A Novel Parking Control Algorithm for a Car-like Mobile Robot

Yaru Zhang, Shuming Tang, Zhilin Zhang
State Key Laboratory of Management and Control for Complex Systems, Beijing Engineering Research Center of Intelligent Systems and Technology
Institute of Automation, Chinese Academy of Sciences
Beijing, China
{yaru.zhang, shuming.tang ,zhilin.zhang}@ia.ac.cn

Abstract—**This paper proposes a novel algorithm for parking motion of a Car-like mobile robot. The algorithm presented here addresses calculating equations for planning a parking path in real time. Moreover, by incorporating the constraints of the mechanical and kinematical characteristics of the car and the geometry of the parking lot in the path planning, we can turn a parking problem into solving algebraic equations. By tracking a planned path, the Car-like mobile robot can drive into the parking area without hitting any boundaries. The efficiency of the proposed algorithm is demonstrated by simulation.**

Keywords-parking control algorithm; path planning; vertical parking; CLMR

I. INTRODUCTION

WITH the development of automobile industry, the number of vehicles is increasing dramatically while driving space available in a city is decreasing drastically. Almost all drivers feel that driving in a city is more difficult and they have to pay more attention than ever before. Meanwhile, traffic accidents occurred frequently in recent years, especially during rush hours every day. Many automobile manufacturers have applied assistant driving system into vehicles to make driving safe and smooth. Furthermore, the increasing number of vehicles also results in the requirement of large amount of parking lots. While the land use for parking spots is limited in a city, the parking lots cannot meet the requirements. It seems that parking is difficult for a lot of drivers especially for new drivers. A new one often lack experience to accurately estimate the space between his vehicle and other objects or parking boundaries while handling the steering, brake, accelerator at the same time. In recent years, parking is given extra attention by many researchers and institutes both in the research area of assistant parking systems and autonomous vehicles.

The parking lot style can be broken down into following three categories, including vertical parking lots, diagonal parking lots, and parallel parking lots as shown in Figures 1-3. Vertical parking and diagonal parking lots exist commonly in a parking area while parallel parking lots are usually seen on both sides of a street.

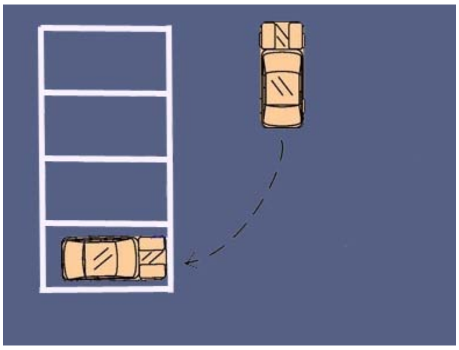


Figure 1. Vertical parking lots

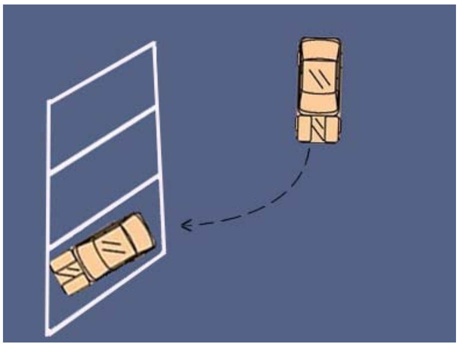


Figure 2. Diagonal parking lots

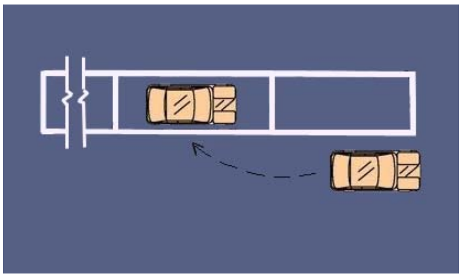


Figure 3. Parallel parking lots

Many parking control algorithms have been developed in recent years. There are mainly two types of parking algorithms: the first type is using a reference path for a car to track, and the other type is a fuzzy logic control method. In the reference-path-planning-based algorithms, a feasible geometric path is generally first designed considering constraints of parking environment and constraints of dynamic and kinematic characteristics of vehicles. Then adopting various control algorithms, the vehicle can track the designed path to a desired parking lot. The geometric paths are generated differently. For example, Literature [1] generated a fifth-order polynomial's curvature considering the slop and curvature constraints for the robotic vehicle to track. Literature [2] designed a parking trajectory with three steps, and used the multiplex double-closed loop method for the trajectory tracking control of a vehicle. In Literature [4], an arc parking curvature was generated for a vehicle to track, and the algorithm was theoretically proved and successfully applied to the OSU-ACT in the DARPA Urban Challenge 2007. There were other curves, such as cubic spiral [3], Bezier curves [12], etc. Most of the aforementioned geometric paths are winding and not easy to follow, thus the path following error may occur, which makes the vehicle fail to park into a parking lot. Therefore, many researchers focus on tracking and posture stabilization to minimize the path following error and solve the problem.

