

## Research Article

# Experimental Test of Artificial Potential Field-Based Automobiles Automated Perpendicular Parking

Yiqun Dong,<sup>1,2,3</sup> Youmin Zhang,<sup>2</sup> and Jianliang Ai<sup>1</sup>

<sup>1</sup>Department of Aeronautics and Astronautics, Fudan University, Shanghai, China

<sup>2</sup>Department of Mechanical and Industrial Engineering, Concordia University, Montreal, QC, Canada

<sup>3</sup>School of Electric and Electronic Engineering (EEE), Nanyang Technological University, Singapore

Correspondence should be addressed to Yiqun Dong; [dongyq@ntu.edu.sg](mailto:dongyq@ntu.edu.sg)

Received 7 June 2016; Accepted 18 September 2016

Academic Editor: Abdelaziz Bensrhair

Copyright © 2016 Yiqun Dong et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Automobiles automated perpendicular parking using Artificial Potential Field (APF) is discussed in this paper. The Unmanned Ground Vehicle (UGV) used for carrying out experiments is introduced first; UGV configuration, kinematics, and motion controller are included. Based on discretized form of the parking space, the APF is generated. Holonomic path for the vehicle parking is found first; path modification to satisfy minimum turning-radius constraint is performed based on Reeds-Shepp curve connections. Optimization efforts are included to remove extra maneuvers and to reduce length of the path. Afterwards waypoints are generated as reference for the vehicle to track. Perpendicular parking tests with several different start configurations are demonstrated; based on the test results the automated parking framework proposed in this paper is considered to be effective.

## 1. Introduction

Recent decades have witnessed increasing of automobiles usage in human lives. Due to parking spaces constraints, it could be a challenge for human drivers to squeeze a car into the congested parking layouts. With the development of informant and intelligent systems on automobiles, an Automated Parking System (APS) could be an alternative to relieve this challenging task.

In industrial field many automobile manufactures claimed the APS has been successfully implemented on their products; some test videos have been released (which readers may find easily from open Internet source); the technique details, however, never run into open and further tests are not disclosed.

In academic area the APS research starts earlier. Though many of the works focused on multiple vehicles management in garage turned out to be effective [1–4], the work described in this paper is more concerned with path planning and navigation/control of single vehicle. Some works using empiricism-based methods have been proposed [5–8], and

the claimed results from these works seem to be success. Objective of this paper, however, is to develop a framework which could be verified and retuned for multiple scenarios (vehicles); a more valid choice is to build up some real physics/geometrics-based algorithms, especially since transferability of the empiricism-based techniques is doubtful.

For the real physics/geometrics-based techniques that have been proposed for automated parking, path planning is the core part and has been addressed in some of the work using “elementary maneuvers”; that is, several segments of the automobiles movement are proposed first, and a final concatenate function is performed to link the final path [9–12]. This method depends heavily (and perhaps fully) on geometrics; all of the computational work needs to be done in real-time. But there does not exist a universal standard for the subdivision of movement segments, and it could be difficult to do this work on the field (online).

What the authors have adopted in this report is the geometric-to-feasible approach, and Artificial Potential Field (APF) is used. Specifically, the path planning consists of two

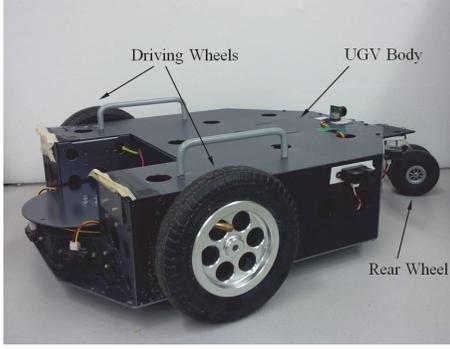


FIGURE 1: Configuration of the UGV used in experimental tests.

steps: (a) plan a geometrical collision-free path and (b) modify the path to satisfy constraints. While the second step could be accomplished using geometrical approaches, the APF used in the first step is generated off-line, which also serves as a standard/rule for connecting the geometrical collision-free path. Compared with the elementary-maneuver method, this line of path planning framework is considered to be more applicable.

This paper is organized as follows. The model vehicle is introduced in Section 2, and APF is briefly discussed in Section 3. General framework for the APF-based parking is illustrated in Section 4, and experimental test results are shown in Section 5. In Section 6 a general conclusion of this paper is delivered; some topics that warrant author's future attention are also listed.

## 2. Vehicle Kinematics and Motion Controller

**2.1. Vehicle Kinematics.** The Unmanned Ground Vehicle (UGV) used for carrying out experiments is shown in Figure 1. This vehicle is 66 cm long and 42 cm wide; it is driven by two front wheels, each actuated by an electric-motor; rear wheel is used for balance support only. There is no steering wheel on the UGV; heading angle adjustment of the vehicle is achieved by differentially driving the two front wheels.

Velocity of the two front driving wheels is controlled by the input voltage. Prior to motion control a mapping function from control input to the wheel speed is decided; see Figure 2. Note that control input of the UGV is restricted in the range of 0.05~0.10. For input below 0.075 (0.05~0.075) the UGV reverses, and for input above 0.075 (0.075~0.10) it moves forward. At an incremental value of 0.0025, voltage input of the two driving wheels is increased, and the functional relation (solid blue and green lines on Figure 2) is returned based on the measured speed at different voltage input (black circles).

