2012 Fifth International Symposium on Computational Intelligence and Design

A Path Planning Method for Assistant Parallel Car-Parking

Shuqiang Liu, Xiangjing An, Erke Shang, and Hangen He
College of Mechatronics and Automation
Natl. Univ. of Defense Technology
Changsha, P.R. China
Email: belongcoo@163.com, anxiangjing@gmail.com, erke1984@qq.com, hangenhe@yahoo.com

***Abstract*—Path planning is one of the key technologies in an intelligent car-parking system. This paper presents a path planning method for parallel parking mode. A scheme of a two-arc parking path is proposed, with the start position given but the goal to be yielded. The parking process is formulated as a constrained nonlinear optimization problem, where two objective functions are studied. And we gained the optimized results by MATLAB Optimization Toolbox.**

***Keywords*-path planning; constraints; nonlinear optimization;**

I. Introduction

Recently years, automatic or assistant car-parking has become an important function in new automobile technologies. It is a system to help drivers who do poor job of parking cars. Usually, the system comprises a rear view camera to provide the road condition behind the car concretely, and sometimes some other sensors are needed, such as sonar, ladar and radar. With the help of these sensors, the system gets to know exactly about the terrain around. Then comes up the problem of path planning, which will be discussed in the present paper.

Path planning aims to find a safest and easiest trajectory from the present position to the parking space. There have been many studies on this problem. [1], [2], [3] and [4] provide detailed overviews on existing path planners. [5] and [6] introduce a famous approach that uses chained form conversion with sinusoidal inputs and polynomial inputs respectively.

Reeds and Shepp [7] designed an algorithm to gain the shortest path between specified starting and ending points taken by a car given minimum turning radius. The paths they considered consist of only line segments and arcs of minimal turning radius, where arcs and segments fit smoothly and cusps are allowed.

Based on the above works, Fraichard and Scheuer [8] proposed Continuous Curvature steering method to plan paths in the absence of obstacles. Their paths are made up of line segments, circular arcs and clothoid arcs. And their algorithm served efficiently in general planning schemes.

Mller et al. [9] developed the two-step trajectory planning algorithm. They inherited and improved Fraichard's theory of Continuous Curvature planner, and carried it out in feedforward control for parking cars [10].

Laumond et al. [11] presented a fast and exact path planner using recursive subdivision of a collision-free path. The resultant trajectory is optimized to give a path of near-minimal length. And they proved the completeness and the complexity of the algorithm in their paper.

Kim et al. [12] used the slice projection technique to implement path planning in car-parking tasks. Their planning scheme proposes not a single solution but a candidate solution set, in order for optimizing with respect to performance criteria. A short time later, they improved their algorithm in [13] to adapt to narrow environment.

In this paper, we firstly propose an approach of parallel parking path, which mainly consists of two arcs. Then, we simulate the process of backing up and list the collision constraints. Two optimizing objects are considered and resolved respectively. The simulation results show that the optimized trajectories are feasible.

II. Two-arc Parking Path

Parallel parking is the only mode we discuss here. Figure 1 shows the two-arc parking path and the motion process. There is no cusp between the two arcs, i.e., the joint of the two segments is smooth. Theoretically, if the car keeps in reverse gear and only turns the steering wheel at the joint, the trajectory should be of this form. What's more, the blue and brown rectangles show inner situations of the motion of backing up.

In the real operation of parking, the process is split into five steps in detail, as shown in Figure 2. Step 1, reverse the car to an appropriate position where the path planning program begins; Step 2, turn the front wheels to some angle, meanwhile the system detects the environment and implements the planning algorithm, and then the program indicates the resultant trajectory and the goal position in order for the driver to decide whether to accept; Step 3, keeping up the front wheels angle, reverse the car to complete the first arc, synchronously the system measures the track, and denotes a stop signal opportunely; Step 4, turn the front wheels to the opposite direction to Step 2 to the maximum, in order for the minimum radius; Step 5, reverse until the car is in an appropriate position.

