IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 19, 2018, accepted August 19, 2018, date of publication September 3, 2018, date of current version October 17, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2868339

A Path Planning and Navigation Control System Design for Driverless Electric Bus

LINGLI YU, DECHENG KONG, XUANYA SHAO, AND XIAOXIN YAN
School of Information Science and Engineering, Central South University, Changsha 410083, China

Corresponding author: Decheng Kong (kongdecheng@csu.edu.cn)

This work was supported in part by the Major Projects of Science and Technology in Hunan under Grant 2017GK1010, in part by the National Key Research and Development Plan under Grant 2018YFB1201602, in part by the Natural Science Foundation of Hunan under Grant 2018JJ2531 and Grant 2018JJ2197, in part by the State Key Laboratory of Robotics and System (HIT) under Grant SKLRS-2017-KF-13, in part by the State Key Laboratory of Mechanical Transmissions of Chongqing University under Grant SKLMT-KFKT-201602, and in part by the National Natural Science Foundation of China under Grant 61403426.

ABSTRACT Extensive studies have been conducted on unmanned vehicle recently due to its bright prospect. However, few of them have studied the technology of driverless bus. In this paper, a path planning and navigation control system is designed for 12-m length driverless electric bus. The global path planning is adopted by the Dijkstra algorithm based on the analysis tool of ArcGIS. For local planning, a triple-layer finite-state machine is proposed to plan the driving maneuver, and it reduces computational complexity and improves driving effectiveness. Then, optimal trajectory is generated by a curve fitting technique based on path planning and driving maneuver, which takes full account of the safety and kinetics for driverless bus. In addition, we address the speed planning by expert rules to imitate an experienced driver. Finally, the trajectory tracking controller is devised by an adaptive rolling window. Moreover, a heading error compensation model and a self-adaptive strategy of controller parameters are set up to improve the control precision. Finally, simulation by TruckSim and scenes experiments verify the validity and practicability of our proposed path planning and navigation control system for driverless bus.

INDEX TERMS Adaptive rolling window, driverless electric bus, driving maneuver planning, trajectory generation, trajectory tracking control.

I. INTRODUCTION

Recently, unmanned vehicle technology is focused on a variety of autonomous missions in both civil [1] and military scenes [2]. In future, public transportation reduce some traffic accidents caused by human mistakes and enhance the traffic utilization [3]. Recently, most researchers study the autopilot by integrated navigation system, in-vehicle sensors, roadside ITS, traffic monitoring, vehicle-to-vehicle (V2V) and vehicle-to-infrastructure(V2I) communications [4]. The development for Defense Advanced Research Projects Agency (DARPA) challenges exhibits great autonomous driving performance [5]–[8]. In addition, lots of companies, such as Google, Baidu, Tesla and so on, focus on the unmanned vehicle. All of them help to promote the feasibility of self-driving technology and develop commercial autonomous vehicles.

Autonomous driving for a simple driving situation becomes a reality soon [9]–[11]. Although driverless technology has gradually stepped out of the laboratory, there are still many restrictions away from commercial feasible for the complexity and uncertainties of transportation. Nowadays, driverless bus can be competent for intercity bus with a fixed line and relatively simple road condition. At the same time, it reduces manpower to avoid the problem of ‘‘driving fatigue’’. Thus, it's feasible for driverless bus coming earlier than intelligent car.

A French company Local Motors introduced the Olli, a self-driving electric 12-seat bus in Fig. 1(a). On September 1, 2016, the world's first unmanned bus line was opened for trial operation in Dubai. The small electric unmanned bus is called EZ10, and it drove its inaugural passengers in a Bavarian town in Fig. 1(b). In Shenzhen of south China, four self-driving buses began trial operations in November 2017. The buses had a designed speed of 10 to 30 kph and began running on a 1.2-kilometer route in the bonded zone of Futian in Fig. 1(c).

Path planning provides safe and reliable maneuvers under various driving environment [12], which contains global path planning and local path planning. Global path planning provides a valid path and key sub-target for local path planning, including driving direction and destination. Meanwhile, local path planning is to avoid the static or dynamic obstacles [13]. The common methods of path planning including Artificial Potential Field [14], Cell Decomposition [11], Visibility

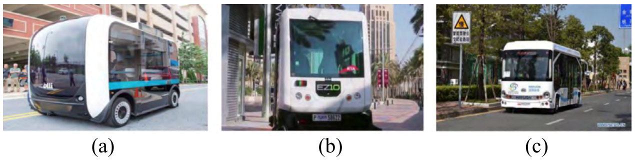


FIGURE 1. Several driverless buses in the world. (a) Olli. (b) EZ10. (c) Shenzhen bus.

graph [15], Voronoi Diagram [16] and Probabilistic Road Maps [17]. During global planning, the global route is determined by a digital map and a localization system [18]. Local path planning is divided into two stages: driving maneuver planning and trajectory generation. Driverless maneuver is determined by global planning and current sensor perception data [19], [20]. Multiple criteria decision making is applied to plan driving maneuver in [21]. An optimization path planner in [22] is capable of planning multiple contingency paths. But these algorithms are usually not applied in real driving scenarios well. However, FSM (Finite State Machine) is to express those sequences state of driving maneuvers [23]. In addition, the trajectory generation exists a variety of methods, including polynomial [24], Bezel curve [25] Spline curve [26], etc. For instance, 4th order polynomial and dynamic bicycle model are utilized to describe the land vehicle motion in [27].

Navigation Control is an essential issue for unmanned vehicle, which attracts lots of researchers and investigators [28]–[31]. PID controller, which is simple to design PID rules, is widely applied to engineering practice [32]. However, it is difficult to meet the precision requirements of unmanned vehicle control system with uncertain nonlinear dynamics. To improve control precision, it is better to predict the motion state. Principal Component Analysis (PCA) and Neural Networks (NN) are proposed to estimate the vehicle sideslip angle [33]. However, they require a large number of data to train parameters. Model predictive control (MPC) is an optimization-based time-domain control method [34], [35]. Compared with most controller, the distinct feature of MPC lies in its capability that predicts future control inputs by its knowledge of the plant dynamics and the desired output. In many MPC applications, the mathematical model for a system cannot be reliably determined. It is difficult to apply MPC to an engineering system [36]. In addition, model-free control approach is difficult to apply to an actual system in [37] and [38].

Navigation control for driverless bus brings greater challenge since bus system belongs to a great inertia and large time delay system. Therefore, a path planning and navigation control system for driverless bus is designed. Several major contributions listed below.

Firstly, in global planning stage, a digital map is built by ArcGIS which is used to establish road network and set up a series of road attributes. Then, global route from starting point to destination is planned by Dijkstra algorithm based on ArcGIS. In local planning stage, a triple-layer finite state machine is proposed for driving maneuver planning. Moreover, a cubic polynomial is used to express optimal trajectory which considers the safety and kinetics model.

Secondly, a lateral controller based on adaptive rolling preview window is designed to improve the control accuracy and reduce the lag time. Meanwhile, an error compensation model is used to predict the heading error for driverless bus. A self-adaptive parameter controller is devised to improve the robustness under different driving scenarios. Finally, a trapezoidal acceleration and deceleration longitudinal controller is designed to improve stability and comfort.

This paper is organized as follows. Section **II** introduces the system architecture of driverless electric bus including hardware and software. Section **III** presents map-building and global planning method. Local path planning and optimal trajectory generation are described in Section **IV**. Our trajectory tracking controller is designed in Section **V**. Finally, simulation and experiments are given in Section **VI** for the performance of driverless bus.

II. SYSTEM ARCHITECTURE OF THE DRIVERLESS ELECTRIC BUS

Driverless electric bus drives itself by recognizing driving environment and deciding maneuvers based on multiple sensors. A general hardware of the driverless electric bus and software framework are described below.

A. HARDWARE SYSTEM OF THE DRIVERLESS ELECTRIC BUS

The driverless electric bus and its sensor configuration are displayed in Fig. 2. The bus platform equipped with electric power steering (EPS), electronically controlled brake system (EBS) and motor-driven system. On-board sensors are used for perception to detect the surroundings, including vehicles, pedestrians, or other obstructions. Two lidars detect adjacent objects on both sides [39], [40], and two millimeter-wave (MMW) radars are installed on the bus head and tail respectively. To detect and classify objects, a color camera is installed in the middle of the windshield.

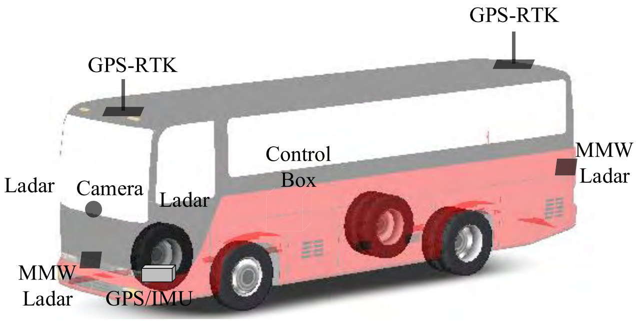


FIGURE 2. Hardware System of the bus platform.

An integrated navigation system, including Real-Time Kinematics GPS and Inertial Measurement Unit (IMU), is equipped to achieve precise localization and dead reckoning for driverless bus. The controllers of EPS, EBS, motor-driven system, perception and planning are integrated in a control box. Besides, the information of on-board sensors and controllers communicate through controller area network (CAN).

The hardware system architecture contains three components. The perception layer, including lidars, MMW radars and GPS/IMU offer the information on surrounding environments. perception controller transfers data through the CAN, and GPS/IMU data pass to planning layer through serial port. Planning and control layer collect the data from perception layer and vehicle state, then pass the calculation results to VCU actuators of the driverless bus.

B. SOFTWARE FRAMEWORK FOR PLANNING AND CONTROL LAYER

The planning and control layer provides four basic functions, including map-building, global planning, local planning and trajectory tracking control (shown in Fig. 3).

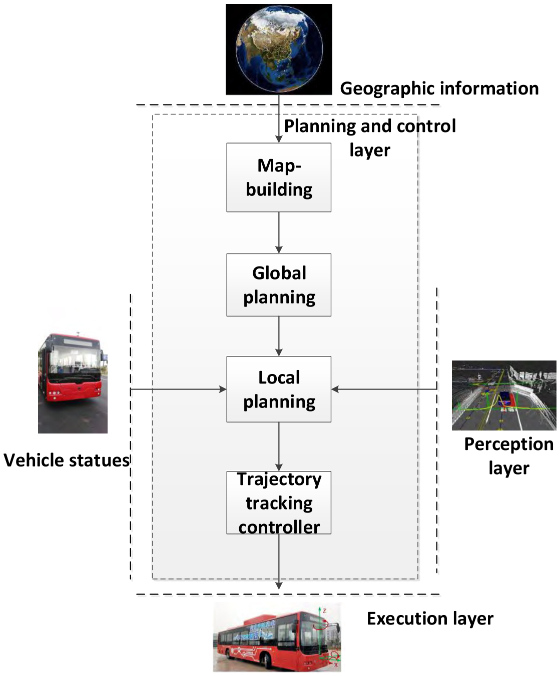


FIGURE 3. Software structure for planning and control layer.