Motion equations of the UGV are

$$\dot{x} = V_C \cos \theta,$$

$$\dot{y} = V_C \sin \theta,$$

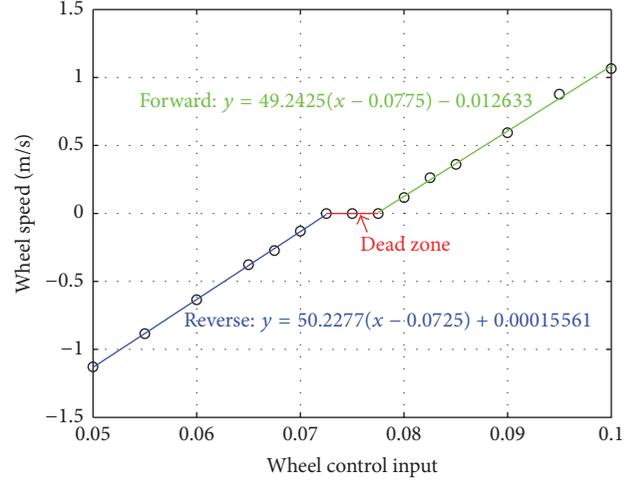


FIGURE 2: Mapping relation from control input to wheel speed.

$$\begin{aligned} \dot{\theta} &= \frac{(V_R - V_L)}{W}, \\ V_C &= \frac{(V_R + V_L)}{2}. \end{aligned} \quad (1)$$

In (1), speeds of the two driving wheels (right  $V_R$  and left  $V_L$ ) are input; output is the vehicle position  $(x, y)$  and heading angle  $(\theta)$ . The width between the two wheels is  $W = 0.40$  m, and speed of axle center of the two driving wheels is  $V_C$ . In our tests minimum turning radius of the vehicle is the vehicle length (66 cm). Also for all the tests discussed herein, the vehicle speed (wheels axle center speed) stays at 0.20 meters per second once it moves.

**2.2. Pure-Pursuit Motion Controller.** Pure-pursuit motion controller is adopted in this paper for driving the vehicle. Firstly proposed and applied in [13], this controller has been widely used for ground and aerial vehicles path tracking; see [14, 15]. A conceptual plot of the pure-pursuit controller is shown in Figure 3. On the plot, the vehicle starts turning at point  $O$ ,  $P$  is the origin of vehicle turning arc,  $r$  is the vehicle turning radius, and goal point of vehicle is located at  $G$ .

Given Figure 3,

$$\begin{aligned} r^2 &= d^2 + y^2, \\ r &= d + |x| = d - x. \end{aligned} \quad (2)$$

Based on (2),

$$r = -\frac{x^2 + y^2}{2x} \triangleq -\frac{L^2}{2x}. \quad (3)$$

In (3),  $L \triangleq \sqrt{x^2 + y^2}$  is defined as the look-forward distance. Note that when lateral displacement  $x$  is 0 (which denotes the vehicle pointing to the goal directly), the infinite turning radius corresponds to a movement along a straight-line

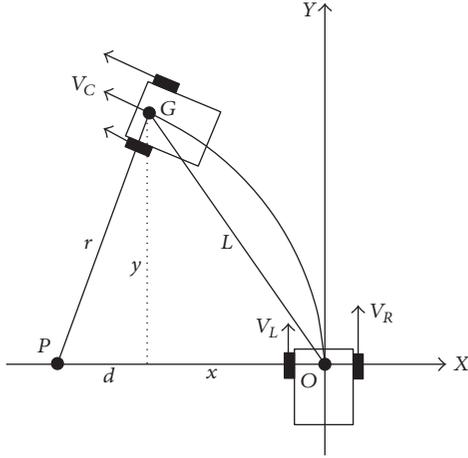


FIGURE 3: Geometrics of pure-pursuit motion controller.

(vehicle heading to the goal directly). Also in (3), as on Figure 3 a positive turning radius indicates the origin being positioned to port side of the vehicle ( $x < 0$ , vehicle turning counter-clockwise).

Given the motion equations in (1),

$$\dot{\theta} = \frac{(V_R - V_L)}{W}, \quad (4a)$$

$$V_C = \frac{(V_R + V_L)}{2}$$

and the vehicle axle center speed

$$V_C = r\dot{\theta}. \quad (4b)$$

We could have

$$V_R - V_L = W \frac{V_C}{r}, \quad (5)$$

$$V_R + V_L = 2V_C.$$

Substitute  $r$  from (3) into (5):

$$V_R = V_C \left( 1 - \frac{W}{L^2} x \right), \quad (6)$$

$$V_L = V_C \left( 1 + \frac{W}{L^2} x \right);$$

desired speed for right and left wheels could be determined. Using the speed obtained in (6), vehicle trajectory will travel to the goal point. Also given the mapping relation presented in previous section, desired control input for the vehicle driving wheels could be reversely decided.

In (6) note that  $x$  is the lateral displacement of goal point expressed in vehicle body axis; speed of the vehicle (axle center) is donated as  $V_C$ . As for the vehicle look-forward distance, however, to the author's best knowledge there does not exist a universal standard. In [15] it was mentioned that the controller is Hurwitz stable if  $L$  is beyond a certain level. In some of the research efforts the look-forward distance

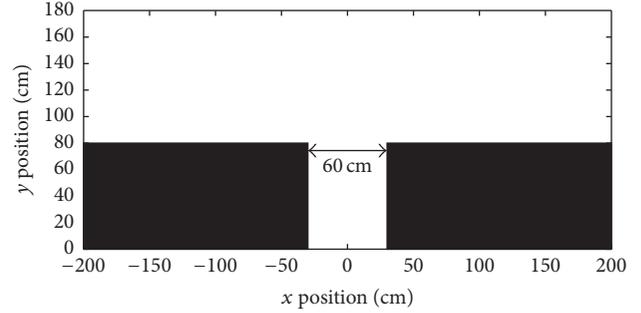


FIGURE 4: Plot of the perpendicular parking plot.

was determined using experimental tests [14]. For the work discussed herein, vehicle speed is 0.20 meters per second; via extensive simulation and experimental validations, a look-forward distance of 0.15 meters is used.

### 3. Artificial Potential Field (APF) Generation

In order to use a potential field for parking, prior knowledge (geometrics, size, etc.) of the parking layouts needs to be characterized first. In daily lives, however, different parking spaces have appeared, and it is hard to trace all of them. Since normally the perpendicular parking layouts are the most "uniform" one, we focus on the discussion of perpendicular parking only. Also the online determination of the layout geometrics (using cameras, odometry, etc.) falls out of the scope of this paper; we measure and extract the layouts sizes prior to the test and generate the APF in advance.

