essential to enable the particle distribution to approximate a continuous distribution. To improve the proposed distribution, after the odometer motion model gives a prediction, a scan is matched using the LIDAR with this prediction as the initial value to find the current bit position that best fits the map. This process is followed by finding the narrow region represented by the observation model. Next, k points are randomly sampled in the vicinity of the observation model and the target distribution is simulated using a Gaussian model:

$$F(\rho) = \eta_K \int k^{-i2\pi\omega x} m(z) f(x) dx \quad (12)$$

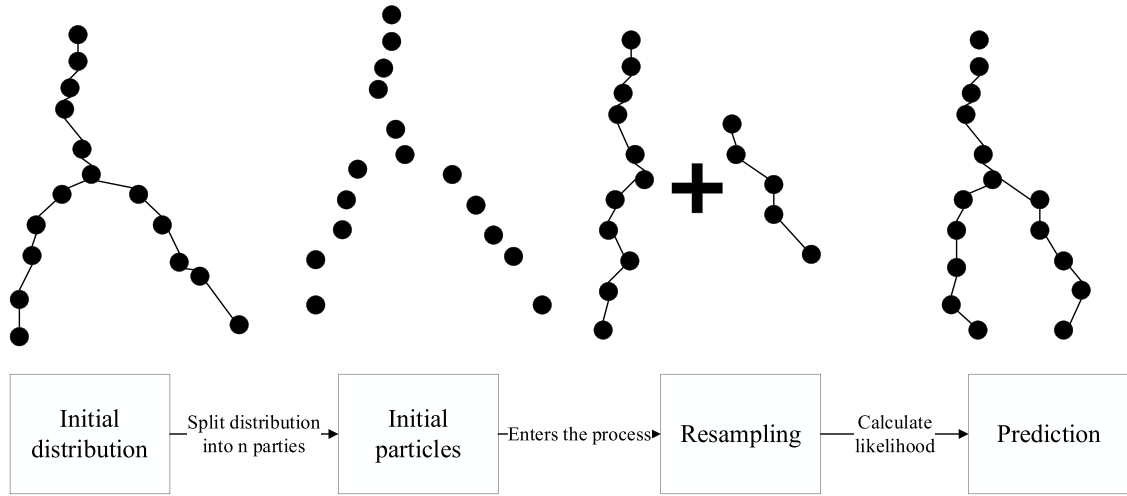


FIGURE 3. Schematic diagram of particle filtering.

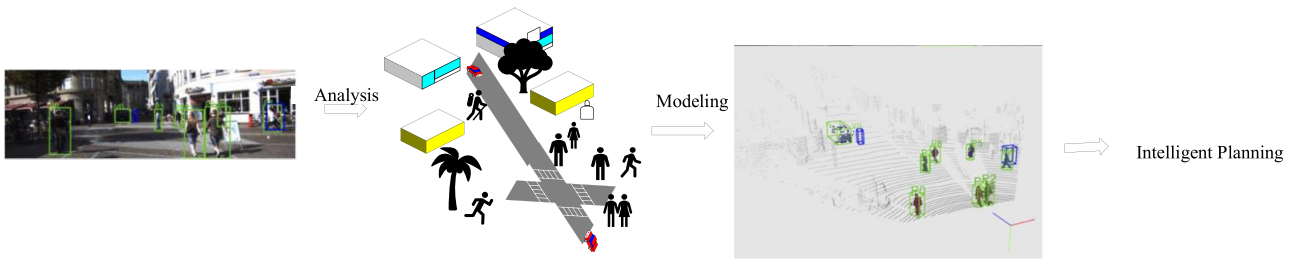


FIGURE 4. Recognition effect of SLAM-based vehicle navigation scenarios.