978-0-7695-4811-1/12 $26.00 © 2012 IEEE
DOI 10.1109/ISCID.2012.168

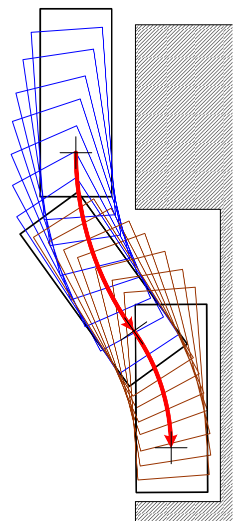


Figure 1. Two-arc path parking

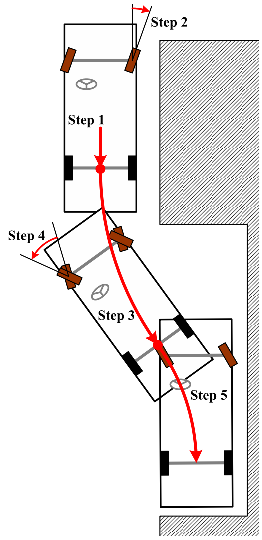


Figure 2. Five steps of parking

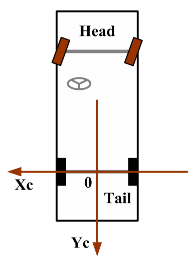


Figure 3. Car's coordinates

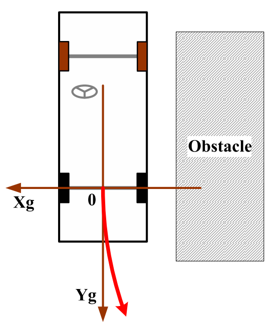


Figure 4. World coordinates

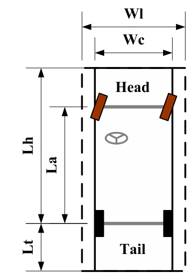


Figure 5. Bodywork variables

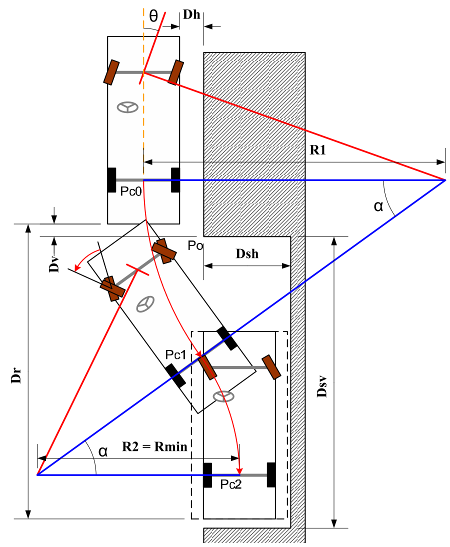


Figure 6. Motion variables

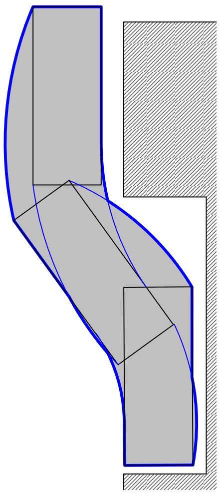
III. Path Planning

A. Definition of coordinates

In order for analytic calculation, we must define car's coordinates and world ones. The car's coordinates are induced by (X_c, Y_c), with the origin at the midpoint of the rear axle, and the X_c-axis points to the left of the car and the Y_c-axis points backwards. The world coordinates are superposed over the car's coordinates at the end-point of Step 1, and induced by (X_g, Y_g).

B. Definition of variables

The definition of variables is also divided into two groups, one for the bodywork, and the other for the motion. A detailed definition makes the latter calculation intelligible. The bodywork variables are defined as follows and illustrated in Figure 5:
L_h: Length between rear axle and car head.
L_t: Length between rear axle and car tail.
L_a: Length between rear axle and front axle.
W_c: Width of the car.
W_l: Width of the safety square, which should be equal to the width of a parking space.
Figure 6 illustrates a scene of parallel parking region surrounded by obstacles and the parking process of a car. The obstacles are usually other cars and road shoulders. Points Pc0, Pc1 and Pc2 are three key points of the trajectory. Point Po is a corner of the obstacle, which is the most possible collision point in Step 3. The motion variables are defined as follows:
D_{sh}: Width of the non-obstacle region.
D_{sv}: Length of the non-obstacle region.
D_h: Lateral distance between the car and obstacles at the end of Step 1.
D_v: Longitudinal distance between the car tail and Po.
D_r: Longitudinal reversing distance of the two arcs.
R_1: Turning radius of Step 3.
R_2: Turning radius of Step 5, equally to the minimum radius R_{min}.
α: Front wheel angle of Step 2.
θ: Angle of the arc of Step 3, equally to that of Step 5.



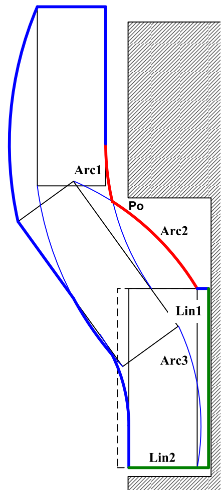


Figure 7. Swept area of the motion Figure 8. Constraint boundaries

C. Analytic declaration of the problem

Figure 7 shows the swept area of the whole parking motion. And Figure 8 highlights those lines and arcs of the boundary of the area with respect to collision constraints. Arc1 and Arc2 are the nearest arcs to Po in Step 3 and 5 separately. Lin1 and Lin2 are edges of the safety square that are near to the inner side of the non-obstacle region.