The perpendicular parking layout used in this paper is shown in Figure 4; drivable areas are plotted in white; obstacles (boundaries) are in black. As in Figure 4 the parking garage is 1.5 times the width (60 cm) and 1.2 times the length (80 cm) of the vehicle. Passage in front of the garage represents a double-lane road, which is 2.5 times the UGV width (100 cm). Length of the passage represents the area wherein the vehicle could maneuver; in this paper it is 6 times (400 cm) the length of UGV (all the values are obtained following the proportion of car size and parking layouts dimensions in real-life).

Prior to APF generation, the working space (parking lot) needs to be discretized. In this paper a resolution of 1 cm  $\times$  1 cm grid is used. After the discretization, a Manhattan contour map is obtained; see Figure 5. Generally for all the points in the white drivable area, the Manhattan distance connects this point to the closest boundary of the working space, and the points that claim the same Manhattan distance are concatenated. Given the Manhattan contour, kinks on the map are connected; see Figure 6. This plot represents the topological skeleton of the parking space. Geometrically, this diagram represents a path that is located as far away as possible from the working space boundaries.

To generate the APF, a goal point needs to be assigned to induce attractive forces in the potential field. For the perpendicular parking discussed herein, two cases might arise; that is, the vehicle could head forward into or reverse

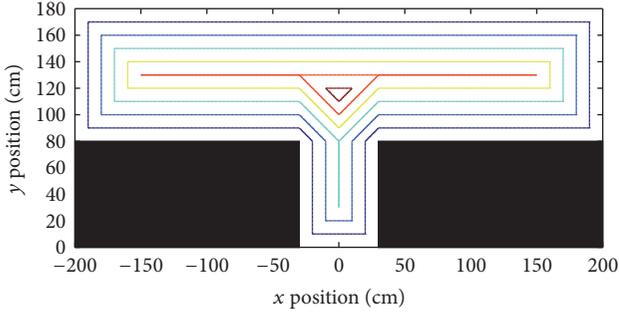


FIGURE 5: Manhattan distance contour of the parking lot.

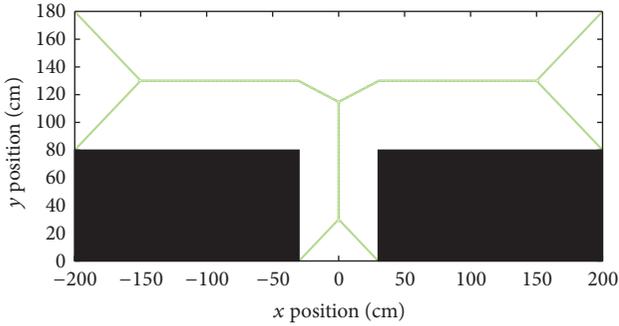


FIGURE 6: Skeleton plot of the parking lot.

backward into the parking garage. In real lives based on authors' experiences this is largely a personal-preference issue; or in this paper, these two parking cases are considered unbiased; that is, both are valid.

Without the loss of generality, the parked UGV is placed in the center of the parking garage; due to the two different vehicle reverse-into or head-into cases, front axle center of the UGV could be in  $T_1$  (reverse-into) or  $T_2$  (head-into) as in Figure 7. Given  $T_1$  and  $T_2$ , the minimum skeleton-distance of a certain point in the working space can be calculated; see the example of  $S$  in Figure 7. Basically  $S$  connects to the skeleton using a shortest path (perpendicular to the closest skeleton branch), and along the skeleton it travels to the goal point (an example as to  $T_1$  is shown in the plot); the overall distance  $S$  has covered (blue thick lines) can be calculated. For all the points in the working space, two values corresponding to vehicle reverse-into ( $V_p(T_1)$ ) and head-into ( $V_p(T_2)$ ) cases are stored. Normally the final potential field is generated using a linear form as

$$V_p = \xi_1 V_p(T_1) + \xi_2 V_p(T_2), \quad (7)$$

wherein  $\xi_1$  and  $\xi_2$  are real numbers. Given the APF in (7), the next step of path planning is achieved by locally moving the vehicle along the maximum gradient drop of the potential field. The potential function in (7) might provoke some competition between the two goal points; that is, at a certain location the attractive forces from the two goal points might contradict each other. If the two forces are equal, then a local minimum is encountered, and the vehicle is stuck at this location.

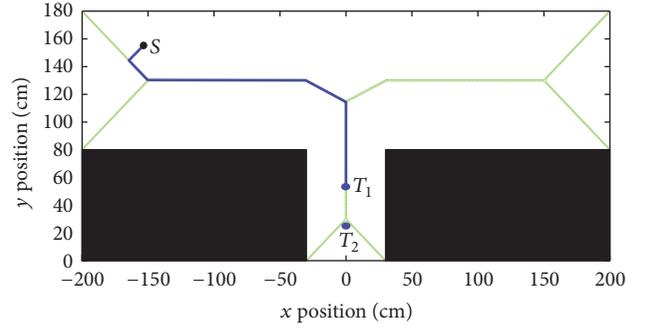


FIGURE 7: Plot of a skeleton-distance example.

One naive approach to avoid this local minimum problem is to use a potential function as

$$V_p = \min \{V_p(T_1), V_p(T_2)\}. \quad (8)$$

Indeed via (8), there does not exist a "local" minimum; a global minimum will fall concurrently to the two goal points. The potential problem of (8), however, is that if the vehicle intends to head into the garage and when the vehicle axle front achieves goal point  $T_1$ , the potential in (8) will drop to zero. While the major part of the vehicle body is still outside the garage, the vehicle stops. A simple way to avoid this problem is to rewrite the potential as

$$V_p = \min \{V_p(T_1), V_p(T_2)\} + \xi \max \{V_p(T_1), V_p(T_2)\}. \quad (9)$$

In (9) a disturbance term of  $\xi$  from the other goal point is used, and in this manner the attractive force from other goal will continue to move the vehicle, even if the front axle has reached a (wrong) goal. The selecting of  $\xi$  stands for the art of potential function tuning. In author's work, simulations using  $\xi$  between 0.05 and 0.20 were conducted; no problem was encountered. In the real tests work herein, a disturbance term of  $\xi = 0.10$  is used.