where the normalization factor η_k has:

$$\eta_k = F_y \cdot \int A \cdot M(x) dx \quad (13)$$

In this way, the best approximation to the Gaussian distribution can be obtained, the next moment particles can be sampled from the Gaussian distribution, and the particle weights at each moment can be established with corresponding iterative relations. When the vehicle is exploring forward in an unknown environment, there is still a high diversity and accuracy of particles after improving the proposed distribution if no closed loop is formed. As mentioned before, too frequent resampling can lead to the loss of particle diversity, and resampling changes the particle distribution in the state space by increasing the weights of particles near the target distribution. The effective number of particles can be calculated by following the steps above as

$$N = \beta e^{-at^2} \cos(2\pi f_c t) \quad (14)$$

The sub-maps are only matched to the most recent laser scan data. So as time progresses errors will be accumulated, at which point a larger space is handled by creating more and smaller sub-maps [22]. Adjustments are made to follow sparse poses, optimizing the information from all scans and sub-maps, and the relative poses of the scanned information

are stored in memory for back-end loopback optimization. In addition to these relative poses, once the sub-map is no longer changing, a closed-loop loopback is performed using the laser scan data and the sub-map, and once the two are detected to be a good match, the resulting relative poses are added to the optimization [23]. Figure 4 shows the recognition effect of SLAM-based vehicle navigation scenarios.

IV. EXPERIMENTAL VERIFICATION AND CONCLUSION

A. EXPERIMENTAL RESULTS AND ANALYSIS

To test the control effect of the model-based following method, this paper uses the Carla simulation platform to set up the following scenario. The front vehicle guides the rear vehicle at a gentle speed (30-40km/h) and a drastic speed change (10-60km/h), respectively. The initial distance between the front and rear vehicles is 10 m, and the fixed following distance is also 10 m. Carla transmits frames per second (fps) at 15. The model is set up assuming that the vehicle accelerates and decelerates at a constant speed. However, the Carla simulator uses the throttle and brake to control the acceleration and deceleration of the vehicle, which are both variable acceleration motions [24]. The front vehicle is guided by a gentle speed change (0-40km/h) and a sharp speed change 40-90km/h). The relative distance changes between the vehicles during the following process of different

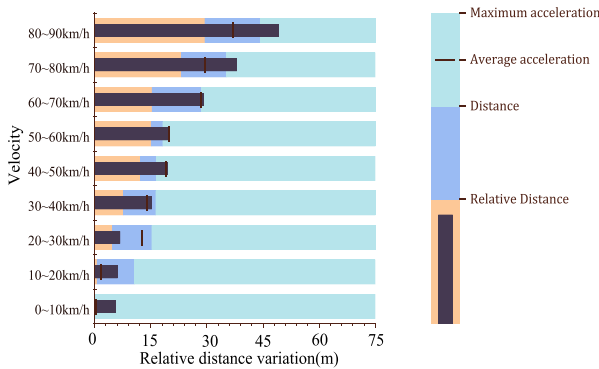


FIGURE 5. Relative distance variation between vehicles during the following process of different methods of controlling the rear vehicle.

methods of controlling the rear vehicle are recorded, as shown in Figure 4. From the relative distance change graph, it can be seen that the regulation process of the minimum safe distance method is relatively smooth, the methods of the three stages and the intelligent driving model are more sensitive to the speed change of the front vehicle, and the method of the fixed target distance does not change much.

From the speed curve, it can be seen that all methods fluctuate and change around the speed curve of the vehicle in front, the minimum safe distance method fluctuates the least, while the fixed target distance method fluctuates the most. In general, when the speed of the front vehicle changes more, the collision time of the vehicle becomes smaller, which means there is a greater risk of collision; the interval time through the same place becomes longer, which means the driving efficiency becomes lower; the jitter of the vehicle is determined by the frequency of the vehicle speed change when the vehicle speed is faster but the change is not frequent, the vehicle jitter is smaller, which means the comfort level is high. The intelligent driving model approach has the best safety with crash time metrics of 33.45 ± 12.96 and 4.43 ± 3.80 at the cost of the lowest travel efficiency 80~90km/h and 0~10km/h respectively.