(1) Map-building: A digital map for driverless bus provide navigation data of lane lines, road junctions, special positions, etc. The bus locates its position and desired path in the map. The method for building the map is based on ARCGIS. The raw data of GPS are employed to build road net, which represents the road geometry with centimeter-level accuracy through ArcGIS.

(2) Global planning: Global planning searches the shortest and safest path to reach the destination from initial position. In this stage, the built map and a searching algorithm is essential for fast routing.

(3) Local planning: Local planning is to plan driving maneuvers exactly and generate optimal trajectory in real time. Driving maneuver reasoning is a rule-based decision process based on triple-layer finite state machine. Additionally, bus kinematic model is considered to generate a safe and feasible trajectory.

(4) Trajectory tracking controller: The controller is to guide the driverless bus along the planned trajectory. The vehicle control should be robust under various driving conditions [41]. Therefore, bus states should be predicted for the specialty of the electric bus with great inertia and great lag. Besides, rolling window is applied to trajectory tracking control.

III. MAP-BUILDING AND GLOBAL PLANNING

A. ESTABLISHMENT OF ROAD NETWORK BASED ON ARCGIS

Map-building, one of key driverless technologies, which is crucial for positioning, navigation and control. In this paper, we construct the driverless bus map by ArcGIS and utilize the Dijkstra algorithm to plan global path.

ArcGIS offers a unique set of capabilities for applying location-based analysis, and gain greater insights using contextual tools to analyze and visualize data [42]. The map for driverless bus is built by collected GPS data. In this paper, a path is divided into two types which are general path and special path respectively.

(1) The general path is mostly straight. For a GIS map, a path contains edges and nodes. Raw GPS points of original path are connected by edges in order, and the distance between nodes is d, as shown in Fig. 4 (a).

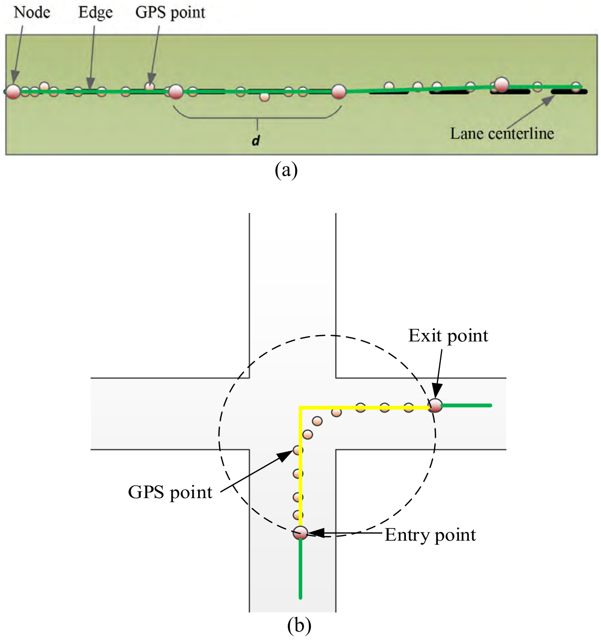


FIGURE 4. Map-building in ArcGIS. (a) General path. (b) Special path.

(2) The special path has a relatively large curvature, such as intersections and ramps. Besides, there is no lane line detected by perception layer. The collected GPS in intersections is saved in special path database. Meanwhile, special mark points are respectively set on two ends of special path, which are the entry point and exit point. They are the key points to distinguish general path and special path. The two points are connected with yellow straight lines to ensure the integrity of the road network for a GIS map (Fig. 4(b)).

Finally, GIS database and special path database are established. GIS database stores road net information created by ArcGIS, and special path database stores GPS data for intersections.

B. GLOBAL SHORTEST PATH PLANNING BASED ON DIJKSTRA ALGORITHM

The road net map consists of two basic elements: edges and nodes. The nodes connect each other by edges of topology network structure. And, the direction of road, markers and the other rules are all created in ArcGIS. The global planning is to search the optimal path from initial position to goal by such network topology.

Dijkstra algorithm is utilized to search the shortest path between nodes in a graph which is a searching algorithm for fast global routing [43]. Besides, ArcGIS supports network analysis which supports an effective shortest path analysis based on the topologic structure of road network, and its basic algorithm is Dijkstra algorithm. Thus, network analysis tool is applied to global path planning. Besides, instead of planning one fixed path to destination, global planning plans the path dynamically based on current location of the driverless bus in map.

C. DATA PROCESSING FOR THE GLOBAL PATH

The optimal path is obtained through global planning and the global reference points are processed to local planning.

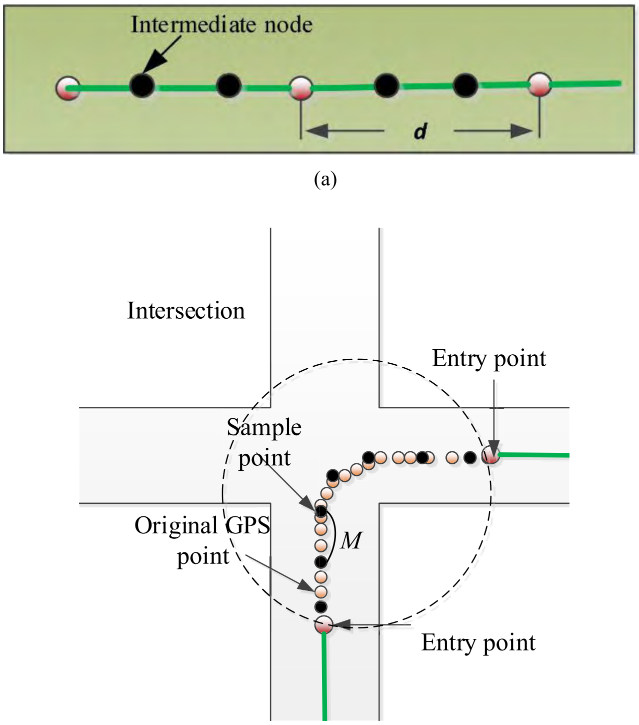
(1) The general path provides reference GPS nodes with distance d in GIS map. To improve precision, linear interpolation is implemented between pair of nodes and N intermediate nodes are inserted, as shown in Fig. 5 (a).

(2) The special path provides reference nodes by special path database. Moreover, these raw GPS points are sampled at sampling interval of M to reduce computational complexity in Fig. 5(b).

IV. LOCAL PATH PLANNING AND OPTIMAL TRAJECTORY GENERATION

A. DRIVING MANEUVER PLANNING BASED ON TRIPLE-LAYER FSM

Driving maneuver planning for driverless bus is to generate a reasonable behavior like an experienced driver. It makes decision according to the driving contexts, including global map, perception information and GPS/IMU system. FSM (Finite State Machine) is an efficient method to express those sequences of driving maneuvers. Detailed state design of FSM improves system intelligence, but the system is too complex, and the design process is much difficult. To facilitate modularity, a triple-layer FSM is used to evaluate feasible behavior in different driving scenes. A comprehensive planning architecture for driverless bus is shown in Fig. 6.



(a)

(b)

FIGURE 5. Data processing for global path. (a) general path. (b) special path.

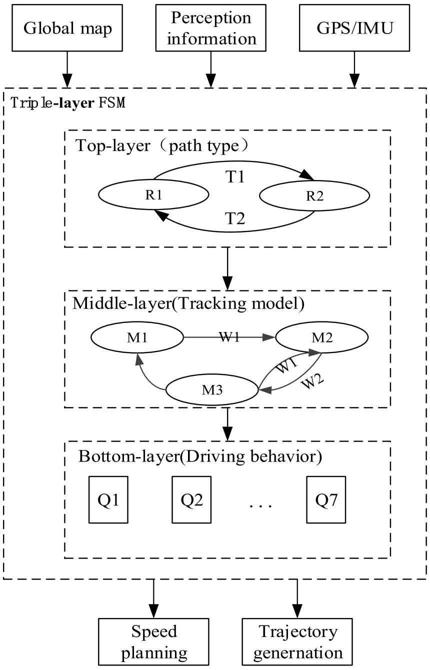


FIGURE 6. Triple-layer FSM structure of driving maneuver planning.

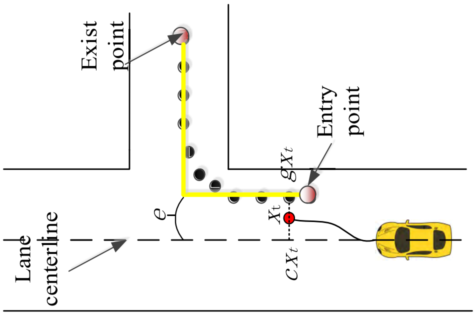


FIGURE 7. The gradient following strategy.

The map of general path and special path comes from GIS database and special path database respectively. The driving maneuvers under two types of paths are different. For instance, overtaking is not allowed in special path to ensure the safe driving for driverless bus. Therefore, the state of top-layer FSM is devised as {general path (R1) and special path(R2)}. Meanwhile, the type of road is distinguished by entry point and exit point, which are marked in global map (in Fig. 5). The trigger events as follows:

T1: The bus passes the entry point.

T2: The bus passes the exit point.

When the perception system fails to identify correct lanes, the planner automatically transits to GPS following mode. Aiming at such a problem, the middle-layer state set is {GPS Following(M1), Lane Keeping (M2), Gradient Following (M3)}. Gradient following is an intermediate mode which is mainly utilized to reduce the lateral error gradually, when the lane keeping mode switch to GPS tracking mode suddenly.

As is shown in Fig. 7, when driverless bus enters special path from general path, abscissa values of driverless bus is cx_t, and global GPS point is gx_t. So lateral error at time t is $e_t = |cx_t - gx_t|$. To reduce the error gradually, the abscissa of endpoint in planned trajectory is set as $x_t = e_t/k$ (k is scaling coefficient). When t increases, e_t decreases continuously, and the endpoint of generation trajectory approaches the global GPS point in map gradually (the method of trajectory generation is introduced in the section B).

The middle-layer state machine trigger conditions are divided into the following two kinds:

W1: Lane recognition is valid.

W2: Lane recognition is invalid.

In bottom-layer FSM, the driving state set is designed as {structured driving (Q1), unstructured driving (Q2), unstructured obstacle avoidance (Q3), following (Q4), overtaking (Q5), buffer adjustment (Q6) and special path (Q7)}. The individual states in the bottom-layer FSM correspond to the following external trigger events:

E1 / E2: The dynamic obstacle in front of the bus exits / disappears.

E3 / E4: The lane information from perception is valid / invalid.

E5: The bus enters buffered road.

E6/E7: The stationary obstacle in front of the bus exits / disappears.

E8: The bus enters the special path.

E9: The bus finishes the special path.

E10: The bus enters the area of destination.