Before discussing the constraints, we firstly list some main expressions in the world coordinates. From the relationship between the car's position and the obstacles' positions in Figure 6, we obtain the coordinates of Po $(-D_h - W_c/2, D_v + L_t)$.And the turning radius of Step 3 is $R_1 = L_a cot\theta$.
In order for formulating convenient, Pc2, the concatenating point of Step 3 and 5, is written separated as:

$$Pc_2.x = -(R_1 + R_2) * (1 - cos\alpha)$$
$$Pc_2.y = D_r = (R_1 + R_2) * sin\alpha$$

The distance from Po to Arc1 and Arc2 are $D_{Po-Arc1}$ and $D_{Po-Arc2}$ respectively, and their square expressions are

$$\begin{aligned} D^2_{Po-Arc1} =& (R_1 - W_c/2)^2 - [-(D_h + W_c/2) + R_1]^2 \\ &- (D_v + L_t)^2 \\ D^2_{Po-Arc2} =& [-(D_h + W_c/2) - (Pc_2.x + R_2)]^2 \\ &+ (D_v + L_t - D_r)^2 - (R_2 + W_c/2)^2 - L_h^2 \end{aligned}$$

With these expressions above, now we come to the constraints.
Firstly, considering the end of parking, the car should be inside the non-obstacle region. Thus in the longitudinal direction, the distance from Po to the tail of the goal position must be bigger than the car's length, and smaller than the region's length, that yields:

$$L_t + L_h < D_r - D_v < D_{sv} \tag{1}$$

And in the lateral direction, the car's left edge must be on the right of the region's left side, and the car's right edge must be on the left of the region's right side. For the sake of higher safety guarantee, a safety square is mounted on the car. The square has the same length of the car, and the same width of a parking space. Practically, it will be better if it is longer than the car, but for an easer modal, only the width is in consideration here. Besides, when we get the material parameters of the car's bodywork, it will be easy to find out that the swept area doesn't go outside of the square. And this is a simplified measure to assure safety. Accordingly, the inequality comes out to be:

$$W_c + D_h \leq -Pc_2.x \leq W_c/2 + D_h + D_{sh} - W_l/2 \tag{2}$$

The negative sign of the mid item owes to the definition of the world coordinates.
Secondly, the square expressions of $D_{Po-Arc1}$ and $D_{Po-Arc2}$ should be positive for rational value. Hence we get the follows:

$$D^2_{Po-Arc1} \geq 0 \tag{3}$$

$$D^2_{Po-Arc2} \geq 0 \tag{4}$$

Finally, the two angles in Figure 6 should be practical. θ, the front wheel angle, is related to turning radius, and has an upper boundary caused by mechanical constraints and reflected on R_{min}. As for α, it should not permit the car tail to reach the inner side of the non-obstacle region, even when the radius of Step 3 shrinks to the minimum. Therefore the two ranges are:

$$0 < \theta < atan(L_a/R_{min}) \tag{5}$$

$$0 < \alpha < asin((D_{sv} + D_v)/(2 * R_{min})) \tag{6}$$

The six constraints above constitute a constraint set in the world coordinates.

D. Optimization and results

There are two optimizing objects we aim at. One claims that the car goes the most into the non-obstacle region laterally, and takes the least distance longitudinally, which is called the best position parking. And the other expects the farthest from Po to Arc1 and Arc2 to reduce collision probability, which is called the safest process parking. The former means that $Pc_2.x$ tends to the maximum absolute value in negative region, and meanwhile $Pc_2.y$ tends to the positive minimum. Thus we obtain the objective function as (7) and its configuration plotted in Figure 9. The green area shows the feasible area.

$$f(\theta, \alpha)_{BestPos} = Pc_2.x * Pc_2.y \tag{7}$$

The latter means that $D_{Po-Arc1}$ and $D_{Po-Arc2}$ tends to their maximum corporately. The resultant objective function is shown as (8) and in Figure 10.

