Advances in Machinery, Materials Science and Engineering Application X
M. Giorgetti et al. (Eds.)

doi:10.3233/ATDE240681

Autonomous Perpendicular Parking of a Car-Like Robot in an Unknown Environment

Timur GAMBEROV [a], Alexander ERYOMIN [a], Tatyana TSOY [a], Edgar Alonso MARTÍNEZ-GARCÍA [b] and Evgeni MAGID [a,c,1]
[a] *Kazan Federal University, Kremlyovskaya str, 35, Kazan, Russian Federation*
[b] *Universidad Autónoma de Ciudad Juárez, Ave. del Charro No. 450 Norte, Zona PRONAF Cd. Juarez Chihuahua, 32315, Mexico*
[c] *HSE University, Tallinskaya str, 34, Moscow, Russian Federation*

Abstract. The trend for vehicle automation in the recent decades caused a significant growth of researches in autonomous navigation of robotized cars, obstacle avoidance, and dynamic path planning. This paper presents an algorithm for autonomous perpendicular parking of an Avrora Unior car-like mobile robot. The parking process includes finding a suitable parking space and a perpendicular park maneuver. The proposed solution uses point clustering to accurately determine a parking space. A free space is qualified as a parking plot automatically when the threshold value of the distance between clusters is reached. The parking maneuver involves aligning the robot to the parking slot and moving along an arc-shaped trajectory. The algorithm uses only LiDAR and wheel encoder data to determine the robot's location relative to parking spaces. The quality of the robot's parking is estimated by the distance to the boundaries of the parking space. The reliability of the algorithm was successfully validated in Gazebo simulation and in real-world experiments.

Keywords. Mobile robot, car-like robot, autonomous driving, autonomous parking, Gazebo

1. Introduction

In the recent decades, automation and integration of robotic systems into various aspects of human activities consistently increases, including vehicle automation. Vehicle automation considers automated driving systems (ADS) and autonomous parking [1]. The earlier are classified according to levels of automation [2], while the later does not have a clear gradation in terms of automation. According to a type of a vehicle motion, a parking process distinguishes a perpendicular parking and a parallel parking.

The perpendicular parking is a maneuver in which a car is positioned orthogonally to a parking space. A trajectory consist of a car alignment relative to a parking slot and linear backward motion [3]. Finding and selecting a parking space, moving to a

---

[1] Evgeni MAGID, Corresponding author, Kazan Federal University, Kremlyovskaya str, 35, Kazan, Russian Federation, HSE University, Tallinskaya str, 34, Moscow, Russian Federation; E-mail: magid@it.kfu.ru.

selected region, and entering a parking slot are important components of the parking process. A parking space search [4] is based on clustering of a point cloud [5], feature extraction [6,7], and an occupancy grid processing [8,9]. A parking location could be selected by a user [10,11] or determined automatically [5]. In general, movement to a parking space is performed using local path planners [12]. A maneuver into a parking slot is classified according to a robot trajectory. A number of algorithms represent parking based on a start of a robot motion from a perpendicular position relative to the parking slot [6, 13-14]. The trajectory includes angular motion along a circle, whose tangent passes through a vertical central line of the parking space. After leveling up, the robot drives straight into the slot. Some algorithms support parking with various initial angular positions of a robot [15-17]. A broad range of vehicle's sensors that provide environmental perception for parking includes 2D [5] and 3D [11] laser rangefinders (LRF), sonars [9,13], monocular cameras [18], depth cameras [19], and satellite systems [13].

Our approach employs a 2D LRF based system. A localization module uses odometry readings to locate the robot. A parking space search involves the robot moving in a straight line along a parking lane. A parking location is determined automatically when a first free slot is found. The robot enters the slot along a circular arc trajectory from a position orthogonal to a row.

2. Parking Algorithm

The algorithm was validated using an autonomous *Avrora Unior* car-like mobile robot with the Ackermann steering geometry (figure 1a). The onboard sensors include four pairs of ultrasonic sensors (in the front and in the back), a Kinect camera (in the front), and a Hokuyo RG-04LX-UG01 2D LRF (in the front) that covers a viewing angle of 240 degrees in a range from 20 to 5600 mm. A virtual robot model (figure 1b) for the Gazebo simulator [20] that corresponds to physical dimensions of the real robot [21] and its sensing capabilities was used for preliminary testing in a virtual environment.