According to the states and trigger events, bottom-layer FSM is used to switch between different states in Fig. 8. In our system, different driving states determine key reference points which is used for trajectory generation. Then, each state is introduced as follows:

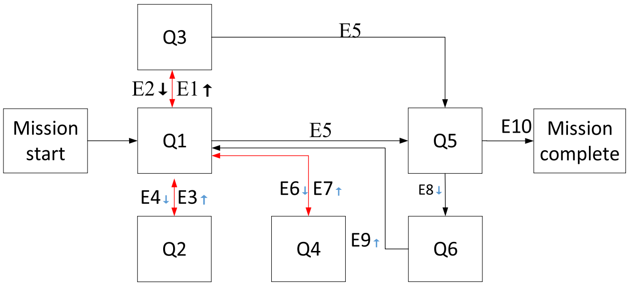


FIGURE 8. Bottom-level state transition.

(1) Structured driving: an initial state and lane recognition are valid. Lane recognition by perception layer is utilized to guide for driverless bus in this stage. So, the global optimal reference points are generated by lane detection.

(2) Following: In this state, the driverless bus follows the vehicle ahead only when the lane detection is valid. The system plans the speed according to vehicle ahead velocity and global optimal reference points based on lane detection.

(3) Overtaking: This stage allows driverless bus to overtake. It is divided into three substages: (a) lane change; (b) overtake; (c) lane return. Safe space is considered to avoid collisions with other vehicles. If $d_1 < d_{th}$, driverless bus changes the lane, here d represents the distance between two vehicles, d_{th} is the safe space. At this time, global optimal reference points are shifted from the current lane to another lane. Driverless bus returns to the lane again, when it overtakes the vehicle, and $d > d_{th}$. The process in detail is shown Fig. 9.

(4) Buffer adjustment: It is an intermediate stage that driverless bus needs to adjust speed especially during driving around a ramp or intersection. The signs of enter and exit are obtained from global map.

(5) Special path: This stage is used to deal with special terrain such as intersections and ramps. The global reference points come from the special path GPS database in this state. This status is activated until the entry point is accepted, and the bus switches to buffer adjustment status until the exit sign of special road is activated.

(6) Unstructured driving: Lane lines fail to be identified by perception in unstructured driving. Therefore, a Check Zone is set up manually, and obstacles in this area is detected as threaten obstacles in Fig. 10. Then check whether a dangerous obstacle in the Check Zone. If it exists, the system can

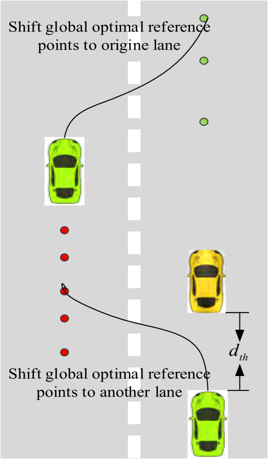

Shift global optimal reference points to origine lane
d_{th}
Shift global optimal reference points to another lane

FIGURE 9. The shift process of global optimal reference points under overtaking state.

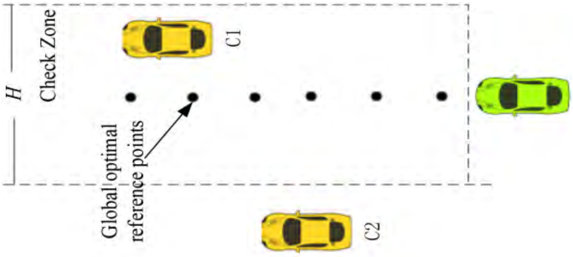


FIGURE 10. The Check Zone under the Unstructured driving, C1 is a threat for the driverless driving while C2 is not.

switch to the unstructured obstacle avoidance state, otherwise, the original state is maintained. Once the lane line is valid, the system switches back to structured driving.

(7) Unstructured obstacle avoidance: It is a combination of unstructured driving state and overtaking state. Driverless bus avoids collisions by setting Check Zone and shifting the global reference points between Check Zones. The state switches to unstructured driving if an obstacle disappears in the Check Zone, otherwise, the original state is maintained.

B. OPTIMAL TRAJECTORY GENERATION

The global optimal reference points are obtained based on global planning and driving maneuver planning under different scenarios. However, the raw reference points are usually sparse and unconstrained. They should be fitted to generate the optimal trajectory which is smooth and consistent with the bus kinetic constraints. In this section, cubic polynomial with constraints is applied to solve the problem.

From Fig. 11(a), cubic polynomial is generally written as

$$Y(X) = a_0 + a_1x + a_2x^2 + a_3x^3 \tag{1}$$

Under the established coordinate system in Fig. 11(b), initial position and initial heading of driverless bus are both

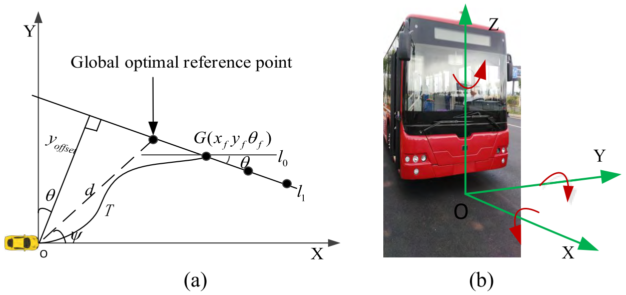


FIGURE 11. (a) shows the sketch of trajectory generation. *G* is a target point which is selected in global optimal reference points. *T* is the cubic polynomial curve, l_0 is parallel to the X axis at the point *G*. (b) displays the bus coordinate system.

zero, trajectory $Y(x)$ satisfies

$$Y(0) = 0 \tag{2}$$

$$dY/dx = 0 \tag{3}$$

combined with (1) -(3), constant coefficients are obtained

$$a_0 = a_1 = 0 \tag{4}$$

According to (4), the first derivative of cubic polynomial is expressed

$$2a_2x_f + 3a_3x_f = \tan\theta \tag{5}$$

Then, offset of the target point y_{offset} is denoted as

$$y_{offset} = y_f \cos\theta + x_f \sin\theta \tag{6}$$

Target point (x_f, y_f, θ) is also in the cubic trajectory, so it satisfies

$$Y_f = a_2X^2 + a_3X_f^3 \tag{7}$$

From (5)-(7),

$$a_2 = \frac{3y_{offset}}{X_f^2\cos\theta} - \frac{4\tan\theta}{X_f} \tag{8}$$

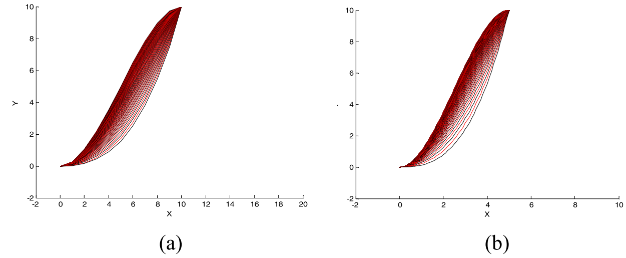
$$a_3 = \frac{3\tan\theta}{X_f^2} - \frac{2y_{offset}}{X_f^3\cos\theta} \tag{9}$$

Therefore, cubic polynomial is related to G, and different target points determine different trajectories. For instance, when x_f and y_f are fixed, but $\theta \in (0, 70°)$, trajectory clusters are shown in Fig. 12(a). Trajectory clusters of different target points are illustrated in Fig.12. An appropriate target point is essential for an optimal trajectory. Besides, several constraints are satisfied. The maximum steering angle of front wheel is limited for driverless bus.

According to geometry relation, turning radius R is estimated by (10)

$$R = d/\sin\delta_f \tag{10}$$

Where δ_f is front wheel angle, d means the wheelbase. The range of front wheel $Rg \in [-38°, 42°]$. From (10), the minimum turning radius $R_{\min} = d/\sin\delta_{f\,\max}$, here $\delta_{f\,\max}$ is the maximum front wheel angle. Meanwhile, the faster



(a) (b)

FIGURE 12. **Trajectory clusters under different target points. (a) (10, 10, 0 ∼ 70°). (b) (5, 10, 0 ∼ 80°).**

of driverless bus travels, the longer of trajectory is. In this paper, a piecewise turning radius is defined by the following equation:

$$L_a = \begin{cases} R_{\min} & v < v_{th} \\ R_{\min} + K_{la}v & v \geq v_{th} \end{cases} \tag{11}$$

where K_{la} is an adjusted coefficient, v represents current speed, v_{th} is a velocity threshold. Hence, turning radius of a feasible target point is larger than L_a. Furthermore, optimal trajectory generation algorithm is described as follows:

Algorithm 1 Optimal Trajectory Generation

```
**Input**: Global optimal reference points
**Output**: Trajectory equation $Y(x)$
**for** i = 1; $R_i \geq L_a$ do
1: Calculate $L_a$ according to the current speed.
2: Calculate the distance $d_i$ from coordinate point
   $(X_i, Y_i)$(i = 1, 2, ..., n, ):
       $$d_i = \sqrt{x_i^2 + y_i^2}.$$
3: Calculate the angel $\Psi_i$ between the line and x-axis:
       $$\Psi_i = \arctan[(y_i - y_0)/(x_i - x_0)].$$
4: Calculate the lateral offset of $(X_i, Y_i)$: $y_{offset}(i) = d_i$
   $\times \sin(\theta_i - \Psi_i)$.
5: Calculate minimum turning radius: $R_i = d_i^2/(2^*$
   $y_{offset}(i))$.
6: **end for**
7: Calculate $Y(x)$ according to $(X_i, Y_i, \theta_i$.
```

C. SPEED PLANNING

In this section, we imitate an experienced driver to solve speed planning problem for driverless bus. An expert system emulates decision-making ability of a human driver [44]. Expert rules are built according to driving maneuver, current speed v_r and reference speed v_d in TABLE 1.

Compared with structured driving, the speed under unstructured driving is slow for security. When driverless bus in the stage of overtaking, the first step is lane change that needs a slow speed. Then driverless bus overtakes ahead vehicle with a higher speed based on time-headway model, here v_f is the speed of ahead vehicle. Finally, driverless bus returns the original lane which is similar with the stage of lane changing. Driverless bus travels according to the front vehicle under the stage of following, which means v_d changes with v_f. It is noted that final desired speed is smaller than the maximum design speed 9 m/s in this system.

TABLE 1. **Expert rules for speed planning.**

stage \ v_r / v_d (m/s)	$0 \leq v_r < 2\ m/s$	$2 \leq v_r < 4\ m/s$	$4 \leq v_r < 6\ m/s$	$6 \leq v_r < 10\ m/s$
Structured driving	5.4 m/s	5.4 m/s	9 m/s	9 m/s
Unstructured driving	4.4 m/s	4.4 m/s	8 m/s	8 m/s
Buffer adjustment	2.7 m/s	2.7 m/s	4 m/s	4 m/s
Lane change/ lane return	2.7 m/s	2.7 m/s	4 m/s	-
Overtake	$1.5v_f$ +2 m/s	$1.5v_f$ +2 m/s	$1.5v_f$ +2 m/s	$1.5v_f$ +2 m/s
Following	v_f	v_f	v_f	v_f
Special path	2.7 m/s	2.7 m/s	4 m/s	-
Emergency stop or mission complete	0	0	0	0

V. TRAJECTORY TRACKING CONTROLLER

Bus trajectory tracking control system is a large delay and strong coupling system. Meanwhile, it has more stringent real-time requirements, so updating frequency of local path planning and controller need as soon as possible. Due to computational complexity of local planning is higher than control algorithm in our system, the position of bus requires real-time prediction and rolling optimization in an update period to ensure the control precision. Besides, bus trajectory tracking control requires stronger robustness in different scenarios.