To evaluate the effect of parameter settings on the control effect of the conventional follow-the-ride control method. Set different PID control parameters for fixed target distance and parameters of the Intelligent Driving Model IDM method. The change curves of the relative distance between the front and rear vehicles and the speed of the rear vehicle were measured again for both methods, in a scenario where the speed of the front vehicle changes gently (30-40km/h), as shown in Figure 5. The dynamic window method has good obstacle avoidance capability by planning the local path based on the obstacle information detected during the navigation of the unmanned vehicle [25]. However, the method ignores the limitation of the global path, so it has the fatal problem of easily falling into the local optimum. The test results show that the parameter settings have a great influence on the traditional follow-the-route control method.

From the above experimental results, it can be seen that the traditional follow-the-leader control method requires preset model parameters. The same model parameters in different scenarios lead to the degradation of control performance, i.e., the control method has poor adaptive capability. Under the same scenario, the control effect varies greatly with different sets of model parameters, and finding a suitable set of model parameters requires a lot of experience in tuning parameters, i.e., the accuracy of the control method is low [26]. For any set of designs, the more expert experience is added to the training ground, the faster the cumulative bonus value converges per round. Different random seeds, i.e., different variations in the speed of the preceding vehicle, have little effect on the cumulative reward value. It indicates that the following strategy is available to the intelligences in different following scenarios and the stability of the training strategy is good. The hyperparameter settings of the algorithm part are consistent with the experiments based on the image state space.

B. SIMULATION COMPARISON OF PATH CURVATURE CHARACTERISTICS

To analyze the path nature and the influence of the path tracking method on vehicle control accuracy, a three-degree-of-freedom vehicle dynamics model is first built using a Simulink simulation environment [27]. However, the vehicle dynamics model does not take into account the hysteresis problem of the vehicle steering actuator. To make the simulation results closer to the actual situation, a limiting link is added before the vehicle front wheel angle calculation step to limit the front wheel speed of the vehicle. Firstly, the simulation is compared for the effect of whether the path curvature is continuous on the control accuracy, and the test scenario uses the limit size parking space of the vertical parking space in the garage specification of the Ministry of Housing and Construction. For the effect of continuous curvature path and discontinuous curvature path on vehicle control accuracy, a curvature discontinuous path is directly planned in this vertical parking space using the improved RRT algorithm, and a curvature continuous path is generated using the optimization method.

The same path-tracking method is then used for each of the two paths to compare the accuracy of the final path-tracking [28]. Firstly, taking the rear-wheel feedback method as an example, the paths were tracked using the rear-wheel feedback method at a speed of 1.5m/s and 3m/s, respectively, and the paths of the curvature discontinuous paths were tracked as shown in Figure 6. It can be seen from the results that under the same control algorithm parameters, the error of the curvature discontinuous path is always larger than the error of the curvature continuous path regardless of the speed conditions, and the error of the curvature discontinuous path increases rapidly with the increase of speed, while the error of the curvature continuous path is not significantly affected by the speed in the experimental conditions. And in the case of 3m/s in Figure 6, the vehicle collided with the left parking space, and finally could not be accurately parked in

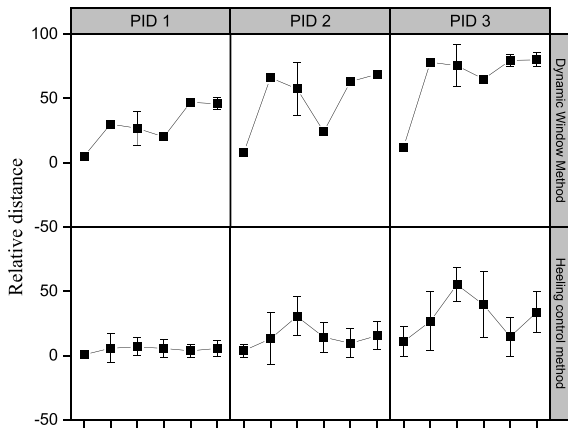


FIGURE 6. Curves of the relative distance between vehicles before and after the smoothing scene and the speed of the rear vehicle.