In the fuzzy logic control method, the human driving experience is utilized to generate a knowledge database in the form of IF-THEN rules which needs to be pre-trained well, and a fuzzy logic controller is designed to control a vehicle to a desired position. Some researchers think that the fuzzy logic control method is more intelligent because it is more like human driving a car. When one drives a car, based on his daily experience, he doesn't contemplate a path at first rather than adjusting the steering angle and speed in real time. In this method, a fuzzy system was usually used to select the most suitable behavior from the knowledge database according to its internal states and objectives [5]. In Literature [6], the authors used fuzzy logic controller to maneuver the steering angle of the car-like mobile robot to a parking lot collision-free.

In this paper, we mainly concentrate on vertical parking problems. A novel parking control algorithm is proposed. We use a circular arc to design a reference path which is easy for vehicle to follow and also computationally efficient. In the simulation section of this paper, this algorithm is efficient with different initial positions. Actually, we are going to make it implemented in a Car-like mobile robot in our future work. Therefore, we have considered this algorithm some restrictions that a Car-like mobile robot may have in this paper such as mechanical restrictions, sensor type restrictions, and etc.

This paper is organized as follows. In Section II, the details of the proposed parking control algorithm will be addressed. In Section III, some computer simulations are conducted and corresponding results are demonstrated the validity of the proposed arc path parking algorithm. Section IV concludes this paper.

II. PARKING CONTROL ALGORITHM

A. Car-like Mobile Robot

As we pointed out above, our proposed parking control algorithm is related to a Car-like mobile robot (CLMR). Therefore, we give a brief introduction to this CLMR here. The architecture of a CLMR has adopted an EMB3850 IPC and an Arduino sensor interface board to perform signal processing and transferring. The Arduino sensor interface communicated with the EMB3850 via a USB cable, and the electronic speed controller (ESC) and servo were connected with the Arduino sensor interface board. There were four cameras installed on a CLMR. All of the cameras are connected to the EMB3850 to process the image information. Two of them were responsible for detecting the parking lot and sent images to the EMB3850 to analyze and compute the coordinates of the vertices of the parking lot. One was installed lower on the CLMR, and the pitch angle was flat in order to detect the parking lot from far away. The position of the other one mounted higher looked down the ground to enable the CLMR seeing the parking lot while approaching it. There were two batteries to provide power to the CLMR. One was an 8V lithium battery providing power for the ESC, and the other was a 12V NiMH battery providing power for the EMB3850, Arduino and cameras. The overall appearance of a CLMR is shown in Figure 4. In the parking control algorithm, we assume that the parking lot has been detected already and mainly focus on the discussion of a parking control algorithm.

Figure 4. The appearance of a CLMR

B. Kinematic Model

Since the speed of a car is commonly very slow during a parking process, we assume the tire slip angle is zero and the vehicle yaw angle equals to the velocity direction of the rear axle. The Ackerman steering vehicle kinematic model [11] shown in Figure 5 was adopted in this paper.

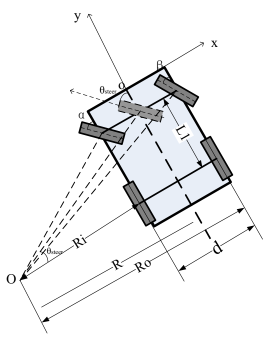


Figure 5. The Ackerman steering vehicle kinematic model

In Figure 5, R is the turning radius of the vehicle; Ri and Ro are the radiuses of the minimum and maximum circles when the turning radius is R, respectively; L1 is the distance from front axle to rear axle; d is the lateral wheel separation; α and β are the left and right steering angle. The origin of local coordinate is chosen to be the projection of the center of the front bumper of the vehicle. We can derive Equations (1) - (5) according to Figure 5:

$$\cot\beta - \cot\alpha = \frac{d}{L1}. \tag{1}$$

$$\cot\theta steer = \cot\alpha + \frac{d}{2*L1}. \tag{2}$$

$$\cot\theta steer = \cot\beta - \frac{d}{2*L1}. \tag{3}$$

$$\cot\theta steer = \frac{\cot\alpha + \cot\beta}{2}. \tag{4}$$

$$R = \frac{L1}{\tan\ \theta steer}. \tag{5}$$

In the equations above, θ_{steer} is the steering angle of the vehicle.