A. COMPUTING PLATFORM

Bus trajectory tracking control demands a higher real time for large inertia system. So update rate of local path planning and trajectory tracking controller need as soon as possible. The runtime environment for two modules is in real-time Beckhoff Industrial Controlling Computer (ICU). Beckhoff created a global standard for automation with the launch of PC-based control technology. As for software, TwinCAT (The Windows Control and Automation Technology) automation suite forms the core of control system. TwinCAT software system turns almost each PC-based system into a real-time control with multiple PLC and robotics runtime systems. TwinCAT software system is utilized for path planning and trajectory tracking controller, it ensures update rate of two modules real-time, and it synchronizes strictly while the program is running.

B. ADAPTIVE ROLLING OPTIMIZATION WITH PREVIEW WINDOW

The trajectory tracking controller accepts local information in the form of a preview window periodically, here trajectory equation is $Y(x)$ and current speed is v_r. The size of preview

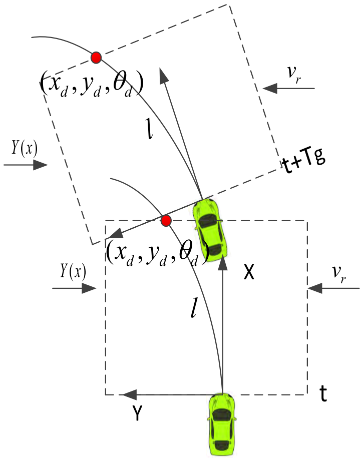


FIGURE 13. Adaptive rolling preview window.

window adjusts adaptively according to actual speed, and the rolling optimization period of preview window is T_g.

Fig. 13 shows the adaptive rolling preview window. Meanwhile, the size of window is defined (12).

$$l = v_r T_p + C \tag{12}$$

Where l is arc length of preview trajectory, v_r is current speed feedback, T_p is preview time and C is a threshold constant. According to $Y(x) = a_0 + a_1x + a_2x^2 + a_3x^3$, preview point $(x_d, y_d, \theta_d]$ on the trajectory is obtained by (13).

$$\begin{cases} Y' = 3a_3x^2 + 2a_2x + a_1 \\ l = \int_{x_0}^{x_d} \sqrt{1 + Y'^2} dx \\ \theta_d = \arctan Y' \end{cases} \tag{13}$$

Where Y' represents the derivative of trajectory, x_d denotes the abscissa value of preview point, θ_d is the desired heading angle at preview point, $x \in [x_0, x_f]$, x_0 is the origin of trajectory within rolling period. In Fig. 13, when the time is $t + T_g$ (next preview window), the window obtains new current speed and local trajectory based on current position, then a new preview point is calculated. With preview window updating, position error is eventually eliminated to heading error.

C. PREDICTIVE HEADING ERROR COMPENSATION MODEL

The front steer is easy to control quickly, and its lag is small in terms of driverless bus. Therefore, heading angle θ of front wheel is

$$\theta = \delta_f + \theta_h \tag{14}$$

Where θ_h is real-time heading of bus from combined navigation system, and δ_f is the front-wheel corner. Although the front-wheel are obtained directly, there exists a relation $\delta_f = f(\delta_s)$, δ_s is the steering wheel angle. By collecting steering wheel angle and corresponding front wheel angle, the least-square method is utilized to fit the functional relationship:

$$\delta_f = \delta_s / k \tag{15}$$

where k is the scale transmission coefficient. To reduce inertia and delay of bus, heading error is predicted in advance. The heading variation of driverless bus in adaptive preview window is related to the longitudinal speed v_r, front wheel angle δ_f. Based on Ackermann steering geometry, The heading variation $\Delta\varphi$ (rad) is approximately calculated as

$$\Delta\varphi = v_r T_g / R \tag{16}$$

where T_g is control period, R represents the radius of driverless bus movement around origin O. Submit (10) to (16), predictive heading compensation in a trajectory updating period is expressed as

$$\Delta\varphi = v_r T_g \sin\delta_f / d \tag{17}$$

where d represents the wheelbase and δ_f is the front-wheel corner. The predictive heading angle of front wheel is E.q.18.

$$\theta_r = \theta + \Delta\varphi \tag{18}$$

D. ADAPTIVE INCREMENTAL PID

According to the adaptive rolling preview window and the predictive error compensation model, the error of lateral control can be obtained as $e = \theta_d - \theta_r$. To improve the control accuracy and reduce the inertia, the incremental PID, which is widely used in engineering, is utilized for lateral control. The control increment Δu:

$$\begin{cases} \Delta u = K_p[e(k) - e(k-1)] + K_i e(k) \\ \qquad +K_d(e(k) - 2e(k) + e(k-2)) \\ e(-1) = 0 \\ e(-2) = 0 \end{cases} \tag{19}$$

where K_p, K_i, K_d denote the parameters of proportion, integral and differential respectively, k is sample number $(k = 0, 1, 2, \ldots\ldots)$, and $e(k)$ means heading errors at k step period. However, there is a coupling relationship between lateral and longitudinal control. The parameters of controller changes with different longitudinal velocities. Therefore, an adaptive relationship between the lateral control parameters and longitudinal velocity is expressed as:

$$\begin{cases} K_p = f_1(v_r) \\ K_i = f_2(v_r) \\ K_d = f_3(v_r) \end{cases} \tag{20}$$

Based on the actual experiments, the corresponding laws are summarized. The lateral error changes rapidly while the speed is high. K_p is appropriately reduced to weaken the system sensitivity. K_i is supposed to decrease appropriately due to the integral effect. K_d is reduced appropriately due to the differential part characterizes error rate. The relationship is defined as

$$K_p = \begin{cases} c/v_r, v_r > v_{set} \\ m, v_r \leq v_{set} \end{cases} \tag{21}$$

$$\begin{aligned} K_i &= ae^{-v_r} \\ K_d &= be^{-v_r} \end{aligned} \tag{22}$$

where a, b, c and m are the constant coefficients, v_{set} is a speed threshold. K_p is concretely expressed with a piecewise function based on current speed of driverless bus. Combined (19)-(22), the output of steering wheel control quantity is

$$u = \delta_s + \Delta u \tag{23}$$

To meet performance requirements of driverless bus, soft limit and hard limit are set for steering control. The hard limit is $u_{\min\_h} \leq u \leq u_{\max\_h}$, $u_{\min\_h}$ and $u_{\max\_h}$ are the upper and lower bound of steering wheel angle respectively; while the soft limit is $u_{\min\_s} \leq \Delta u \leq u_{\max\_s}$, $u_{\min\_s}$ and $u_{\max\_s}$ are the upper and lower bound of control increment respectively.

E. LONGITUDINAL CONTROL STRATEGY

The control signal output to the VCU actuators of driverless bus is target speed $v_{t\arg et}$ and deceleration a for the longitudinal control here. A protection module is set up which only allows one of the two signals to output at a time for protecting the execution layer. A trapezoidal acceleration and deceleration control method is designed to improve the stability and comfort.

(A) Initial stage:
If $v_r < v_{th}, v_d < (v_{th} + step)$, then, $v_{t\arg et} = v_d$.
If $v_r \geq v_{th}, v_d < (v_{th} + step)$, then, initial stage is over.
If $v_r < v_{th}, v_d \geq (v_{th} + step)$, then, $v_{t\arg et} = v_{th}$.
If $v_r \geq v_{th}, v_d \geq (v_{th} + step)$, $v_r < v_{t\arg et}$, then, $v_{t\arg et} = v_r + step$.
If $v_r \geq v_d$, then, initial stage is over.
(B) Acceleration stage:
If $v_d \leq (v_r + step)$, then, $v_{t\arg et} = v_d$.
If $v_d > (v_r + step)$, then, $v_{t\arg et} = v_r + step$.
(C) Deceleration stage:
Firstly, Calculate the deceleration: $a = (v_d - v_r)/t_{th}$.
If $a < a_{th}$, then, $a = a_{th}$.
If $a \geq a_{th}$, then, a remains unchanged.
(D) Emergency stop: $a = a_{\max}$.

Where v_r is the actual speed, v_{th} is the speed threshold, v_d means the result of local speed planning, and $v_{t\arg et}$ represents the expected speed to the execution layer, $(v_{th} + step)$ denotes the range of speed judgment. $t_{th}$is the customized deceleration time threshold; a_{th} is a customized comfortable deceleration threshold and $a_{\max}$ is the maximum response of deceleration for driverless bus.

Therefore, the entire lateral and longitudinal control structure is shown in Fig. 14. Combined the information of local path planning and driverless bus status, the lateral and longitudinal controllers update with adaptive rolling preview window to achieve rolling optimization and eliminate control error.

The algorithm of trajectory tracking runs in Beckhoff real-time controller, and its control period is 10 ms in our system. Desired speed v_d and desired tractor $Y(x)$ are generated by local planning. The method of adaptive rolling preview can calculate the desired heading angle θ_d at the preview point. Besides, the variation of heading in a rolling optimization period is predicted by (17). The steering wheel control quantity u is calculated through adaptive incremental PID. And trapezoidal acceleration and deceleration control method perform drive or brake for longitudinal control. The algorithm of trajectory tracking is described as follows:

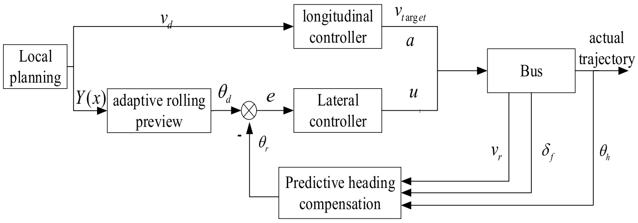


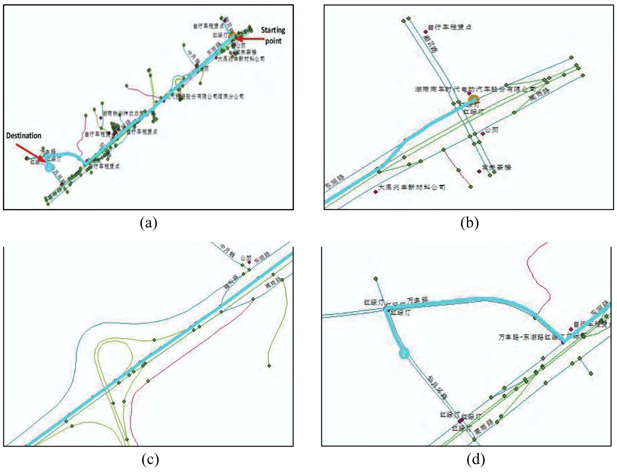
FIGURE 14. Lateral and Longitudinal control block.