#### 4. APF-Based Path Planning

The APF-based path planning consists of three steps:

- (a) Find a collision-free holonomic path using geometrical information.
- (b) Perform subdivision on the path till all endpoints are linked by a collision-free curve which satisfies the nonholonomic minimum turning-radius constraints.
- (c) Run an optimization to remove extra maneuvers and reduce overall length of the path; generate waypoints along the path.

In the following we detail each of the three steps.

*4.1. APF-Based Collision-Free Holonomic Path.* As in previous contents we adopt the assumption that geometrics of the

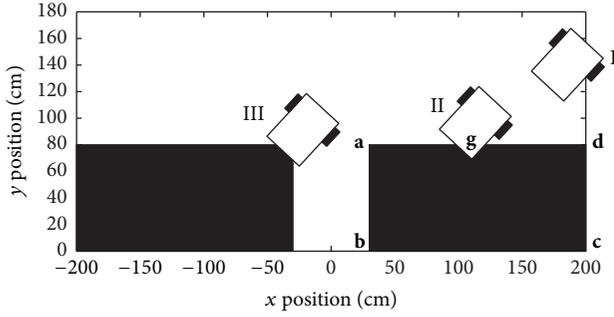


FIGURE 8: Conceptual plot of three collision cases.

parking layouts are preknown and fixed. APF of the parking layouts is generated in advance. In this chapter we discuss the APF-based holonomic (geometrical) path generation.

Collision detection is adopted herein to ensure that the vehicle moves only in drivable areas. Given the parking lot in Figure 4, generally three collision cases might arise; see Figure 8. In case I, part of the UGV body goes outside the passage. Given coordinates of the four points which enclose the passage and coordinates of the UGV body, this case can be examined. Cases II and III correspond to rectangular intersection problems. For case II, given the four coordinates (**a**, **b**, **c**, **d**) of the nondrivable black area (obstacle) and a certain point on UGV body (**g**), by checking

$$\begin{aligned}
 \Pi_1 &= \mathbf{ab} \times \mathbf{ag}, \\
 \Pi_2 &= \mathbf{bc} \times \mathbf{bg}, \\
 \Pi_3 &= \mathbf{cd} \times \mathbf{cg}, \\
 \Pi_4 &= \mathbf{da} \times \mathbf{dg},
 \end{aligned} \tag{10}$$

wherein the vector cross-multiplication is used (**ab** indicates the vector pointing from **a** to **b**), for the four cross-multiplied values in (10), if all the algebraic signs are identical (positive for a counter-clockwise **abcd** sequence on the figure), the inspected point (**g**) is laid within the obstacle. The same criteria can be applied to case III, which indicates that the obstacle is laid inside the UGV. For cases II and III a collision will be returned.

Given vehicle (front axle center) start position, based on the potential field generated in (9), the vehicle movement is defined locally along the maximum gradient drop of the artificial potential. For each step of the move, however, collision detection is conducted. If in a certain step a collision is returned, there is not a collision-free path for the automated parking. For all the tests in this paper, a collision-free path exists.

**4.2. Reeds-Shepp Trajectory-Based Nonholonomic Solution.** The path generated from Section 4.1 might be sharp; that is, since the nonholonomic minimum turning-radius constraint of the vehicle is not included, there could be sudden turning or lateral drift in the path connection. The Reeds-Shepp (RS) trajectory is adopted to modify the path in order to meet these constraints.

Proposed in [16], the RS trajectory generally represents a curve (subject to a minimum curvature) which connects the vehicle from a start to the goal configuration (heading and position). Also in RS trajectories reverse movement is allowed. Of all the possible connections subject to minimum curvature, the RS trajectory claims the shortest length. Totally 48 different RS trajectory cases might arise; please refer to [16] for more details.

Given the holonomic path from Section 4.1, two endpoints of this path (vehicle start and goal configuration) are linked using RS trajectory. If the RS link is collision-free, this path will be passed on for next step. Or if the RS link is not collision-free, the vehicle configuration halfway along the holonomic path is used as a subgoal, and two subproblems arise as to connecting the four endpoints of the two subpaths using RS trajectory. This process will be performed repeatedly till all the points along the holonomic path are connected using collision-free RS trajectories. Final path will claim a feasible trajectory which connects the vehicle from the start to goal configuration, nonholonomic constraint being satisfied.

**4.3. Path Optimization and Waypoints Generation.** Given the feasible path from Section 4.2, by randomly picking up two points on the path, the RS trajectory is used to connect these two points. If the RS connection is collision-free (and certainly it will be shorter than the original path), the original path is replaced by the new RS trajectory; or if not, an empty trial is returned. When the accumulated empty trials number exceeds a certain level (100 times), the optimization process is completed. Since in this process we are always trying to substitute the original path using a shorter connection, overall length of the path evolves at a nonincreasing manner. And since both the original and new RS connections satisfy the nonholonomic minimum turning-radius constraint, the new path will also be a feasible one subject to the constraint.

Along the optimized path, discrete waypoints are generated as subgoals to guide the vehicle from the start to goal configuration. Beginning from the vehicle start configuration (also added as the first waypoint), points along the path are checked; if distance between this point and previous waypoint exceeds a certain level, this point is designated as a new waypoint, and this process will be repeated till end of the path (vehicle goal configuration). Generally speaking, using of more waypoints yields a more detailed description of the planned path, although distance between every two consecutive waypoints will be decreased, and an issue of the pure-pursuit controller tracking stability could arise. In the tests discussed herein, via extensive simulation and experimental validations, a distance of 15 centimeters for the waypoints generation is used.