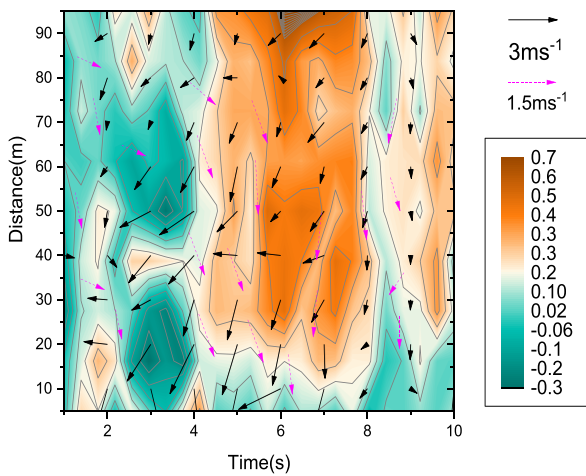


FIGURE 7. Curvature discontinuous path tracking results.

the correct position of the parking space. By comparing the results of the two experiments, it can be seen that curvature continuity is extremely helpful to improve vehicle control accuracy.

The traditional optimization-based method for path planning of parking lot parking, but does not consider the curvature continuity problem, and it does not stop or decelerate at the curvature abrupt change position, the simulation experiment shows that the path planned by the method in this paper can achieve higher control accuracy [29]. The detailed results of the error of the path tracking method using rear-wheel feedback for different paths are shown in Figure 7. By comparing the above pictures, it can be seen that the control accuracy of both lateral error and heading error curvature continuous path is significantly improved. As can be seen from the resulting graph of the curvature discontinuous path, the effect of curvature discontinuity leads to the sudden change of the expected front wheel turning angle, and the vehicle steering system cannot immediately reach the expected control amount, which results in a large difference between the curve of the actual front wheel turning angle

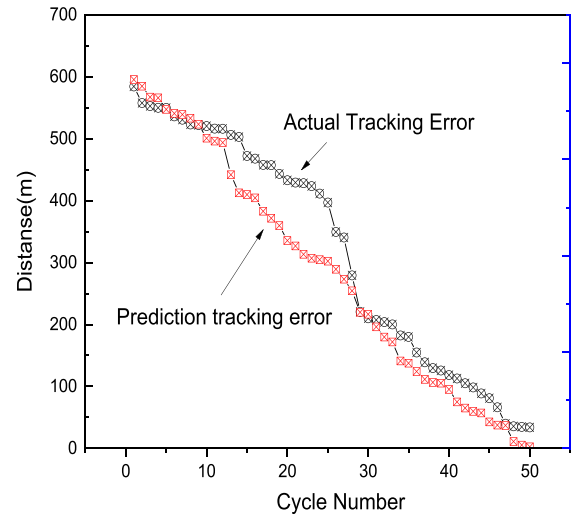


FIGURE 8. Curvature discontinuous path tracking error.

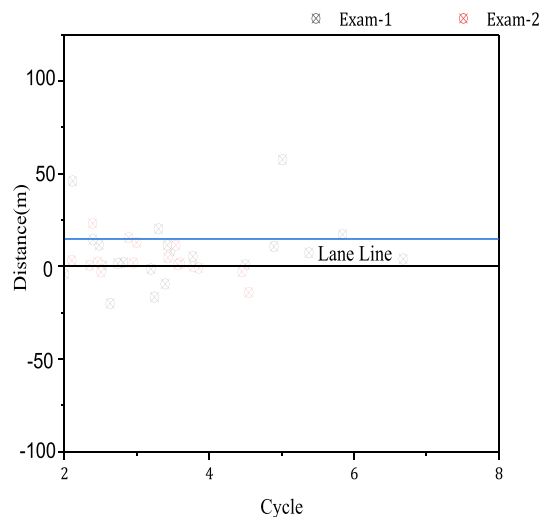


FIGURE 9. Path planning test for the irregular road surface.

and the curve of the expected front wheel turning angle, thus leading to the error. In contrast, the results of the curvature continuous path shown in Figure 7 have a high degree of overlap between the expected front wheel turning angle and the actual front wheel turning angle and the control output fluctuates slightly at the equilibrium position, so the control accuracy is high.