C. Calculate the Final Point

We choose the final point at the center of the entrance boundary of the parking lot. Its coordinate is calculated by the coordinates of the vertices of the parking lot. Compared to Literature [4], the advantage of choosing such final point is that we can ignore some geometry constraints of parking lot and, also the parking control algorithm is simple and easy to be calculated. After the position of the final point was chosen, the parking control algorithm will plan a final circular arc connecting an initial point and the final point. A CLMR controls its steering angle to adjust its position from time to time in order to follow the planned arc which can pull into the parking lot without hitting boundaries. When the CLMR arrives at the final point, the orientation of the CLMR overlaps with the longitudinal orientation of the parking lot. Thus after reaching the final point, the CLMR just pull straightforward into the parking lot and stop in the middle of it.

As we mentioned above, we choose a local coordinate at the projection of the center of the front bumper to simplify the path planning procedure, unlike others chose the local coordinate in the parking lot [4] or some other places [8]. In this case, the Y axis is the heading direction of the CLMR. Thus, the local coordinate is moving along with the CLMR and the coordinates of the final point is a dynamic point. That is to say the final point appears at different position in each coming picture frame according to camera detection, and its coordinates need to be recalculated according to the current origin of local coordinate. The final point can be calculated by Equations (6) and (7), illustrated in Figure 6.

$$X_F = \frac{X_A + X_B}{2}. \tag{6}$$

$$Y_F = \frac{Y_A + Y_B}{2}. \tag{7}$$

where (X_F, Y_F) is the coordinate of the final point.

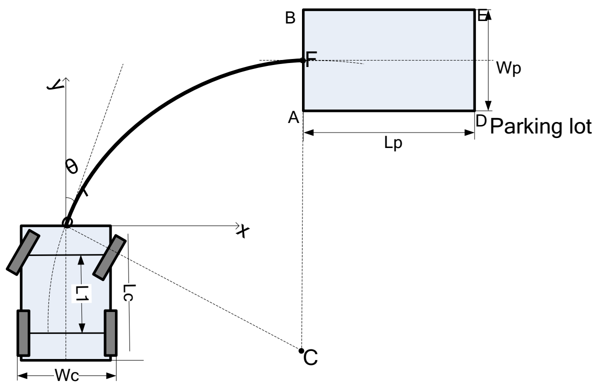


Figure 6. Path planning

In Figure 6, points A, B, D, E are the vertices of the parking lot, and point C is the center point of a desired circular arc. Lc and Wc are the length and width of the CLMR, respectively. Lp and Wp are the length and width of the parking lot, respectively. The point F is the calculated final point.

D. Parking Control Algorithm

In order to park without hitting any boundaries, the following parking constraints were considered.

1）The parking lot is a destined area with specific width and length.

2) The CLMR is 1/10 of a general real car, so the parking lot is also 1/10 of a real one.

3) The turning radius of CLMR is bigger than its minimum turning radius.

Considering the above three items, let's discuss our parking control algorithm. In this algorithm, we just control the steering angle of a CLMR, because its speed is very low. For the sake of simplicity, we assume that the speed of a CLMR only has three values in this paper, which can be represented as an enumeration set {driving forward, stop, driving backward}. In driving forward, we set the speed at a positive constant small value. In driving backward, we set the speed at a negative constant small value. The stop speed is set to zero. Then, the parking algorithm is described as follow.

1) First, the CLMR drives forward while calculating a desired path repeatedly in order to find a final path. A desired path is a circular arc which connecting the origin of the local coordinate and the final point. At the same time, the radius of the circular arc is greater than the minimum turning radius of the CLMR and the tangent of the circular arc at the final point is to the centerline of the parking lot. When no desired path is found, the CLMR will pull backward until one is available. When the car is pulling backward, the steering angle turns to the opposite direction with respect to the previous steering angle. For a final path, it has to meet the follow two requirements: 1) it is a desired path first; and 2) its tangent at the origin point is Y axis.