VI. SIMULATION AND EXPERIMENTAL RESULTS

A. SIMULATION OF MAP-BUILDING AND GLOBAL PLANNING

In this section, a simulation is conducted to test the blocks of map-building and global planning. Firstly, the road net structure is built by importing a high precision electronic map which is a part of map in Zhuzhou of China into ArcGIS. The road network structure contains ramps and several lanes. Then several geographical attributes are set up, such as types of roads, whether a one-way road, length of road, etc. Furthermore, the network analysis tool in ArcGIS based on Dijkstra algorithm is used to plan a global path.

Fig.15(a) displays the whole planning path in the road network and Fig. 15(b), Fig. 15 (c) and (d) are the local enlarged plots which is clearly shown the path in different parts. From these results, planning path is the shortest path with the constant considerations of ramps, crossroads and one-way road.



(a)

(b)

(c)

(d)

FIGURE 15. Global path planning. Blue line represents the result of global planning. (a) The whole planning path. (b) Initail part. (c) Middle part. (d) End part.

B. SIMULATION OF OPTIMAL TRAJECTORY GENERATION

The driverless bus drives in a series of scenarios, including lane changing, lane keeping, turning and so on. Several

Algorithm 2 Trajectory Tracking Controller

```
Input: $Y(x)$, $v_r$,    Output: $u$, $v_{target}$ and $a$
Initialize $c, f, b, Tp, Tg, Tc, v_{set}$;
$i \leftarrow 1, k \leftarrow 0$;
While (not arrive destination)
    for $t < i^* T_s$ do
    // rolling preview window
        $l \leftarrow v_r T_p + C$;
        $y' \leftarrow 3a_3x^2 + 2a_2x + a_1$;
        $\theta_d \leftarrow \arctan Y'$;
        $l \leftarrow \int_{x_0}^{x_d} \sqrt{1 + Y'^2} dx$;
        for $k < T_g/T_c, t < l_g$ do
          // Predict heading error
              $\delta_f \leftarrow \delta_s/k$;
              $\Delta\varphi \leftarrow v_r T_c/R$;
              $R \leftarrow d/\sin\delta_f$;
              $\theta_r \leftarrow \theta + \Delta\varphi$;
              $\theta \leftarrow \delta_f + \theta_h$;
          // lateral control
              $e(k) \leftarrow \theta_d - \theta_r$
              $K_p \leftarrow f_1(v_r)$
              $K_i \leftarrow f_2(v)_r$
              $K_d \leftarrow f_3(v_r)$
              $\Delta u \leftarrow K_p[e(k) - e(k-1)]$
                  $+K_i e(k) - 2e(k) + e(k-2))$
              $u \leftarrow \delta_s + \Delta_u$
          // longitude  control
              if initial  stage
                then $v_{target} = f/(initial\ stage)$;
              if acceleration  stage
                then $v_{target}/f(acceleration\ stage)$;
              if  deceleration  stage
                then $a = f/(deceleration\ stage)$;
              if emergency  stop
                then  $a = a_{\max}$;
          // update control parainents
                $k \leftarrow k + 1$
                $e(k) \leftarrow e(k-1)$
                $e(k-l) \leftarrow e(k-2)$
        end for
    end for
    $i \leftarrow i + l$
    update $Y(x)$, $v_r$
  end while
```

typical driving scenarios are selected to simulate. The trajectory is generated by a cubic polynomial when an appropriate preview target point is determined by current state of driverless bus.

Fig.16 shows that local planning generates different trajectories according to the current direction of driverless bus in a state of lane changing. And the trajectory is displayed in global and local coordinate system respectively. When driverless bus moves forward, the red line is global path and the blue line represents the local trajectory which is shown in Fig. 16(a). Besides, Fig. 16(b) shows the result in the local coordinate system accordingly.

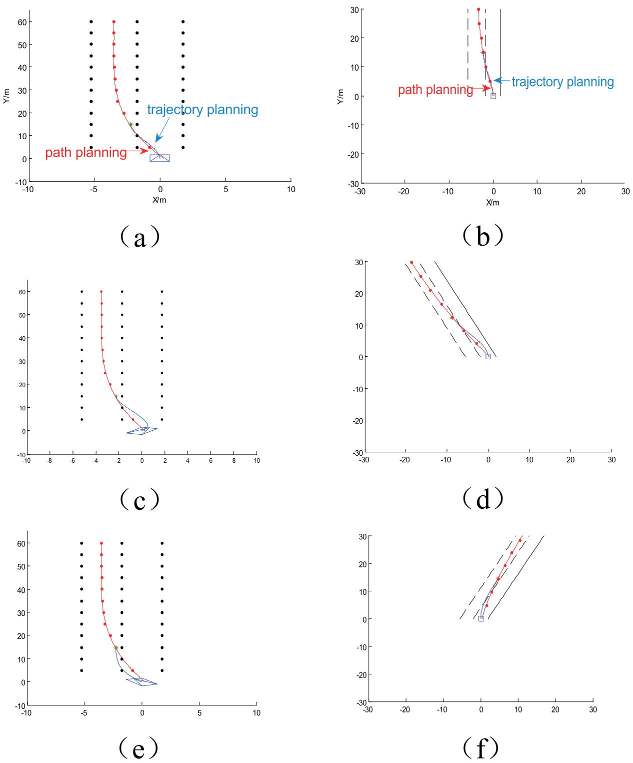


FIGURE 16. Blue lines represent the results of trajectory generation for lane change according to initial heading of driverless bus, which are showed in global coordinate system and local coordinate system respectively. Driverless bus initially towards forward ((a), (b)), right ((c), (d)) and left ((e), (f)) respectively.

Fig.16 shows that local planning generates different trajectories according to the current direction of driverless bus in a state of lane-changing. And the trajectory is displayed in global and local coordinate system respectively. When driverless bus moves forward, the red line is global Path, and the blue line represents the local trajectory which is showed in Fig. 16(a). Besides, Fig. 16(b) shows the result in the local coordinate system accordingly.

The initial direction of trajectory is consistent with the direction of driverless bus. When driverless bus towards the front right of the current lane and it exists heading error between driverless bus and global path, the generated trajectory is showed in global coordinate system in Fig.16(c) and local coordinate system in Fig. 16(d) respectively. Similarly, Fig. 16(e) and Fig. 16(f) show the situation of driverless bus towards the front right of current lane. The local trajectory and the direction of heading are tangent. Driverless bus to track the local trajectory smoothly and avoid a sharp turn although it exits large heading error.

Fig.17 and Fig.18 show the trajectory planning while driverless bus on the ramp and right turn respectively. The local path planning generates optimal trajectory based on the

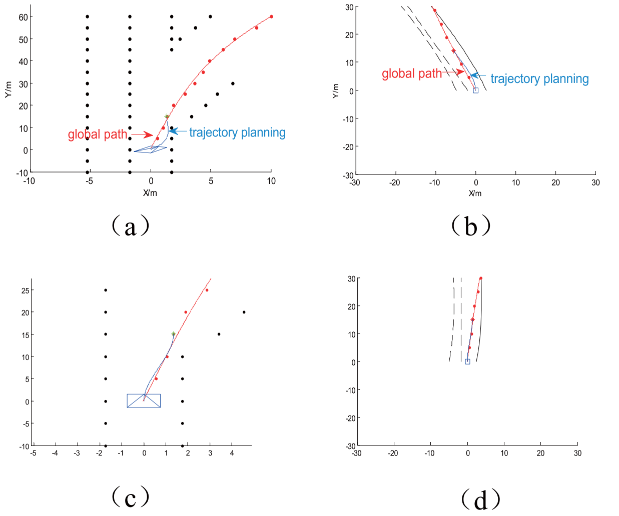


FIGURE 17. The results of trajectory generation for ramp. Driverless bus initially towards right ((a), (b)), forward ((c), (d)) respectively.

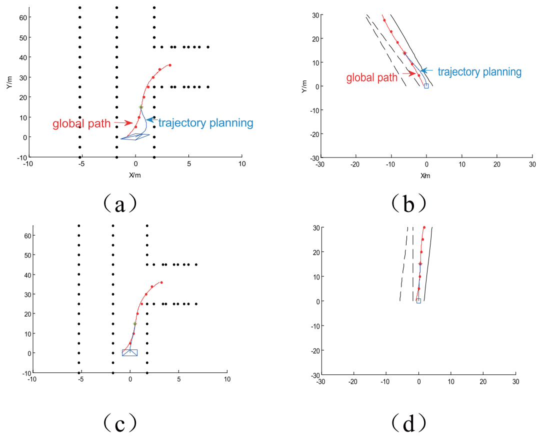


FIGURE 18. The results of trajectory generation for right turn. Driverless bus initially towards right ((a), (b)), forward ((c), (d)) respectively.

driving direction of driverless bus. The initial direction of trajectory is consistent with driving direction. Furthermore, the generated trajectory by cubic polynomial is smooth and meet driverless bus kinematics.

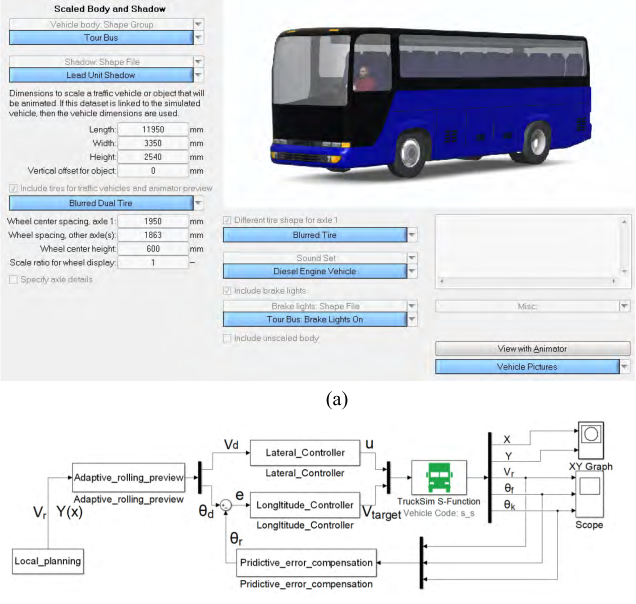
C. SIMULATION OF TRAJECTORY TRACKING CONTROLLER

In this section, driverless bus dynamic model is established by TruckSim. It delivers accurate, detailed, and efficient methods for simulating the performance of vehicles. It is universal tool for developing and testing vehicle controllers [45]. The real parameters of the electric bus are set in TruckSim in Fig.19(a). A more accurate dynamic model is established, which reflects the real characteristics of driverless bus. The real parameters of driverless bus are showed in Table 2.

The established bus model is extended by Simulink. Combined with trajectory tracking controller, the controller simulation platform is shown in Fig. 19(b). Different tracking trajectories are designed to further verify the performance of our controller. The effectiveness of controller is illustrated through simulation in an environment with various curvature profiles. Fig. 20(a) illustrates the tracking effects of proposed algorithm and PID. Trajectory of driverless bus in proposed algorithm is more consistent with the given trajectory compared with PID. Fig. 20(b) shows the error curve in travel process. Initial heading error is 0.2 rad, but the proposed algorithm is faster than PID algorithm in convergence. Because the heading of bus in next moment is predicted and the desired heading of given trajectory is not current position but the preview point. The proposed algorithm jointly considers heading information of forward trajectory and predicted heading of bus. To further test controller, an elliptic curve is given

TABLE 2. Vehicle parameters of driverless bus.