However, there is still a certain difference between the two curves at the moment of forward and reverse switching, however, the vehicle speed is very low at this time, so it does not introduce a large error, but still causes control deviation [30]. The overall results show that a path with continuous curvature can significantly improve the accuracy of vehicle path tracking. It can be seen from Figure 8 that the system can also complete the path planning well on irregular road surfaces and realize the lane line tracking driving. Collating and analyzing the data, we can get that the maximum error value is 53-pixel values, which appears near the curves. The

width of the lane line is about 25 m, so the maximum error rate is 20.38%. Calculating all the data, the average error in the 30s is 3.9 m, so the average error rate is 5.35%. The test results show that the unmanned vehicle can accurately extract the navigation lines and accurately identify the curve types after increasing the filtering. Although the error is larger at the junction of straight and curved paths of lane lines, it is smoother when going straight or turning, and the average error rate is stable at about 5%.

V. CONCLUSION

The automotive industry is undergoing profound and dramatic changes as the electrification, intelligence, networking and sharing of cars become widespread around the world. Driverless is also a trend that is also changing and developing in this environment. In this paper, we investigate the application of SLAM algorithms for mapping and navigation. The integration of the more common low-cost sensor Kinect depth camera and 2D laser radar at the data level improves the accuracy of Gmapping mapping. In this paper, a probabilistic kinematic model of an unmanned cart is developed, and the odometry model and velocity motion model are compared and analyzed. The working principles of depth camera and single-line LiDAR are analyzed, and then the observation model of 2D LiDAR is established.

Two RGB-D camera principles were studied for depth camera calibration experiments, and the correspondence between environmental spatial coordinate points and image pixel points was established. A two-dimensional raster map model was established, and the process of predicting and updating the two-dimensional occupied raster state was derived and analyzed. Based on the constructed probabilistic kinematic model and sensor observation model, a Bayesian filtering scheme based on Bayesian theory, a Gaussian filtering scheme based on extended Kalman filtering, and a nonparametric filtering scheme based on particle filtering principle are analyzed in conjunction with the unmanned vehicle. For the problem that the traditional model-based lane line detection algorithm has a poor detection effect in the complex and changing road environment, the up-convex curve model lane line detection algorithm is proposed. The algorithm is suitable for lane line detection under a variety of road conditions, with better detection capability and robustness, and improves the shortcomings of linear and parabolic model lane line detection.