(a) (b)

Figure 1. (a) Avrora Unior Robot, (b) virtual model of the Avrora Unior robot.

An autonomous parking of the Avrora Unior robot includes a parking space search [22] and a parking maneuver. The algorithm collects and analyzes sensory data while searching a free slot. A parking maneuver involves a motion trajectory planning and continuously transferring velocity commands to the robot.

Before starting the parking spot search algorithm, a vehicle should be aligned orthogonally to a parking lane. The vehicle moves along a line of other vehicles and obstacles (e.g., a utility pole) and continuously receives data from LRF, which is further converted into Cartesian space points and stored as array M. Next, clusters of such points (i.e., detected static obstacles) are constructed (figure 2). Two points from array M belong to the same cluster C_k if a Euclidean distance between them is less than the threshold ε. A recursive update function is defined as:

$$f_{n+1}(x) = f\left(f_n(x)\right) \text{ for } n \geq 2 \text{ with } f_1(x) = f(x) \quad (1)$$

Each point in cluster C is compared with a point from M to find the nearest point using the following $f(C)$ function:

$$f(C) = \left\{P_i \middle| d\left(P_i, P_j\right) \leq \varepsilon, \forall P_i \in M, \forall P_j \in C, P_i \neq P_j, \nexists P_i \in C\right\} \quad (2)$$

where P_i, P_j are points of array M and C, respectively, and $d(a, b)$ denotes a Euclidean distance between a and b. Let's set up the initial cluster $C_{k,init} = \{P_0\}$, where P_0 is a point that does not belong to any cluster. Array H_c contains sets of nearest values. Then cluster C_k is equal to a union of all nearest points:

$$H_c = \left\{C_{k,init}, f_1\left(C_{k,init}\right), f_2\left(C_{k,init}\right), \dots, f_n\left(C_{k,init}\right)\right\}, \left|f_n\left(C_{k,init}\right)\right| = 0 \quad (3)$$

$$C_k = \bigcup_{h_m \in H_c, m=[0..n]} h_m \quad (4)$$

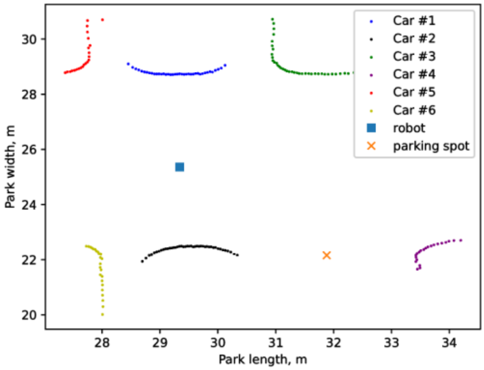

Figure 2. Clustering a point cloud with car points.

Each cluster contains a center of mass, and neighboring clusters are compared with each other. When a threshold value of a distance between clusters is reached, the algorithm searches for the nearest cluster points. The center of a span of the nearest points is marked as a center of a free parking space. After determining a parking point, the robot drives up to a starting point of the parking maneuver.

An autonomous motion of the robot within the parking maneuver is performed by continuously providing corresponding velocity commands to the wheels. A ROS-node that generates the velocity commands receives three types of commands from the parking space searching module: messages "*Drive*", "*Parking*", and "*Stop*" are responsible for moving forward, running the parking maneuver and stopping, respectively. When the robot finds a free space, it drives forward until reaching this space. Then the robot turns the front wheels and drives backwards until it aligns parallel to the parking slot. After completing the turn, the robot drives back, standing in a center of the parking space. The robot's position is generated from odometry readings.

3. Experimental Validation

The Gazebo simulator was used for the algorithm validation. Constructed virtual environments contain walls of a parking area and static car models, which are parked orthogonally to a free lane (figure 3). Three virtual environments with a single available parking space between vehicles were constructed; they differ by a width of the parking space (which is defined by a length and a width of the parked cars, figure 4a) that was selected as 1.5, 2, and 2.5 meters.