2) Since the origin of the local coordinate is the projection of the center of the front bumper of the CLMR and the Y axis

is the heading direction of the CLMR, the heading angle is always 90°with respect to X axis. Then we can calculate the tangent angle (θ_{ta}) of a desired path at the origin of the local coordinate by Equation (8). We use θ_r to specify the difference between the heading angle and the tangent angle. In the first time to calculate the desired path, it is generally not the final path because it is most possible to have a difference between the heading angle and the tangent angle. In order to derive the final path, the CLMR moves by turning right or left with the maximum steering angle and pulling forward or backward and tries to minimize θ_r until the heading angle and the tangent angle are equal. Actually, when absolute value of θ_r is less than a predetermined threshold, we reckon that the heading angle and the tangent angle are equal.

$$\theta_{ta} = \tan^{-1}(-\frac{x_c}{y_c}). \quad (8)$$

where θ_{ta} is the tangent angle at the origin of the local coordinate, (x_c, y_c) is the coordinate of the center of the circular arc shown in Figure 6.

3) When the heading angle and the tangent angle are equal, the tangent at the origin point is to Y axis, the center of the final reference path is on the X axle, and its radius is the horizontal coordinates, as illustrated in Figure 7. Thus, the CLMR can follow the final path into the parking lot via controlling its steering angle calculated by Equation (5). In the engineering application, because it is impossible to accurately control the steering angle, the CLMR may hit boundaries. At such circumstances, the CLMR can pull backward until a new desired path is available and then pulls forward using the algorithm described in step 2 until a new final path is derived.

4) When the front wheel is parallel to the CLMR body, the CLMR will stop in the parking lot and the parking process is over.

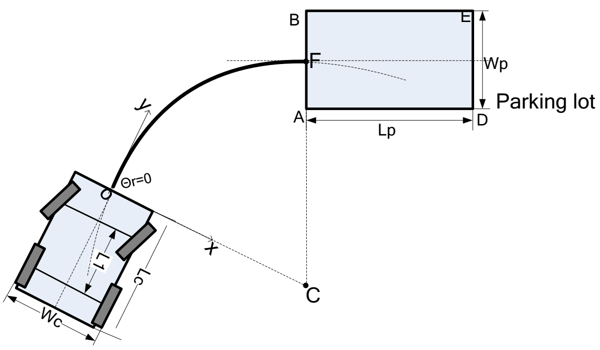


Figure 7. Θr=0 the final path is found

The detailed flowchart of the parking algorithm is shown in Figure 8.

III. Simulations

In this section, some computer simulations are conducted and corresponding results are demonstrated the validity of the proposed arc path parking algorithm.

The simulations were conducted at Visual Studio 2008 platform in C programming language. Figure 9 shows the simulation results of the parking control algorithm at different initial positions. As we can see, our algorithm can always find a final path to execute parking successfully. In Fig. 9(a), the parking trajectory was smooth. By adjusting the steering angle, the car found some desired paths and the final path was acquired quickly. Fortunately the CLMR followed the final path into the parking lot without hitting any boundary in driving forward. In Fig. 9(b), the car approached the parking lot from another direction, and the parking trajectory was also smooth. The reason of this simulation result is similar with that in Fig. 9(a). In Fig. 9(c), the parking trajectory was winding. The car found a desired path and pull forward. When it was about to hit the boundary, the car pull backward until finding another desired path. Then it pull forward again and drove into the parking lot. The simulation result showed that the parking trajectory of the CLMR depended on the initial position and heading direction of the car.