REFERENCES

- [1] Z. Guo, K. Yu, A. Jolfaei, G. Li, F. Ding, and A. Beheshti, "Mixed graph neural network-based fake news detection for sustainable vehicular social networks," *IEEE Trans. Intell. Transp. Syst.*, early access, Jul. 7, 2022, doi: 10.1109/TITS.2022.3185013.
- [2] N. Wang, Y. Zhang, C. K. Ahn, and Q. Xu, "Autonomous pilot of unmanned surface vehicles: Bridging path planning and tracking," *IEEE Trans. Veh. Technol.*, vol. 71, no. 3, pp. 2358–2374, Mar. 2022.
- [3] Y. Wu, K. H. Low, and C. Lv, "Cooperative path planning for heterogeneous unmanned vehicles in a search-and-track mission aiming at an underwater target," *IEEE Trans. Veh. Technol.*, vol. 69, no. 6, pp. 6782–6787, Jun. 2020.
- [4] Z. Guo, K. Yu, K. Konstantin, S. Mumtaz, W. Wei, P. Shi, and J. J. P. C. Rodrigues, "Deep collaborative intelligence-driven traffic forecasting in green Internet of Vehicles," *IEEE Trans. Green Commun. Netw.*, vol. 7, no. 2, pp. 1023–1035, Jun. 2023.
- [5] H.-Y. Lin and X.-Z. Peng, "Autonomous quadrotor navigation with vision based obstacle avoidance and path planning," *IEEE Access*, vol. 9, pp. 102450–102459, 2021.
- [6] A. Basiri, V. Mariani, G. Silano, M. Aatif, L. Iannelli, and L. Glielmo, "A survey on the application of path-planning algorithms for multi-rotor UAVs in precision agriculture," *J. Navigat.*, vol. 75, no. 2, pp. 364–383, Mar. 2022.
- [7] Z. Zhao, B. Zhu, Y. Zhou, P. Yao, and J. Yu, "Cooperative path planning of multiple unmanned surface vehicles for search and coverage task," *Drones*, vol. 7, no. 1, p. 21, Dec. 2022.
- [8] M. Li, J. Mou, Y. He, X. Zhang, Q. Xie, and P. Chen, "Dynamic trajectory planning for unmanned ship under multi-object environment," *J. Mar. Sci. Technol.*, vol. 27, no. 1, pp. 173–185, Mar. 2022.
- [9] D. Zhu and S. X. Yang, "Path planning method for unmanned underwater vehicles eliminating effect of currents based on artificial potential field," *J. Navigat.*, vol. 74, no. 5, pp. 955–967, Sep. 2021.
- [10] A. A. Saadi, A. Soukane, Y. Meraihi, A. B. Gabis, S. Mirjalili, and A. Ramdane-Cherif, "UAV path planning using optimization approaches: A survey," *Arch. Comput. Methods Eng.*, vol. 29, no. 6, pp. 4233–4284, Oct. 2022.
- [11] B. Zhou, J. Yi, X. Zhang, L. Chen, D. Yang, F. Han, and H. Zhang, "An autonomous navigation approach for unmanned vehicle in outdoor unstructured terrain with dynamic and negative obstacles," *Robotica*, vol. 40, no. 8, pp. 2831–2854, Aug. 2022.
- [12] M. H. Garibeh, A. M. Alshorman, M. A. Jaradat, A. B. Younes, and M. Khaleel, "Motion planning of unmanned aerial vehicles in dynamic 3D space: A potential force approach," *Robotica*, vol. 40, no. 10, pp. 3604–3630, Oct. 2022.
- [13] Z. Wang, G. Li, and J. Ren, "Dynamic path planning for unmanned surface vehicle in complex offshore areas based on hybrid algorithm," *Comput. Commun.*, vol. 166, pp. 49–56, Jan. 2021.
- [14] B. Tong, H. Duan, and L. Chen, "A path planning method for UAVs based on multi-objective pigeon-inspired optimisation and differential evolution," *Int. J. Bio-Inspired Comput.*, vol. 17, no. 2, p. 105, 2021.
- [15] D. Zhu and S. X. Yang, "Bio-inspired neural network-based optimal path planning for UAVs under the effect of ocean currents," *IEEE Trans. Intell. Vehicles*, vol. 7, no. 2, pp. 231–239, Jun. 2022.
- [16] J. Zhao, J. Du, B. Zhu, Z. Wang, Z. Chen, and X. Tao, "Intelligent vehicle longitudinal cruise control based on adaptive dynamic sliding mode control," *Automot. Eng.*, vol. 44, no. 1, pp. 8–16, 2022.
- [17] X. Xu, P. Cai, Z. Ahmed, V. S. Yellapu, and W. Zhang, "Path planning and dynamic collision avoidance algorithm under COLREGs via deep reinforcement learning," *Neurocomputing*, vol. 468, pp. 181–197, Jan. 2022.
- [18] X. Tao, P. Meng, B. Zhu, and J. Zhao, "A fast and accurate camera-IMU calibration method for localization system," *Insight-Autom. Control*, vol. 6, no. 1, p. 578, Jul. 2023.
- [19] S. Xie, J. Hu, P. Bhowmick, Z. Ding, and F. Arvin, "Distributed motion planning for safe autonomous vehicle overtaking via artificial potential field," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 11, pp. 21531–21547, Nov. 2022.
- [20] B. Zhu, Z. Wang, J. Zhao, and X. Tao, "Automatic parking space recognition method based on laser radar bounding boxes," in *Proc. Annu. Conf. China Soc. Automot. Eng.*, vol. 1, 2021, pp. 157–168.
- [21] M. Moussa and G. Beltrame, "Real-time path planning with virtual magnetic fields," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 3279–3286, Apr. 2021.
- [22] Z. Bing, T. Xiao-Wen, Z. Jian, K. Min, W. Zhi-Wei, and L. Xin, "Two-stage UWB positioning algorithm of intelligent vehicle," *J. Traffic Eng.*, vol. 21, no. 2, pp. 256–266, 2021.
- [23] Z. Hu, X. Gao, K. Wan, Y. Zhai, and Q. Wang, "Relevant experience learning: A deep reinforcement learning method for UAV autonomous motion planning in complex unknown environments," *Chin. J. Aeronaut.*, vol. 34, no. 12, pp. 187–204, Dec. 2021.
- [24] B. Zhu, X. Tao, J. Zhao, M. Ke, H. Wang, and W. Deng, "An integrated GNSS/UWB/DR/VMM positioning strategy for intelligent vehicles," *IEEE Trans. Veh. Technol.*, vol. 69, no. 10, pp. 10842–10853, Oct. 2020.
- [25] H. Lu, Q. Zong, S. Lai, B. Tian, and L. Xie, "Real-time perception-limited motion planning using sampling-based MPC," *IEEE Trans. Ind. Electron.*, vol. 69, no. 12, pp. 13182–13191, Dec. 2022.