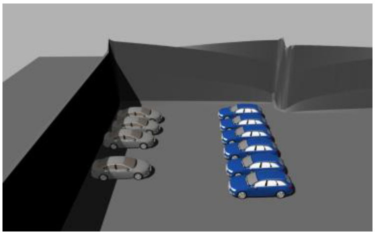

Figure 3. Virtual environment of a parking area.

(a) (b)

Figure 4. Gazebo simulation: (a) a free parking space, (b) the parked robot.

In the Gazebo, 60 experiments were conducted for the three environments, 20 experiments for each. Figure 4b shows a perpendicular parking of the Avrora Unior robot at an equal distance of 1.5 m between the two cars. An evaluation of parking spot search algorithm quality was performed using a number of missed parking spaces and false positives; all experiments, both in virtual and real world, were successful without missing a parking space or providing false positives. An evaluation of parking maneuver algorithm quality was performed using the following metrics: a working time of the algorithm (including calculations and mechanical operations), a distance to a wall, a distance to a frontal marking line (a line that defines an opening at a start of a parking slot), a distance to a right car, a distance to a left car, a count of collisions with other cars, which denote a number of car accidents. A width of a parking space between cars served as an input parameter that allowed evaluating the quality of the algorithm based on other metrics' values.

Next, the algorithm was tested in a real-world parking zone of the university campus at Kremlyovskaya 35 street (figure 5). A parking space of 505x248 cm was selected. Distances from neighboring cars to marking lines (of the Avrora Unior's target parking slot) on the right and on the left were 40 cm and 30 cm, respectively. Three experiments were run using the same setup.

(a) (b)

Figure 5. (a) Parking slot in a real parking area (the marking line is highlighted with a red dotted line), (b) the parked Avrora Unior robot.

The virtual and real-world experimental results are presented in table 1; it includes mean and standard deviation (in parentheses) values of the algorithm working time and distances to a wall, to a frontal marking line, to a right and left car, a distance to a left car to visualize statistics of the experiments. The collision metric was excluded from the table due to a lack of robot interaction with cars (i.e., there we no car accidents). The distances were measured from the center of the robot to a closest point of an obstacle. The analysis demonstrated that mean values of the distances to the left and right cars differ by 0.2-0.4 meter, which corresponds to the robot deviation to the left while maneuvering; it is interesting to note that the deviation was inversely proportional to the width of the slot. In the real world experiments, the Avrora Unior robot found an empty parking space and successfully completed the parking process without hitting any vehicles or leaving the marking. Yet, the abovementioned mean deviation increased; therefore, additional research of the deviation effect will be further conducted.

Table 1. Experimental results.

Parking space width, m	Environment	Time, sec	Distance to wall, m	Distance to marking line, m	Distance to left car, m	Distance to right car, m
1.5	Gazebo	42.43 (0.31)	1.53 (0.02)	3.13 (0.02)	0.56 (0.03)	0.94 (0.03)
2		42.89 (0.17)	1.49 (0.03)	3.17 (0.03)	0.87 (0.02)	1.13 (0.02)
2.5		43.21 (0.13)	1.44 (0.03)	3.22 (0.03)	1.17 (0.02)	1.33 (0.02)
3.28	Real world	86 (1.0)	3.06 (0.12)	1.59 (0.14)	1.74 (0.46)	1.33 (0.30)

4. Conclusions

The paper presented the autonomous perpendicular parking algorithm for the Avrora Unior mobile robot. The algorithm used data from a 2D LRF and clustered them to find a free space. The parking slot was selected automatically based on the first place found. The park maneuver involved aligning the robot relative to the parking slot and moving along an arc-shaped trajectory. The odometry data was used for precise positioning and maneuvering. The reliability and stability of the algorithm were confirmed by tests in the Gazebo simulator and in real world environment in a selected parking area.

Acknowledgement

This paper has been supported by the Kazan Federal University Strategic Academic Leadership Program (“PRIORITY-2030”).

References

[1] Yurtsever E, Lambert J, Carballo A, and Takeda K. A survey of autonomous driving: Common practices and emerging technologies. IEEE Access. 2020;8:58443–69.