Symbol	Quantity	value
m_{cg}	vehicle quality / kg	17800
l	Length / mm	11950
w	Width / mm	2540
h	height / mm	3350
d	Wheelbase / m	5.9
a	Distance from center of mass to front axle / m	2.795
b	Center of mass to rear axle distance /m	3.105
Iz	vehicle yaw moment of inertia / kg.m^2	20000
C_F	Front wheel cornering stiffness coefficient N/rad	6500
C_R	Rear wheel cornering stiffness coefficient N/rad	5200
Rg	Range of the front wheel /°	-38° ~ 42°


Scaled Body and Shadow

Vehicle body: Shape Group: Tour Bus

Shadow: Shape File: Lead Unit Shadow

Dimensions to scale a traffic vehicle or object that will be animated. If this dataset is linked to the simulated vehicle, then the vehicle dimensions are used.

Length: 11950 mm
Width: 3350 mm
Height: 2540 mm
Vertical offset for object: 0 mm

☑ Include tires for traffic vehicles and animator preview

Blurred Dual Tire

Wheel center spacing, axle 1: 1950 mm
Wheel spacing, other axle(s): 1863 mm
Wheel center height: 600 mm
Scale ratio for wheel display: 1 –

□ Specify axle details

☑ Different tire shape for axle 1

Blurred Tire

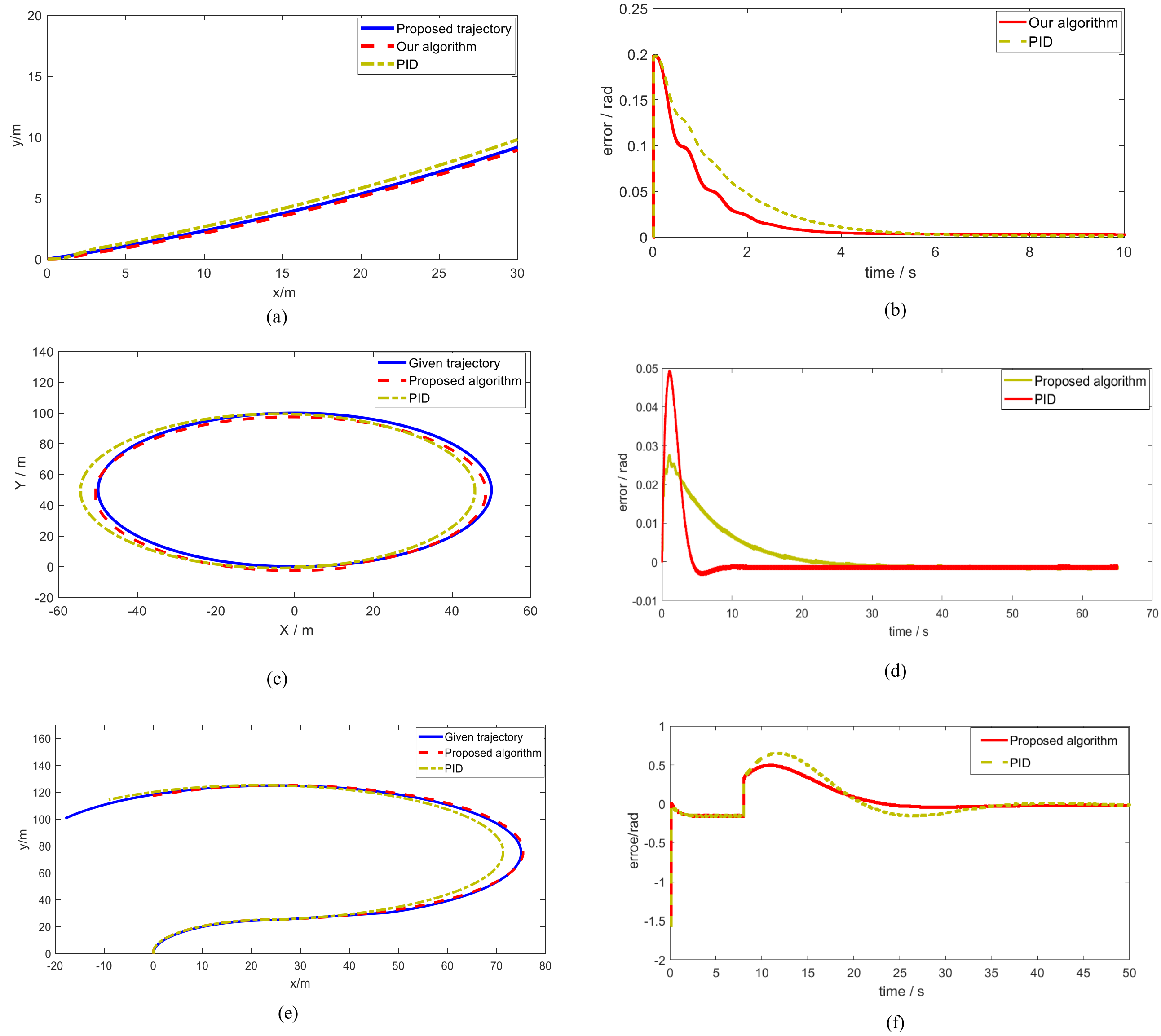
Sound Set: Diesel Engine Vehicle

☑ Include brake lights

Brake lights: Shape File: Tour Bus: Brake Lights On

□ Include unscaled body

FIGURE 19. Driverless bus dynamic model based on TruckSim and Simulink. (a) Parameters setting in TruckSim. (b) Control block diagram in Simulink.



(a) (b) (c) (d) (e) (f)

FIGURE 20. Different given trajectories are designed for Simulation of trajectory tracking controller. (a) and (b) show the effects of trajectory tracking and error curve for a given cubic polynomial curve respectively. Similarly, the results of elliptic curve and irregular curve are showed in (c)-(f).

in Fig. 20(c). Even though the heading error for PID algorithm in Fig. 20(d) converges to 0, this method fails to track the given trajectory in Fig. 20(d). This is because PID only utilizes the heading error in current position. It cannot track the given trajectory well when given curve curvature changes fast. More complex curve is given in Fig. 20(e)(f), the proposed algorithm is significantly better than PID algorithms.

D. FIELD EXPERIMENTS

Several field tests are conducted to evaluate the validity of path planning and navigation control system in real-world scenarios. Here, experimental field is an open environment, including various social vehicles passing in and out the filed frequently. Fig. 21 shows the results, and each picture is a snapshot from the driving videos, which are taken from inside, and outside of driverless bus.

Fig. 21(a)-(c) show the experiment in 8-shped road. Although the curvature of road is larger, driverless bus travels smoothly and tracks the trajectory precisely. Driverless bus drives itself from general path to special path based on the planning module and controller, and Fig. 21(d)-(f) show the process of turning. Besides, driverless bus switches from structured driving to overtaking state automatically when a vehicle drives slowly in front. Then, an optimal trajectory is generated for overtaking. The whole process is shown in Fig. 21(g)-(i). In Fig. 21(j)-(l), driverless bus activates emergency stop for safe, when pedestrians or vehicles enter the road suddenly. Fig. 21(m) shows the process of driverless bus merging into traffic. Driverless bus arrives at destination and stopped accurately in Fig.21(n). The driverless electric bus not only drives safely, but also completes the navigation missions perfectly. Meanwhile, driverless bus suc-

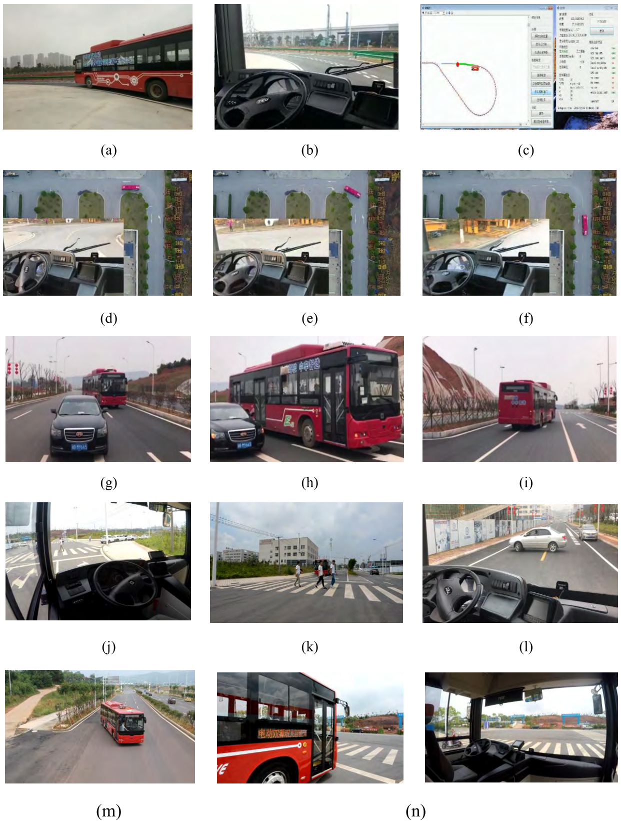


FIGURE 21. **Results of field tests under different scenarios. (a) Onside iamge. (b) Inside image. (c) Navigation map. (d) Turn start. (e) On road. (f) Turn end. (g) Lane change. (h) Overtake. (i) Lane return. (j) Inside image. (k) Outside image. (l) Outside image. (m) Initial stage. (n) Automatic parking from outside and inside.**

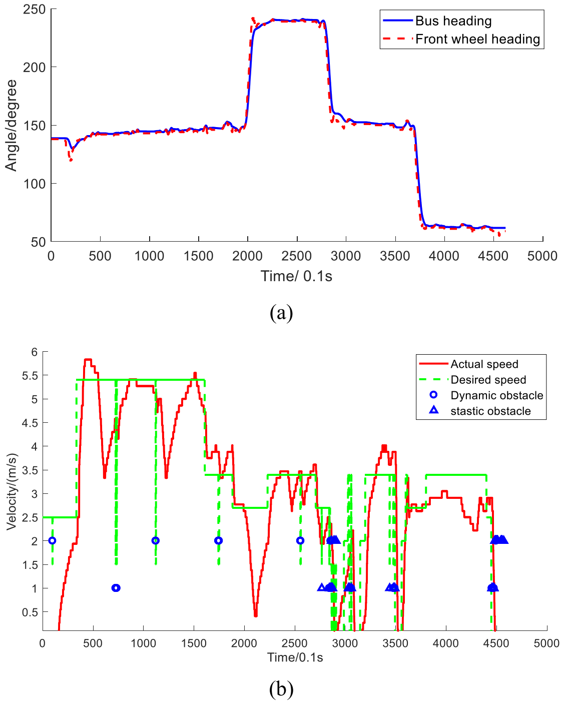


FIGURE 22. **Driverless bus heading and longitudinal control. (a) The front wheel heading and vehicle heading. (b) The result of longitudinal control.**