[26] X. Li, X. Tao, B. Zhu, and W. Deng, "Research on a simulation method of the millimeter wave radar virtual test environment for intelligent driving," *Sensors*, vol. 20, no. 7, p. 1929, Mar. 2020.

[27] X. Bai, B. Li, X. Xu, and Y. Xiao, "A review of current research and advances in unmanned surface vehicles," *J. Mar. Sci. Appl.*, vol. 21, no. 2, pp. 47–58, Jun. 2022.

[28] X. Tao, B. Zhu, S. Xuan, J. Zhao, H. Jiang, J. Du, and W. Deng, "A multi-sensor fusion positioning strategy for intelligent vehicles using global pose graph optimization," *IEEE Trans. Veh. Technol.*, vol. 71, no. 3, pp. 2614–2627, Mar. 2022.

[29] P. Petráček, V. Krátký, M. Petrlík, T. Báca, R. Kratochvíl, and M. Saska, "Large-scale exploration of cave environments by unmanned aerial vehicles," *IEEE Robot. Autom. Lett.*, vol. 6, no. 4, pp. 7596–7603, Oct. 2021.

[30] J. Zhao, J. Du, B. Zhu, X. Luo, and X. Tao, "Indirect tire pressure monitoring method based on the fusion of time and frequency domain analysis," *Measurement*, vol. 220, Oct. 2023, Art. no. 113282.

[31] Z. Guo, Q. Zhang, F. Ding, X. Zhu, and K. Yu, "A novel fake news detection model for context of mixed languages through multiscale transformer," *IEEE Trans. Computat. Social Syst.*, early access, Aug. 21, 2023, doi: [10.1109/TCSS.2023.3298480](https://doi.org/10.1109/TCSS.2023.3298480).

[32] L. Yang, Y. Li, S. X. Yang, Y. Lu, T. Guo, and K. Yu, "Generative adversarial learning for intelligent trust management in 6G wireless networks," *IEEE Netw.*, vol. 36, no. 4, pp. 134–140, Jul./Aug. 2022.