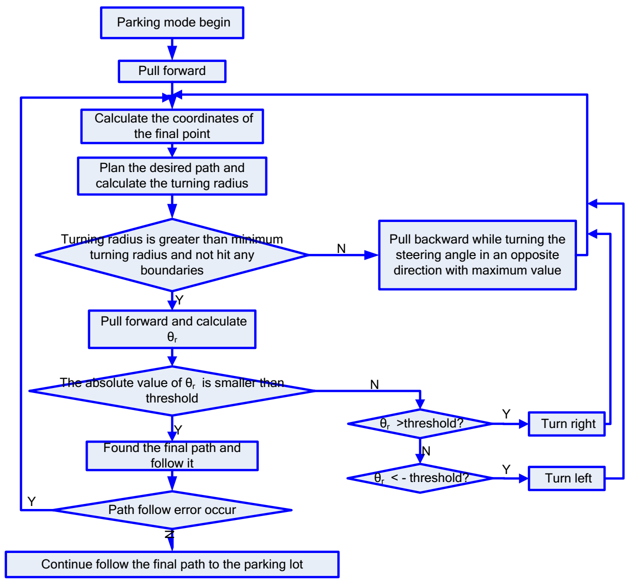
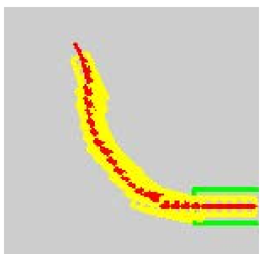
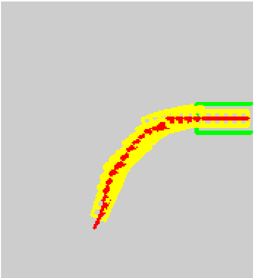


Figure 8. Flowchart of parking algorithm



(a) Parking from position 1



(b) Parking from position 2

(c) Parking from position 3

Figure 9. Parking algorithm simulation

IV. CONCLUSION

We proposed a novel parking control algorithm. By incorporating the constraints of the mechanical and kinematical characteristics of the car and the geometry of the parking lot in the path planning, we can turn a parking problem into solving algebraic equations. The advantage of the proposed algorithm is practical and computationally efficient.

The method of choosing local coordinate simplifies equations for calculating a reference path. By tracking a final path, the CLMR can drive into the parking area without hitting any boundaries. The simulation results also showed that this algorithm is able to execute parking at different initial positions.

In our future work, on one hand, we will apply this algorithm to a real Car-like mobile robot. On the other hand, we will research how to detect the coordinates of a parking lot via sensors on a CLMR.

ACKNOWLEDGMENT

This work was supported in part by National Natural Science Foundation of China under Grant 60974095, Grant 60921061, and Grant 90920305.

REFERENCES

[1] Douglas Lyon, Parallel parking with curvature and nonholonomic constraints, Intelligent Vehicles Symposium, 1992, 341-346.

[2] Chen Jiajia, Mei Tao, Zhu Hui, Study of autonomous vehicle parking system based on path planning and the double-closed loop method, Electronic measurement technology, 2011, vol 34 no 3 pp20-24.

[3] Kanayama, Y., Hartman, B., Smooth local path planning for autonomous vehicles, TRCS8815, Department of Computer Science, University of California, Santa Barbara, CA.

[4] Ming Feng Hsieh, Umit Ozguner, A parking algorithm for an autonomous vehicle, IEEE Intelligent vehicle symposium, 2008.

[5] F.Gomez-Bravo, F.Cuesta, A.Ollero, Parallel and diagonal parking in nonholonomic autonomous vehicles, Engineering Applications of Artificial Intelligence, 2001, 14, pp 419-434.

[6] Tzuu-Hseng S.Li, Shih-Jie Chang, Autonomous Fuzzy Parking Control of a Car-like Mobile Robot, IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans, Vol. 33, No. 4, 2003, pp 451-465.

[7] Igor E. Paromtchik, Christian Laugier, Autonomous parallel parking of a nonholonomic vehicle, Proceedings of IEEE International Conference on Fuzzy Systems, 1996, 3117-3122.

[8] Daobin Wang, Huawei Liang, Nina Yang, Hang Gao, On path planning of two self-parking methods, Electronic Measurement Technology, 2011,vol. 34, no.1, pp27-38.

[9] Wei Cheng Wang, Rongshun Chen, A vision-based fuzzy logic controller for backing-up an autonomous vehicle, Journal of Intelligent and Fuzzy System, 2008 pp273-284.

[10] Young-Woo Seo, Chris Urmson, David Wettergreen, Jin-Woo Lee, Building lane-graphs for autonomous parking, IEEE/RSJ International Conference on Intelligent Robots and Systems, 2010.

[11] http://en.wikipedia.org/wiki/Ackermann_steering_geometry.

[12] Yu Liu, Yanli Ma, Tao Li, Parallel parking path generation based on Bezier curve fitting, Science and technology review, vol. 29, no. 11, 2011.