$$f(\theta, \alpha)_{SafestPro} = D^2_{Po-Arc1} * D^2_{Po-Arc2} \tag{8}$$

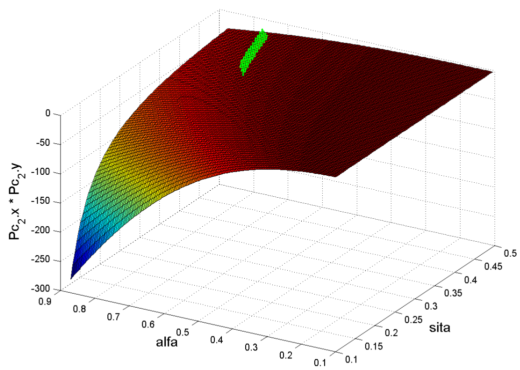


Figure 9. Best position objective function

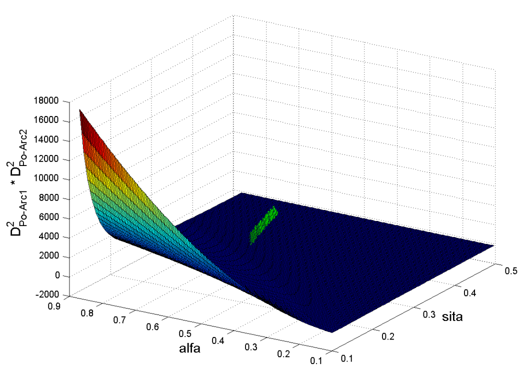


Figure 10. Safest process objective function

Replacing the symbols of the constraint set and the objective functions with the values in Table I, and implement the optimization via MATLAB Optimization Toolbox, yields the optimal results in Table II.

Table I
SIMULATION PARAMETERS

Sym.	Val.	Sym.	Val.
Lh	3.6m	R2	5.0m
La	2.7m	Dh	0.6m
Lt	1.1m	Dv	0.3m
Wc	1.8m	Dsv	7.3m
Wl	2.4m	Dsh	2.2m

Table II
OPTIMAL RESULTS

Object function	θ	α
Best position	0.327 rad	0.632 rad
Safest process	0.359 rad	0.679 rad

IV. CONCLUSION

In this paper, an approach of parallel parking path consisting of two arcs is proposed. The process of backing up is formulated into a collision constraint set. Two optimizing objects are considered and resolved respectively. This scheme is to help practical uses in parking assistant systems.

ACKNOWLEDGMENT

This research was supported in part by the National Natural Science Foundation of China under Grant 90820015 and 90820302.

REFERENCES

[1] M. Brady, *Robot motion: Planning and control*. the MIT Press, 1982.

[2] J.-C. Latombe, *Robot motion planning*. Springer, 1991.

[3] J. Laumond, S. Sekhavat, and F. Lamiraux, “Guidelines in nonholonomic motion planning for mobile robots,” *Robot motion planning and control*, pp. 1–53, 1998.

[4] S. LaValle, *Planning algorithms*. Cambridge Univ Pr, 2006.

[5] S. Sekhavat and J. Laumond, “Topological property for collision-free nonholonomic motion planning: The case of sinusoidal inputs for chained form systems,” *Robotics and Automation, IEEE Transactions on*, vol. 14, no. 5, pp. 671–680, 1998.

[6] D. Tilbury, R. Murray, and S. Shankar Sastry, “Trajectory generation for the n-trailer problem using goursat normal form,” *Automatic Control, IEEE Transactions on*, vol. 40, no. 5, pp. 802–819, 1995.

[7] J. Reeds and L. Shepp, “Optimal paths for a car that goes both forwards and backwards,” *Pacific Journal of Mathematics*, vol. 145, no. 2, pp. 367–393, 1990.

[8] T. Fraichard and A. Scheuer, “From reeds and shepp’s to continuous-curvature paths,” *Robotics, IEEE Transactions on*, vol. 20, no. 6, pp. 1025–1035, 2004.

[9] B. Muller, J. Deutscher, and S. Grodde, “Trajectory generation and feedforward control for parking a car,” in *Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control, 2006 IEEE*. IEEE, 2006, pp. 163–168.

[10] ——, “Continuous curvature trajectory design and feedforward control for parking a car,” *Control Systems Technology, IEEE Transactions on*, vol. 15, no. 3, pp. 541–553, 2007.

[11] J. Laumond, P. Jacobs, M. Taix, and R. Murray, “A motion planner for nonholonomic mobile robots,” *Robotics and Automation, IEEE Transactions on*, vol. 10, no. 5, pp. 577–593, 1994.

[12] D. Kim and W. Chung, “Motion planning for car-parking using the slice projection technique,” in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*. IEEE, 2008, pp. 1050–1055.

[13] D. Kim, W. Chung, and S. Park, “Practical motion planning for car-parking control in narrow environment,” *Control Theory & Applications, IET*, vol. 4, no. 1, pp. 129–139, 2010.