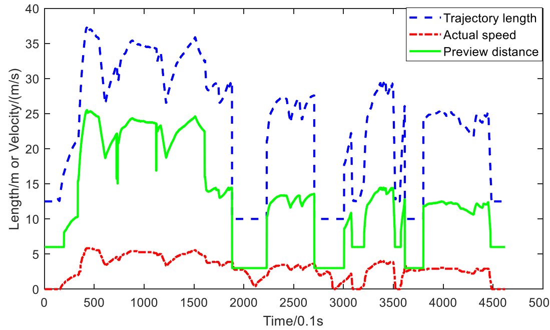


FIGURE 23. **The result of adaptive rolling preview.**

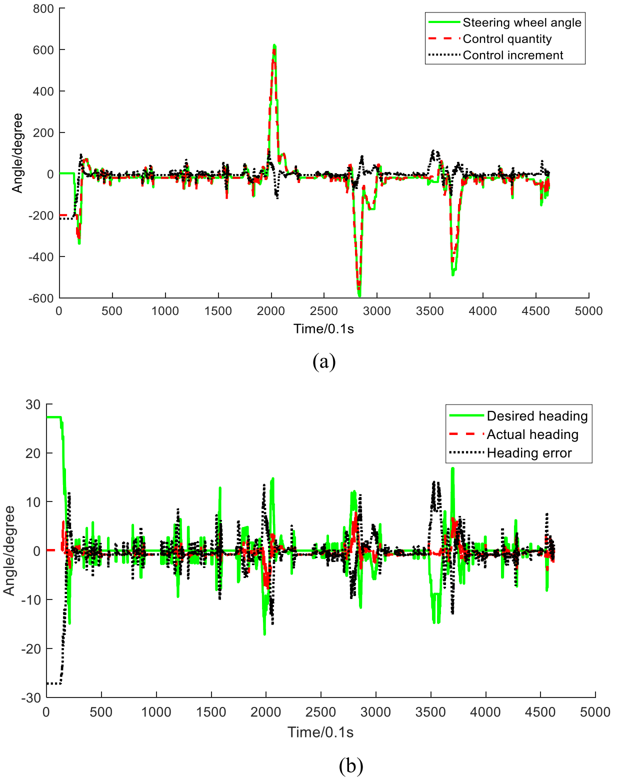
cessfully demonstrates merging intersection, overtaking and autonomous stop. Furthermore, the driving data are recorded during tests and detailed analysis.

The front wheel heading and vehicle heading are shown in Fig.22(a). the front wheel heading exceeds the vehicle heading. Therefore, the front wheel is suitable for control variable of trajectory tracking controller, which reduces the influence of time delay. In Fig. 22(b), desired speed is planned based on expert rules which is designed in local planning. The actual speed of driverless bus response the desired speed quickly. In addition, it varies trapezoidal mode which is devised in trajectory tracking controller no matter acceleration or deceleration. The blue circle represents a dynamic obstacle that forces driverless bus to slow down automatically. The blue triangle represents a static dangerous obstacle, for instance, pedestrian or vehicle breaks into the lane, which stop driverless bus slowly, or overtake it to avoid collision. No matter acceleration or deceleration, the controller is upgrading comfort and smoothness.

Trajectory tracking controller is devised based on the adaptive rolling window. In Fig. 23, trajectory length, actual speed and preview distance are collected during the test. The preview window length is adjusted adaptively by actual speed of driverless bus and local trajectory. Besides, desired tracking trajectory is updated within a preview window.

In Fig. 24(a), control quantity curves and steering wheel angle are well coincident. Besides, the change rate of actual heading and desired heading is almost synchronous during the test in Fig. 24(b). Thus, the lag time of lateral control is small. However, the values of desired heading and actual heading are different due to the amplitude limiting.

Fig. 25 illustrates the steering control with constraints which ensures driverless bus smoothly in straight line. It is significant that the constraints are added only when driverless bus travels in a straight lane. Meanwhile, driverless bus needs a large steering wheel angle to verify the controller effective and compare with almost straight line. Therefore, there are three peaks without constraints in Fig.25 since driverless bus turns. From above analysis, driverless bus travels smoothly in



(a)

(b)

FIGURE 24. The tracking results of steering wheel heading (a) shows the relationship of steering wheel angle and control quantity; (b) displays desired heading and actual heading of front wheel.

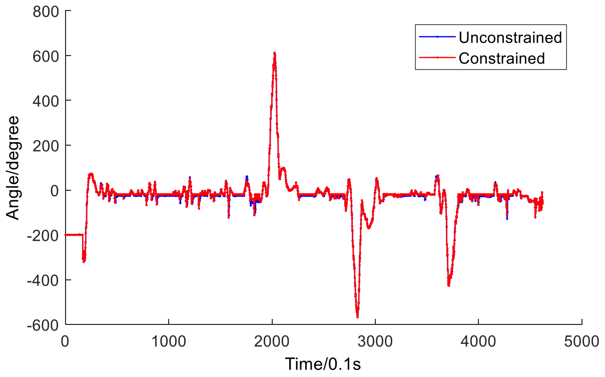
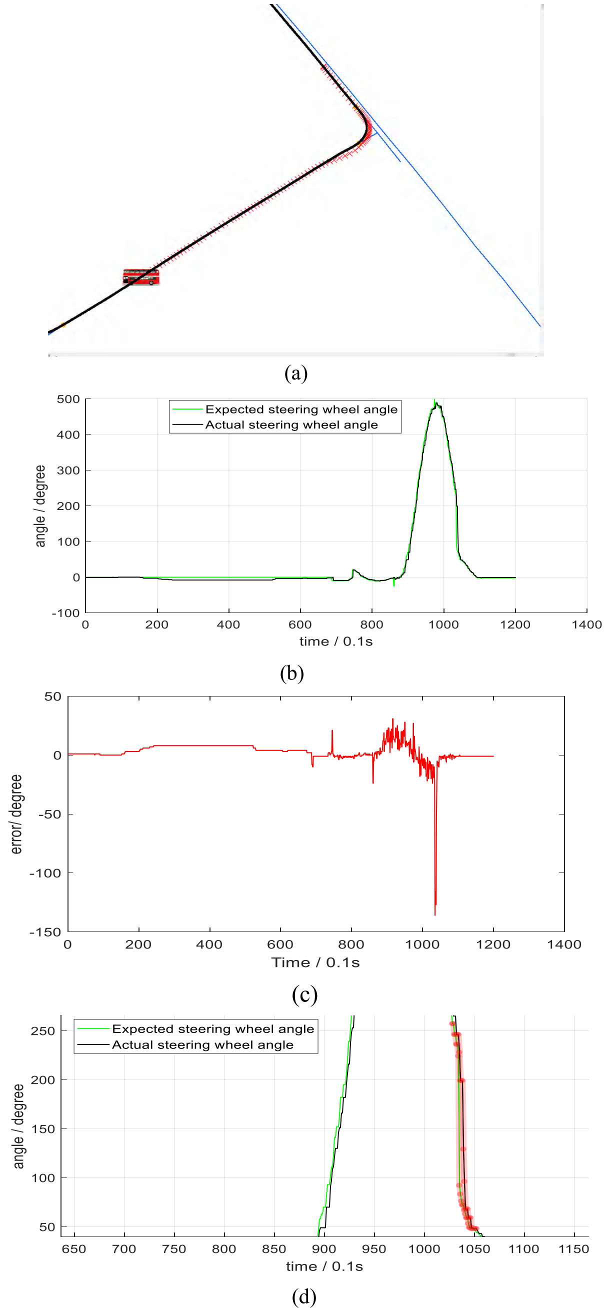


FIGURE 25. Steering control quantity with constraint.



(a)

(b)

(c)

(d)

FIGURE 26. Trajectory tracking analysis in the field test. (a) Actual trajectory of the driverless bus in global map. Blue line represents road, black line is global path, and red forks record the actual trajectory. (b) Expected steering wheel angle and actual steering wheel angle. (c) Error curve. (d) Partially enlarged view of (b).

straight lane with constraints and has a good performance in turning.

Test data for trajectory tracking in a typical road is analyzed in detail in Fig. 26. From Fig. 26(a), actual trajectory of the driverless bus and global path are recorded in global map. High tracking accuracy in straight path but it exits tracking error around the large turning which are less than 0.7m. Meanwhile, actual steering wheel angle is shown in Fig.26(b), which is consistent with expected steering wheel angle, and the peak in the curve of driverless bus represents its turning. The error angle in Fig. 26(c) is almost to zero when the bus drives in straight line. But the error angle looks larger while the bus is turning. In fact, the transmission ratio of the steering wheel to the front wheel is 23.1 for the bus. A special phenomenon is found in Fig. 26(c), there is a steep peak at about 100th second. We magnify the corresponding position in Fig. 26(d). The expected steering wheel reduces drastically, but the actual steering wheel angle cannot change quickly due to the lag. Meanwhile, the performance of tracking controller is not affected during the field test, because steering wheel rotation is gentle. However, many improvements are required to reduce lag and tracking error in the future, especially for turning tracking.

VII. CONCLUSIONS

This paper proposes a path planning and navigation control system for driverless bus. Before local path planning, map-building and global path planning are realized for driverless by ArcGIS. The simulations verify that global path planning meets the demand of good accuracy and rapidity. A triple-layer finite state machine is proposed to plan the driving maneuver, which reduces computational complexity. Considered the kinetics and safety of bus, cubic polynomial with constraints is utilized to generate optimal trajectory. Meanwhile, expert rules for speed planning are set up to imitate an experienced driver. For trajectory tracking controller, a lateral control based on adaptive rolling preview window is designed to solve the heavy inertia and great lag problems of driverless bus control system. Moreover, the heading error is predicted by a kinetics model. In addition, we devise the adaptive relationships between lateral control parameters and longitudinal speed through plenty of experiments to improve the robustness. Besides, a trapezoidal acceleration/deceleration controller is designed to improve the stability and comfort. Furthermore, simulations verify the global planning, local path planning and trajectory controller. Besides, filed tests are conducted to prove the practicability, and reliability of our system by driverless bus platform with 12 meters length.

In the future, we will focus on more accurate dynamic model of bus, as well as on the reliability of each module complicated dynamic and uncertain scenarios. Besides, many improvements are required to further reduce lag and tracking error especially when turning.

REFERENCES

[1] E. Fiorelli, N. E. Leonard, P. Bhatta, D. A. Paley, R. Bachmayer, and D. M. Fratantoni, "Multi-AUV control and adaptive sampling in Monterey Bay," *IEEE J. Ocean. Eng.*, vol. 31, no. 4, pp. 935–948, Oct. 2006.

[2] R. Marthiniussen, K. Vestgard, R. A. Klepaker, and N. Storkersen, "HUGIN-AUV concept and operational experiences to date," in *Proc. MTS/IEEE Techno-Ocean*, vol. 2, Nov. 2004, pp. 846–850.

[3] T. Litman, "Autonomous vehicle implementation predictions: Implications for transport planning," in *Proc. Transp. Res. Board Annu. Meeting*, Jan. 2014, pp. 36–42.