## 5. Experimental Test Results

Final experimental tests are conducted in Concordia Networked Autonomous Vehicles (NAV) laboratory. One important facility of the lab is the Optitrack positioning software. Infrared cameras are used to capture reflective markers fixed on the vehicle. Given location of the marker within camera's

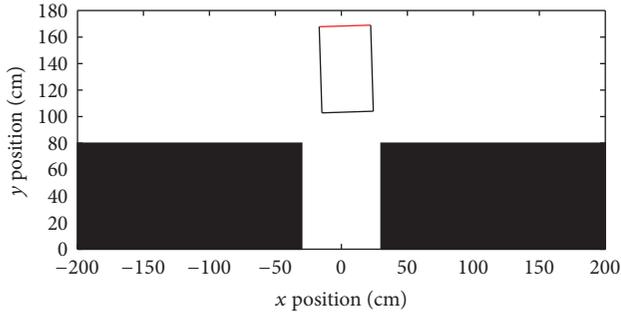


FIGURE 9: Start configuration, experiment #1.

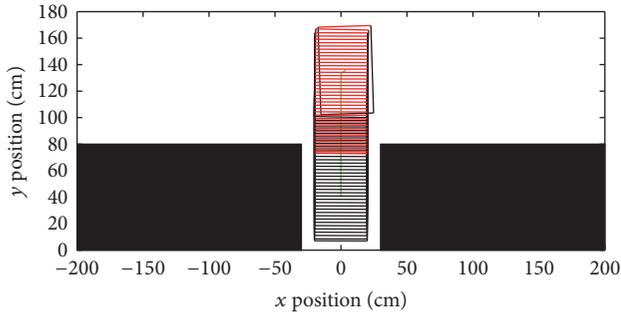
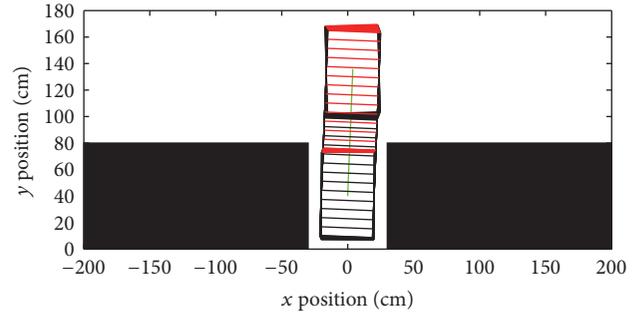


FIGURE 10: Holonomic path, experiment #1.

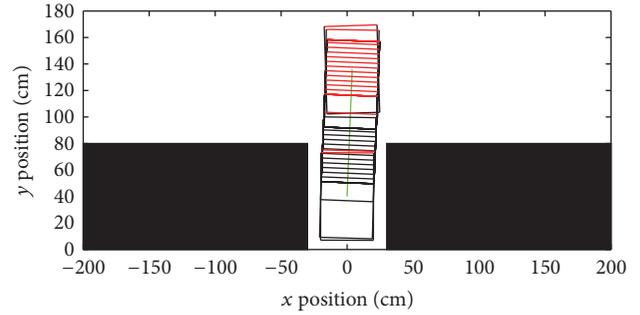
image plane and position/orientation of each camera, coordinates of the marker could be decided. With markers fixed on the vehicle rigid body, position/heading of the vehicle could be calculated; further path planning and navigation/control are conducted. Also as in previous contents we adopt the assumption that the parking layout is fixed, and associated geometrics are preknown. In the real tests we set up the parking environment first (following Figure 4); then we use the Optitrack system to measure the size.

Start configuration of the first test is illustrated in Figure 9. Red thick line represents rear side of the vehicle. Given Figure 9, based on driving experience a direct approach to the garage is possible, which is confirmed by the APF-based holonomic path shown in Figure 10. Using RS path connection, this holonomic path is modified and optimized; see Figures 11(a) and 11(b). Waypoints generation is shown in Figure 12, wherein record of the UGV actual response is also presented. The UGV is navigated through the waypoints one by one, and when it is close enough to the goal configuration, the navigation will stop; given Figure 12, the UGV is finally parked into the garage (video records of all the tests have been posted online at <https://www.youtube.com/watch?v=oVk59rZiZnS>).

Vehicle start configuration in test #2 is similar to #1, the only difference being that vehicle heads out of the garage; see Figure 13. As has been mentioned as long as the vehicle is positioned to center of the garage, vehicle head-into or reverse-into cases are treated as unbiased. Compared with test #1, a valid option for test #2 in Figure 13 is to reverse backward. An APF-based holonomic path connection is

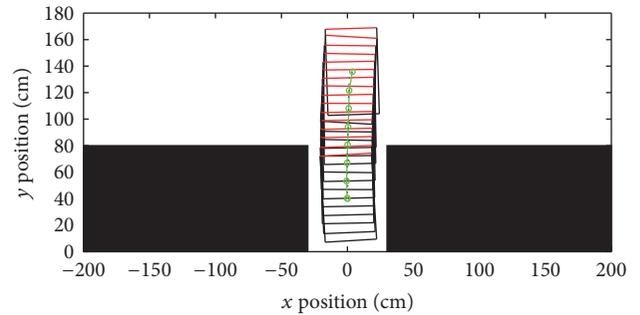


(a)



(b)

FIGURE 11: (a) Nonholonomic path, experiment #1. (b) Optimized nonholonomic path, experiment #1.



—○— Desired  
 ..... Actual

FIGURE 12: UGV path waypoints generation and response history, experiment #1.

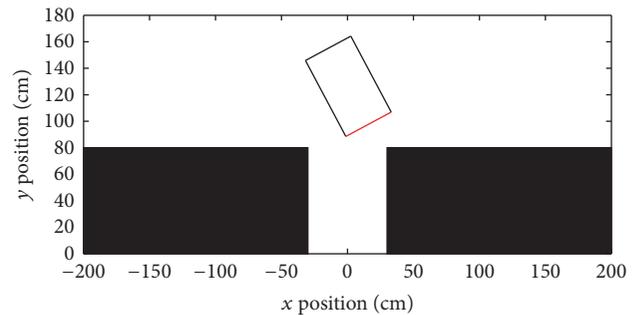


FIGURE 13: Start configuration, experiment #2.