[33] L. Zhao, Z. Bi, A. Hawbani, K. Yu, Y. Zhang, and M. Guizani, "ELITE: An intelligent digital twin-based hierarchical routing scheme for software-defined vehicular networks," *IEEE Trans. Mobile Comput.*, vol. 22, no. 9, pp. 5231–5247, Sep. 2023, doi: [10.1109/TMC.2023.3179254](https://doi.org/10.1109/TMC.2023.3179254).

[34] Q. Li, L. Liu, Z. Guo, P. Vijayakumar, F. Taghizadeh-Hesary, and K. Yu, "Smart assessment and forecasting framework for healthy development index in urban cities," *Cities*, vol. 131, Dec. 2022, Art. no. 103971.

[35] Z. Zhou, Y. Li, J. Li, K. Yu, G. Kou, M. Wang, and B. B. Gupta, "GAN-Siamese network for cross-domain vehicle re-identification in intelligent transport systems," *IEEE Trans. Netw. Sci. Eng.*, vol. 10, no. 5, pp. 2779–2790, Sep./Oct. 2022, doi: [10.1109/TNSE.2022.3199919](https://doi.org/10.1109/TNSE.2022.3199919).

[36] X. Yuan, Z. Zhang, C. Feng, Y. Cui, S. Garg, G. Kaddoum, and K. Yu, "A DQN-based frame aggregation and task offloading approach for edge-enabled IoMT," *IEEE Trans. Netw. Sci. Eng.*, vol. 10, no. 3, pp. 1339–1351, May 2023.

[37] Q. Zhang, Z. Guo, Y. Zhu, P. Vijayakumar, A. Castiglione, and B. B. Gupta, "A deep learning-based fast fake news detection model for cyber-physical social services," *Pattern Recognit. Lett.*, vol. 168, pp. 31–38, Apr. 2023, doi: [10.1016/j.patrec.2023.02.026](https://doi.org/10.1016/j.patrec.2023.02.026).

[38] Q. He, Z. Feng, H. Fang, X. Wang, L. Zhao, Y. Yao, and K. Yu, "A blockchain-based scheme for secure data offloading in healthcare with deep reinforcement learning," *IEEE/ACM Trans. Netw.*, early access, Jun. 5, 2023, doi: [10.1109/TNET.2023.3274631](https://doi.org/10.1109/TNET.2023.3274631).



YAPENG LIU was born in Dingzhou, Hebei, in 1980. She received the bachelor's degree in engineering from Hebei Normal University. She is currently an Associate Professor with the Hebei Vocational College of Labor Relations. Her research interests include mechanical electrical engineering and control engineering.



PANPAN DONG was born in Hebei, in 1987. She received the bachelor's and master's degrees in engineering from Yanshan University. She is currently a Lecturer with the Hebei Jiaotong Vocational and Technical College. Her research interests include vehicular technology, systems engineering, and artificial intelligence.



JIANWEI LIANG was born in Hebei, in 1980. He received the bachelor's degree in engineering from Hebei Normal University, in 2004, and the master's degree in engineering from Yanshan University, in 2009. He is currently an Associate Professor with the Department of Mechanical and Electrical Engineering, Shijiazhuang College of Applied Technology. He has participated in more than ten scientific research projects and published more than ten articles. His research interests include vehicle engineering and hybrid intelligent systems.



LU CHEN was born in Shijiazhuang, Hebei, in 1988. She received the bachelor's degree in engineering from Yanshan University and the master's degree in engineering from Shijiazhuang Tiedao University. She is currently a Lecturer with the Shijiazhuang College of Applied Technology. Her research interests include intelligent control, deep learning, and vehicular engineering.



AIBING WANG was born in Xingtai, Hebei, in 1979. He received the master's degree in engineering from Harbin Engineering University. He is currently a Professor with the Hebei Jiaotong Vocational and Technical College. His research interests include intelligent vehicles, autonomous unmanned systems, and computer vision.

...