[4] L. Shi and P. Prevedouros, "Autonomous and connected cars: HCM estimates for freeways with various market penetration rates," *Transp. Res. Procedia*, vol. 15, pp. 389–402, Jun. 2016.

[5] S. Thrun *et al.*, "Stanley: The robot that won the DARPA grand challenge," *J. Field Robot.*, vol. 23, no. 9, pp. 661–692, 2007.

[6] C. Urmson *et al.*, "A robust approach to high-speed navigation for unrehearsed desert terrain," *J. Field Robot.*, vol. 23, no. 8, pp. 467–508, 2006.

[7] M. Montemerlo *et al.*,"Junior: The stanford entry in the urban challenge," *J. Field Robot.*, vol. 25, no. 9, pp. 569–597, 2008.

[8] C. Urmson *et al.*, "Autonomous driving in urban environments: Boss and the urban challenge," *J. Field Robot.*, vol. 25, no. 8, pp. 425–466, 2008.

[9] S. Glaser, B. Vanholme, S. Mammar, D. Gruyer, and L. Nouveliere, "Maneuver-based trajectory planning for highly autonomous vehicles on real road with traffic and driver interaction," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 3, pp. 589–606, Sep. 2010.

[10] U. Ozguner, C. Stiller, and K. Redmill, "Systems for safety and autonomous behavior in cars: The DARPA grand challenge experience," *Proc. IEEE*, vol. 95, no. 2, pp. 397–412, Feb. 2007.

[11] J. Hwang, D. Lee, K. Huh, H. Na, and H. Kang, "Development of a path planning system using mean shift algorithm for driver assistance," *Int. J. Automot. Technol.*, vol. 12, no. 1, pp. 119–124, 2011.

[12] R. Kala and K. Warwick, "Motion planning of autonomous vehicles in a non-autonomous vehicle environment without speed lanes," *J. Eng. Appl. Artif. Intell.*, vol. 26, no. 5, pp. 1588–1601, 2013.

[13] K. Zhou, L. Yu, Z. Long, and S. Mo, "Local path planning of driverless car navigation based on jump point search method under urban environment," *Future Internet*, vol. 9, no. 3, p. 51, 2017.

[14] Y.-B. Chen, G.-C. Luo, Y.-S. Mei, J.-Q. Yu, and X.-L. Su, "UAV path planning using artificial potential field method updated by optimal control theory," *Int. J. Syst. Sci.*, vol. 47, no. 6, pp. 1407–1420, 2016.

[15] T. Lozano-Pérez. and M. A. Wesley, "An algorithm for planning collision-free paths among polyhedral obstacles," *Commun. Acm*, vol. 22, no. 10, pp. 560–570, 1979.

[16] O. Takahashi and R. J. Schilling, "Motion planning in a plane using generalized Voronoi diagrams," *IEEE Trans. Robot. Autom.*, vol. 5, no. 2, pp. 143–150, Apr. 1989.

[17] N. M. Amato and Y. Wu, "A randomized roadmap method for path and manipulation planning," in *Proc. IEEE Int. Conf. Robot. Automat.*, vol. 1, pp. 113–120, Apr. 1996.

[18] C.-C. Tsai, H.-C. Huang, and C.-K. Chan, "Parallel elite genetic algorithm and its application to global path planning for autonomous robot navigation," *IEEE Trans. Ind. Electron.*, vol. 58, no. 10, pp. 4813–4821, Oct. 2011.

[19] T. Bucher *et al.*, "Image processing and behavior planning for intelligent vehicles," *IEEE Trans. Ind. Electron.*, vol. 50, no. 1, pp. 62–75, Feb. 2003.

[20] C. R. Baker and J. M. Dolan, "Street smarts for boss," *IEEE Robot. Autom. Mag.*, vol. 16, no. 1, pp. 78–87, Mar. 2009.

[21] A. Furda and L. Vlacic, "Enabling safe autonomous driving in real-world city traffic using multiple criteria decision making," *IEEE Trans. Intell. Transp. Syst.*, vol. 3, no. 1, pp. 4–17, Mar. 2011.

[22] J. Hardy and M. Campbell, "Contingency planning over probabilistic obstacle predictions for autonomous road vehicles," *IEEE Trans. Robot.*, vol. 29, no. 4, pp. 913–929, Aug. 2013.

[23] K. Jo, J. Kim, D. Kim, C. Jang, and M. Sunwoo, "Development of autonomous car—Part II: A case study on the implementation of an autonomous driving system based on distributed architecture," *IEEE Trans. Ind. Electron.*, vol. 62, no. 8, pp. 5119–5132, Aug. 2015.

[24] C. Xiu and H. Chen, "A behavior-based path planning for autonomous vehicle," in *Intelligent Robotics and Applications—ICIRA* (Lecture Notes in Computer Science), vol 6425, H. Liu, H. Ding, Z. Xiong, and X. Zhu, Eds. Berlin, Germany: Springer, 2010.

[25] P. Zhao *et al.*, "Dynamic motion planning for autonomous vehicle in unknown environments," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jul. 2011, pp. 284–289.

[26] M. Wang, T. Ganjineh, and R. Rojas, "Action annotated trajectory generation for autonomous maneuvers on structured road networks," in *Proc. 5th Int. Conf. Automat., Robot. Appl.*, Dec. 2011, pp. 67–72.

[27] Y. Cong, O. Sawodny, H. Chen, J. Zimmermann, and A. Lutz, "Motion planning for an autonomous vehicle driving on motorways by using flatness properties," in *Proc. IEEE Int. Conf. Control Appl.*, Sep. 2010, pp. 908–913.

[28] P. Papadakis, "Terrain traversability analysis methods for unmanned ground vehicles: A survey," *Eng. Appl. Artif. Intell.*, vol. 26, no. 4, pp. 1373–1385, 2013.

[29] J. Liu, P. Jayakumar, J. L. Stein, and T. Ersal, "A multi-stage optimization formulation for MPC-based obstacle avoidance in autonomous vehicles using a LIDAR sensor," in *Proc. ASME Dyn. Syst. Control Conf.*, 2014, p. V002T30A006.

[30] G. Zhang, C. A. Duncan, J. Kanno, and R. R. Selmic, "Unmanned ground vehicle navigation in coordinate-free and localization-free wireless sensor and actuator networks," *J. Intell. Robotic Syst.*, vol. 74, nos. 3–4, pp. 869–891, 2014.

[31] P. Velaskar, A. Vargas-Clara, O. Jameel, and S. Redkar, "Guided navigation control of an unmanned ground vehicle using global positioning systems and inertial navigation systems," *Int. J. Elect. Comput. Eng.*, vol. 4, no. 3, pp. 329–342, 2014.

[32] I. Cervantes and J. Alvarez-Ramirez, "On the PID tracking control of robot manipulators," *Syst. Control Lett.*, vol. 42, no. 1, pp. 37–46, 2001.

[33] M. D. Martino, F. Farroni, N. Pasquino, A. Sakhnevych, and F. Timpone, "Real-time estimation of the vehicle sideslip angle through regression based on principal component analysis and neural networks," in *Proc. IEEE Int. Syst. Eng. Symp.*, Oct. 2017, pp. 1–6.

[34] D. Q. Mayne *et al.*, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[35] H. Li and Y. Shi, "Robust receding horizon control for networked and distributed nonlinear systems," in *Studies in Systems Decision & Control*. Cham, Switzerland: Springer, 2017.

[36] R. Findeisen and F. Allgower, "An introduction to nonlinear model predictive control," in *Proc. 21st Benelux Meeting Syst. Control*, Veldhoven, The Netherlands, vol. 11, 2002, pp. 119–141.

[37] H. AbouaÏssa, M. Fliess, and C. Join, "On ramp metering: Towards a better understanding of ALINEA via model-free control," *Int. J. Control*, vol. 90, no. 5, pp. 1018–1026, 2017.

[38] L. Menhour, B. D'Andréa-Novel, M. Fliess, D. Gruyer, and H. Mounier, "An efficient model-free setting for longitudinal and lateral vehicle control: Validation through the interconnected pro-SiVIC/RTMaps prototyping platform," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 2, pp. 461–475, Feb. 2018.

[39] J. Han, D. Kim, M. Lee, and M. Sunwoo, "Enhanced road boundary and obstacle detection using a downward-looking LIDAR sensor," *IEEE Trans. Veh. Technol.*, vol. 61, no. 3, pp. 971–985, Mar. 2012.

[40] Q. Li, L. Chen, M. Li, S.-L. Shaw, and A. Nuchter, "A sensor-fusion drivable-region and lane-detection system for autonomous vehicle navigation in challenging road scenarios," *IEEE Trans. Veh. Technol.*, vol. 63, no. 2, pp. 540–555, Feb. 2014.

[41] V. Milanes, J. Villagra, J. Perez, and C. Gonzalez, "Low-speed longitudinal controllers for mass-produced cars: A comparative study," *IEEE Trans. Ind. Electron.*, vol. 59, no. 1, pp. 620–628, Jan. 2012.

[42] M. Neteler, M. H. Bowman, M. Landa, and M. Metz, "GRASS GIS: A multi-purpose open source GIS," *Environ. Model. Softw.*, vol. 31, pp. 124–130, May 2012.

[43] T. J. Misa, "An interview with Edsger W. Dijkstra," *Commun. ACM*, vol. 53, no. 8, pp. 41–47, 2010.

[44] W. Myers, "Introduction to expert systems," *IEEE Expert*, vol. 1, no. 1, pp. 100–109, Apr. 1986.

[45] R. Grace, A. Guzman, J. Staszewski, B. A. Peters, M. Mallis, and D. F. Dinges, "The carnegie mellon TruckSim: A tool to improve driving safety," in *Proc. 17th DASC. AIAA/IEEE/SAE. Digit. Avionics Syst. Conf.*, vol. 2, Nov. 1998, pp. I35/1–I35/8.

LINGLI YU received the Ph.D. degree in control science and engineering from the School of Information Science and Engineering, Central South University of China. She was a Visiting Scholar with Michigan State University from 2015 to 2016.

She is currently an Associate Professor with the Central South University of China. Her research interests focus on robot navigation control and path planning.

DECHENG KONG received the B.Eng. degree in automation engineering from the Huazhong Agricultural University of China, Wuhan, China, in 2016. He is currently pursuing the M.S. degree in control science and engineering with the School of Information Science and Engineering, Central South University of China. His interests focus on evolutionary computation, robot decision-making, and path planning.

XUANYA SHAO received the B.Eng. degree in measurement and control technology and instrument from the Central South University of China, Changsha, China, in 2017, where he is currently pursuing the M.S. degree in control science and engineering with the School of Information Science and Engineering. His interests focus on reinforcement learning, robot decision-making, and path planning.

XIAOXIN YAN received the B.Eng. degree in automation from the Chongqing University of Posts and Telecommunications, Chongqing, China, in 2017. He is currently pursuing the M.S. degree in control science and engineering with the School of Information Science and Engineering, Central South University of China. His interests focus on path planning and path tracking.