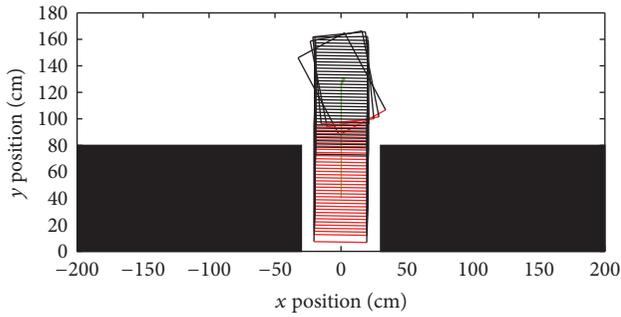
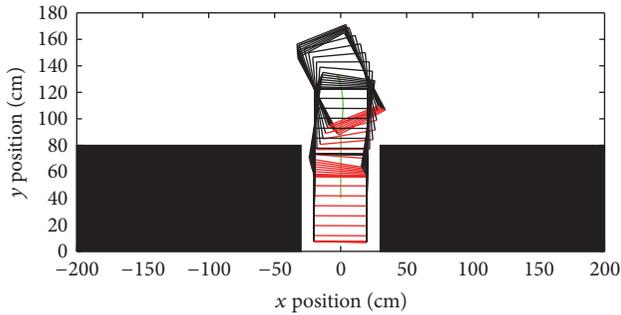
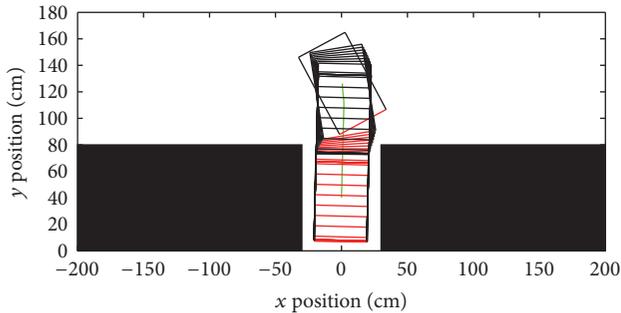


FIGURE 14: Holonomic path, experiment #2.



(a)



(b)

FIGURE 15: (a) Nonholonomic path, experiment #2. (b) Optimized path, experiment #2.

shown in Figure 14; feasible and optimized path are presented in Figures 15(a) and 15(b). Waypoints generation and vehicle response history could be found in Figure 16. Based on Figure 16, again the APF-based vehicle automated parking is considered a success.

Tests #3 and #4 stand for more challenging cases; see the vehicle start configurations in Figures 17 and 21; the vehicle heading is aligned vertically to the goal configuration. Again as has been mentioned vehicle heading into or reversing into the garage is treated as unbiased. In test #3 the vehicle heads into the garage, while in #4 it will reverse into the goal position; see the APF-based holonomic path in Figures 18 and 22. Note that, in Figures 18 and 22, the vehicle slides to center line of the passage first prior to turning into the garage; this is induced by the skeleton of the parking lot—to stay as far away as possible from the boundaries to avoid collision.

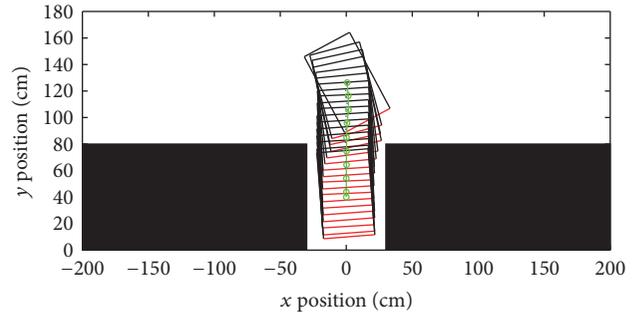


FIGURE 16: UGV path waypoints generation and response history, experiment #2.

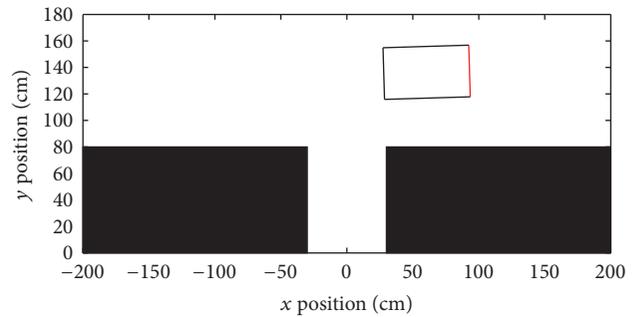


FIGURE 17: Start configuration, experiment #3.

Nonholonomic and optimized solution of the path are presented in Figures 19 and 23. On Figure 23(b), a curb exists, which indicates that the vehicle has to reverse, move forward to adjust the heading, and then move backward again into the goal position (commonly known as a 3-point turning). In Figures 20 and 24, path waypoints for tests #3 and #4 are illustrated; in Figure 20 the vehicle is navigated through the waypoints to the final goal position. In Figure 24 for test #4, however, the curb disappeared; this is mainly due to the large distance (15 cm) which we have used to generate the waypoints; that is, the curb has been stepped over since its length is lower than 15 cm. While the pure-pursuit tracking stability could be guaranteed, final test run is also a success as shown by the vehicle response history in Figure 24. The vehicle could be parked into the goal position.

## 6. Conclusion

This paper reports the experimental tests for automobiles automated perpendicular parking using Artificial Potential Field (APF). Configuration and kinematics of the vehicle are firstly illustrated; pure-pursuit controller for driving the vehicle is introduced. To generate the potential field, parking lots are discretized first; skeleton diagram of the parking space is plotted, and skeleton-distance based potential function is generated. Final path planning for automated parking

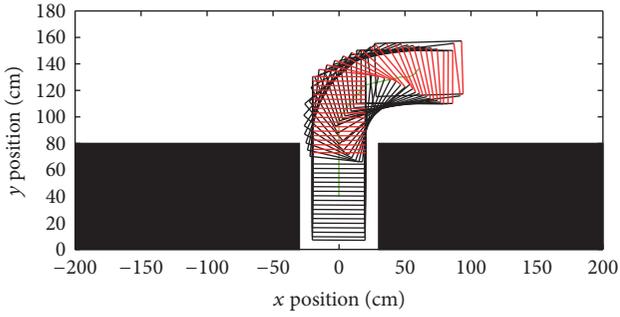


FIGURE 18: Holonomic path, experiment #3.