[2] Arena F, Pau G, Severino A. An overview on the current status and future perspectives of smart cars. Infrastructures. 2020; 5(7):53.

[3] Imameev D, Shabalina K, Sagitov A, Su K-L, Magid E. Modelling autonomous parallel parking procedure for car-like robot avrora unior in gazebo simulator. Proceedings of the 2020 International Conference on Artificial Life and Robotics (ICAROB2020); 2020; Japan.

[4] Ma Y, Liu Y, Zhang L, Cao Y, Guo S, Li H. Research review on parking space detection method. Symmetry. 2021;13(1):128.

[5] Imameev D, Zakiev A, Magid E, Martinez-Garcia E, Li H. Modelling autonomous perpendicular parking procedure for car-like robot avrora unior in gazebo simulator. 2022 International Conference on Information, Control, and Communication Technologies (ICCT). IEEE; 2022. p. 1-4.

[6] Morales DP, Kermorgant O, Quijada SD, Martinet P. Laser-based control law for autonomous parallel and perpendicular parking. International Conference on Robotic Computing. IEEE; 2018. p. 64-71.

[7] Li Q, Lin C, Zhao Y. Geometric features-based parking slot detection. Sensors. 2018;18(9):2821.

[8] Schmid MR, Ates S, Dickmann J, von Hundelshausen F, Wuensche H-J. Parking space detection with hierarchical dynamic occupancy grids. Intelligent Vehicles Symposium. IEEE; 2011. p. 254-259.

[9] Das S, Yarlagadda Y, Vora PB, Nair SRP. Trajectory planning and fuzzy control for perpendicular parking. 2017 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI). IEEE; 2017. p. 518-523.

[10] Suhr JK, Jung HG. Automatic parking space detection and tracking for underground and indoor environments. IEEE Trans. Ind. Electron. 2016;63(9):5687-5698.

[11] Jiménez F, Clavijo M, Cerrato A. Perception, positioning and decision-making algorithms adaptation for an autonomous valet parking system based on infrastructure reference points using one single LiDAR. Sensors. 2022;22(3):979.

[12] Zhang Y, Chen G, Hu H, Gao Z. Hierarchical parking path planning based on optimal parking positions. Automot. Innov. 2023;6(2):220-230.

[13] Li MH, Tseng PK. Implementation of an autonomous driving system for parallel and perpendicular parking. IEEE/SICE International Symposium on System Integration. IEEE; 2016. p. 198-203.

[14] Han I. Geometric path plans for perpendicular/parallel reverse parking in a narrow parking spot with surrounding space. Vehicles. 2022;4(4):1195-1208.

[15] Kim DJ, Chung CC. Automated perpendicular parking system with approximated clothoid-based local path planning. IEEE Control Syst. Lett. 2020;5(6):1940-1945.

[16] Kim J. Perpendicular parking of car-like robots allowing a cusp on the path. IEEE Access. 2020.

[17] Shin H, Kim MJ, Baek S, Crane CD, Kim J. Perpendicular parking path generation and optimal path tracking algorithm for auto-parking of trailers. Int. J. Control Autom. Syst. 2022;20(9):3006-3018.

[18] Marchthaler R, Wolfmaier R. Deep neural networks for the detection of perpendicular parking spaces. ATZelectronics Worldwide. 2019;14(9):42-47.

[19] Cao B, He Y, Zhuang H, Yang M. Infrastructure-based vehicle localization system for indoor parking lots using RGB-D cameras. J. Shanghai Jiaotong Univ. Sci. 2023;28(1):61-69.

[20] Gazebo. Open Robotics. [cited 2024 Jun 28]. Available from: https://gazebosim.org/home.

[21] Shabalina K, Sagitov A, Su K-L, Hsia K-H, Magid E. Avrora unior car-like robot in gazebo environment. International Conference on Artificial Life and Robotics; 2019. p. 116-119.

[22] Imameev D, Zakiev A, Tsoy T, Bai Y, Svinin M, and Magid E. LIDAR-based parking spot search algorithm. Thirteenth International Conference on Machine Vision; 2021: SPIE; p. 9–15.