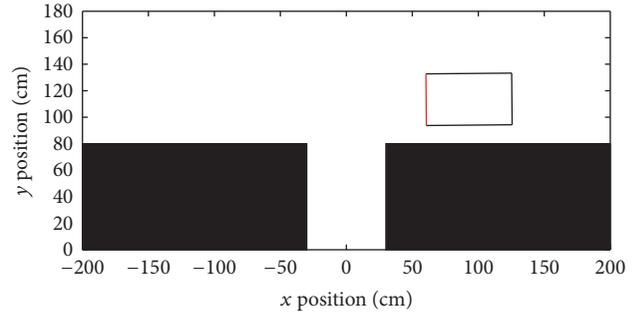
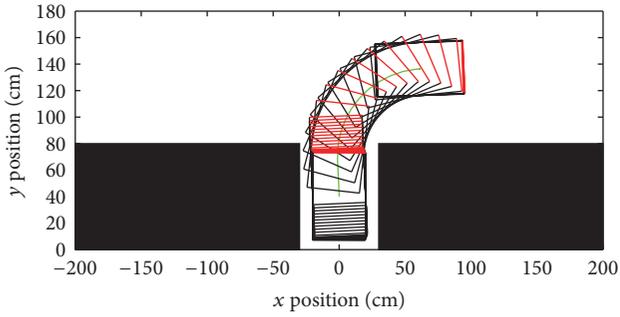
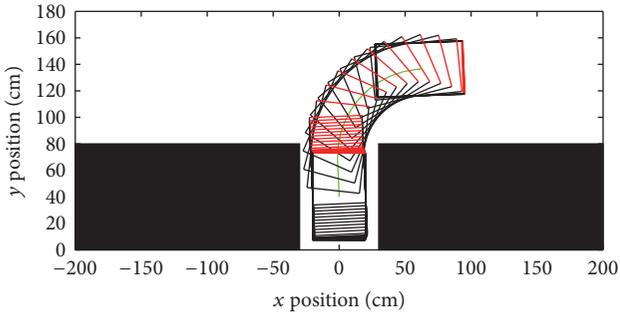


FIGURE 21: Start configuration, experiment #4.



(a)



(b)

FIGURE 19: (a) Nonholonomic path, experiment #3. (b) Optimized path, experiment #3.

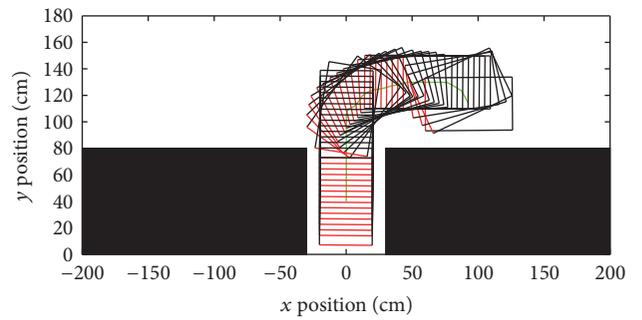
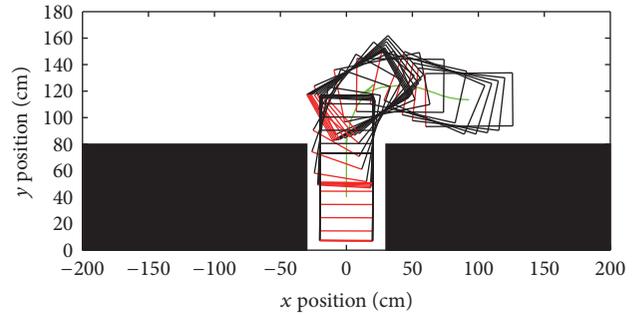
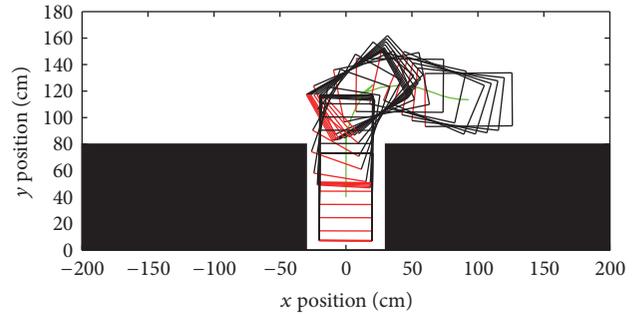


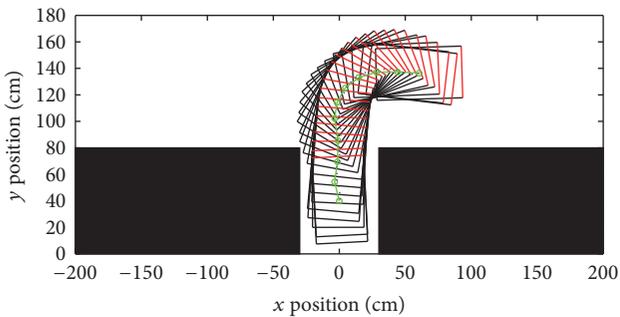
FIGURE 22: Holonomic path, experiment #4.



(a)



(b)



—○— Desired  
 ..... Actual

FIGURE 20: UGV path waypoints generation and response history, experiment #3.

FIGURE 23: (a) Nonholonomic path, experiment #4. (b) Optimized path, experiment #4.

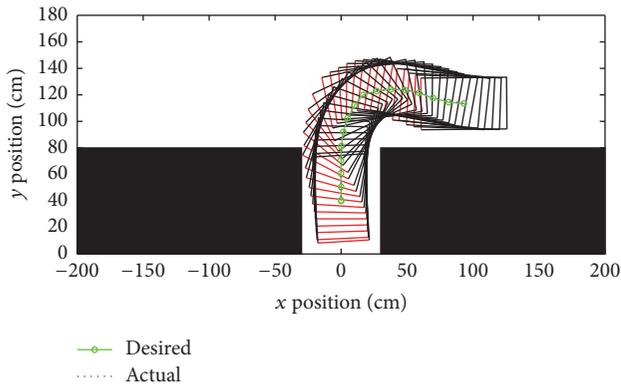


FIGURE 24: UGV path waypoints generation and response history, experiment #3.

involves three steps; holonomic path is found first, non-holonomic path is then connected using Reed-Shepp (RS) trajectory, and finally path optimization is conducted wherein overall length of the path is reduced. Discretized waypoints are generated along the path and will be delivered to the pure-pursuit motion controller for vehicle path tracking.

Test results reported in this paper include four different scenarios. For the first two, the vehicle was placed in front of the garage, vehicle body parallel to the goal configuration; in the other two tests vehicle body was placed along the lane line and 90-degree turning was needed to complete the parking. Holonomic, nonholonomic, and optimized path connections are presented; waypoints generation was illustrated. The vehicle is navigated through the waypoints to the goal configurations. Based on the test results, the proposed automated parking framework is considered to be effective.

Future work may include tests for more scenarios, especially for some “odd” start positions. Also on real cars the parking space detection could be a problem; computer vision-based parking lines recognition has been formulated; further tests will be delivered lately.

## Disclosure

Yiqun Dong is previously a Ph.D. student with Department of Aeronautics and Astronautics in Fudan University Shanghai China.

## Competing Interests

The authors declare that they have no competing interests.

## Acknowledgments

The authors would like thank Mr. Mohammad Ali Askari Hemmat, Dr. Walaaeldin Ghadiry, and Mr. Mohamed Atef Kamel with Department of Mechanical and Industrial Engineering at Concordia University for hardware implementations in the experimental tests. Yiqun Dong is also indebted to China's Scholarship Council (CSC) that supports Yiqun Dong as visiting Ph.D. student to Concordia University.

## References

- [1] H. Al-Kharusi and I. Al-Bahadly, “Intelligent parking management system based on image processing,” *World Journal of Engineering and Technology*, vol. 2, no. 2, pp. 55–67, 2014.
- [2] D. J. Bonde, R. S. Shende, A. S. Kedari, K. S. Gaikwad, and A. U. Bhokre, “Automated car parking system commanded by Android application,” in *Proceedings of the 4th International Conference on Computer Communication and Informatics (ICCCI '14)*, IEEE, Coimbatore, India, January 2014.
- [3] M. Aggarwal, S. Aggarwal, and R. S. Uppal, “Comparative implementation of automatic car parking system with least distance parking space in wireless sensor networks,” *International Journal of Scientific Research Publications*, vol. 2, no. 10, pp. 1–8, 2007.
- [4] M. M. Rashid, A. Musa, M. A. Rahman, N. Farahana, and A. Farhana, “Automatic parking management system and parking fee collection based on number plate recognition,” *International Journal of Machine Learning and Computing*, vol. 2, no. 2, pp. 93–98, 2012.
- [5] Y.-W. Ryu, S.-Y. Oh, and S.-Y. Kim, “Robust automatic parking without odometry using an evolutionary fuzzy logic controller,” *International Journal of Control, Automation, and Systems*, vol. 6, no. 3, pp. 434–443, 2008.
- [6] W. A. Daxwanger and G. K. Schmidt, “Skill-based visual parking control using neural and fuzzy networks,” in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, vol. 2, pp. 1659–1664, October 1995.
- [7] Y. Zhao and E. G. Collins Jr, “Robust automatic parallel parking in tight spaces via fuzzy logic,” *Robotics and Autonomous Systems*, vol. 51, no. 2–3, pp. 111–127, 2005.
- [8] R. J. Oentaryo and M. Pasquier, “Self-trained automated parking system,” in *Proceedings of the 8th International Conference on Control, Automation, Robotics and Vision (ICARCV '04)*, pp. 1005–1010, IEEE, Kunming, China, December 2004.
- [9] G. Casalino, A. Bicchi, and C. Santilli, “Planning shortest bounded-curvature paths for a class of nonholonomic vehicles among obstacles,” *Journal of Intelligent and Robotic Systems*, vol. 16, no. 4, pp. 387–405, 1996.
- [10] M. F. Hsieh and Ü. Özgüner, “A parking algorithm for an autonomous vehicle,” in *Proceedings of the IEEE Intelligent Vehicles Symposium (IV '08)*, pp. 1155–1160, IEEE, Eindhoven, The Netherlands, June 2008.
- [11] C. Pradalier, S. Vaussier, and P. Corke, “Path planning for a parking assistance system: implementation and experimentation,” in *Proceedings of the Australian Conference on Robotics and Automation (ACRA '05)*, Sydney, Australia, December 2005.
- [12] M. B. Oetiker, G. P. Baker, and L. Guzzella, “A navigation-field-based semi-autonomous nonholonomic vehicle-parking assistant,” *IEEE Transactions on Vehicular Technology*, vol. 58, no. 3, pp. 1106–1118, 2009.
- [13] “Integrated mobile robot control,” in *Proceedings of the SPIE, The International Society for Optics and Photonics*, O. Amidi, C. Thorpe, W. Chun, and W. Wolfe, Eds., vol. 1388, pp. 504–523, Boston, Mass, USA, March 1991.
- [14] Y. Kuwata, J. Teo, G. Fiore, S. Karaman, E. Frazzoli, and J. P. How, “Real-time motion planning with applications to autonomous urban driving,” *IEEE Transactions on Control Systems Technology*, vol. 17, no. 5, pp. 1105–1118, 2009.

- [15] S. Park, J. Deyst, and J. P. How, "Performance and Lyapunov stability of a nonlinear path-following guidance method," *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 6, pp. 1718–1728, 2007.
- [16] J. A. Reeds and L. A. Shepp, "Optimal paths for a car that goes both forwards and backwards," *Pacific Journal of Mathematics*, vol. 145, no. 2, pp. 367–393, 1990.



**Hindawi**

Submit your manuscripts at  
<http://www.hindawi.com>

