Development of an autonomous smart-parallel-parking robot for the prototype of collaborative robots

Wulandari¹*, A Malada¹, M H Arrazi¹, K Priandana¹, S Wahjuni¹, M Hardhienata¹ and A Buono¹

¹Department of Computer Science, IPB University, Bogor 16680, Indonesia

*E-mail: wulandari.ilkom@apps.ipb.ac.id

Abstract. One of the main problems of the collaborative mobile robot application is to share the exact information of the robot itself and the surrounding area. Each robot needs to maintain its stability and positioning in order to achieve the target. As one of the samples of achieving the positioning task, a parallel parking problem was used in this paper. This paper used a car-like robot to do a parallel parking task. Front wheels were steered by using a connected joint and a servo motor. Meanwhile, each of the rear wheels was connected to a motor DC. Four ultrasonic sensors were used to find the distance between the robot and its surrounding (fixed in front, back, middle right, and middle of right - back side). The sensors connected to an Arduino Uno as the main microcontroller. The robot used a positioning algorithm based on the distance to nearest objects. The robot is designed only for parking on the right side of the car with an assumption there is no obstacle in the left side of the car. The experimental results confirmed that our system can solve the parallel parking problem. However, during the test, the output of the sensor was were affected by the noise from the environment. Another problem was the robot hardly to move straight because the rubber tires were not installed neatly. In the future works, the data output needs to filter and corrected and the servo degree needs to be initially corrected based on the chassis and tires angle.

1. Introduction

The exact information of the position of the robot is a fundamental problem in a collaborative mobile robot application. The robots need to communicate with each other in real time in an unknown environment. Besides the robot needs to inform its location to others to find the target [1], another important task in the robot itself is controls their own position [2] to avoid collision with each other and avoid the obstacles [3]. In order to acquire a stable position during target finding, knowledge about the area and surrounding objects are needed [4], especially to avoid the obstacle. In this research, in order to model the problem, we developed a prototype of an autonomous four-wheeled robot and developed a positioning algorithm based on distance obtained by an ultrasonic sensor in order to position itself for parallel parking. Wheeled-robot is chosen in this research because it was easier to control instead of the legged robot [5].

Recently, the development of a self-driving car and autonomous car are rapidly growing [6]. One of the tasks of this system is a parallel parking system. The hurdle faced by this task is positioning the car during parking. Parking itself considered as a difficult matter even for a human. It needs a high skill to do it properly in a short time. Many researchers have tried to optimize the ways of parking in order to reduce the driver's burden. One of the common methods is using the S-shaped trajectory parking path [7].
Commonly, there are three types of parking methods, which are angle parking, perpendicular parking, and parallel parking. Among them, parallel parking has high difficulties even experience driver may cause a collision. Parallel parking means parking the car in line with the other cars parallel to the curb, front bumper to the rear bumper, which left small spaces between the cars. Parallel parking usually occurs on the streets where there are no parking facilities. Due to a small space, high accuracy of the position control is needed to avoid the obstacles such as a neighboring car.

2. Materials
This research adopts a typical four-wheeled robot as a mechanical platform. Each wheel in the rear is connected by a shaft and driven by a motor DC. Meanwhile, the front wheels are linked by a shaft and thus connected to a motor servo. The motor servo is used to steering the angle based on the motor servo rotation.

In the prototype, we used four HC-SR04 ultrasonic sensors. The sensors are fixed in front, back, middle right, and middle of right and back side. The robot is designed only for parking on the right side of the car with an assumption there is no obstacle in the left side of the car. The robot used an Arduino Uno as the main controller. H-bridge L298 is used for controlling the speed of DC motors.

![Schematic diagram of the prototype](image)

Fig 1 Schematic diagram of the prototype

![Experimental setup of the wheeled robot (Top view)](image)

Figure 2. Experimental setup of the wheeled robot (Top view)

3. Methods
Essentially, the goal of this prototype is to park parallel in an obstacle-free environment. The prototype was designed for identifying the free space which fit for the width and length itself. First, to find the fitted length area, the robot will scan the distance using right side ultrasonic sensor while moving forward. If the distance between the robot and the wall are enough for the width of the robot, the length area calculation was started. The width of the robot is fixed on 9 cm and the length is 12 cm. Thus, if the length was considered enough to fit a robot's body length, the robot will start the parking algorithm. Meanwhile, if the area is not enough to fit the robot body, thus it will keep moving forward.

The robot will maneuver the body to right backward with 45 degrees’ rotation if the parking area was found until the back-side distance with an obstacle is around 2 to 3 cm. However, if the robot found that there is an obstacle behind, the robot will move forward for around 2 seconds and repeat the previous step. Thus, if the back-side distance is already appropriate, the motor servo will rotate to 90 degrees to align parallels within the space. If the degree is still not correct, the servo will steer the front wheels.
Lastly, the robot will move based on the appropriate distance with the front and back object to align with them.

![Flowchart of parking algorithm]

**Figure 3.** Flowchart of parking algorithm

### 4. Results and Discussion

The robot movement is tested using a sample environment using boxes as obstacles. The boxes were placed in the environment in order to substitute the other cars existence. At first, the robot started to move from the right side of the first box. The robot initialized the position and detected the distance in the right side using the ultrasonic sensor (Fig. 4 (a)). Thus, the robot started to move straight forward and scanned the distance (Fig. 4 (b)). If the length and width were considered as appropriate to fit the robot (Fig. 4 (c)), the robot will start the parking maneuver.

The algorithm was working well during the testing. However, during the test, the sensor's output was not correct with the real distance. It might be caused by noise from the environment. In future works, the data output needs to filter and corrected. Another problem was the robot hardly to move straight because the rubber tires were not installed neatly. To solve that problem, the servo degree was initially corrected based on the chassis and tires angle.
Figure 4. Movement result of the robot

5. Conclusion
A prototype of an autonomous wheeled car is implemented and well-integrated with ultrasonic sensors to accomplish the parallel parking task. The experimental results confirmed that our system can solve the parallel parking problem even though the movement of the robot still rigid. However, this system is still far from completion. It needs a better algorithm to filter the noise from the sensors, such as the fuzzy filter. Furthermore, the ability to sense the unknown environment need to be increased by adding several sensors, such as camera, GPS, and accelerometer.

References
[1] Borenstein J, Everett HR, Feng L and Wehe D 1997. Mobile robot positioning-sensors and techniques. Naval Command Control And Ocean Surveillance Center Rdt And E Div San Diego Ca.
[2] Li Y 2017 Internet Things Cloud Comput. 5 52.
[3] Sebi SA and Sunny D 2016 Procedia Technol. 25 364.
[4] Mutlu, L and Uyar E 2012 IFAC Proc. 45 355.
[5] Priandana, K and Buono A 2017 Hexapod leg coordination using simple geometrical tripod-gait and inverse kinematics approach, Advanced Computer Science and Information Systems (ICACISIS), International Conference on IEEE p 35.
[6] Wahab MFB, Moe AL, Abu AB, Yaacob ZB and Legowo A 2015 *ARPN J. Eng. Appl. Sci.* **10** 7107.

[7] Maung SYM, Aye YY and Win NN 2017 *Am. Sci. Res. J. Eng. Technol. Sci. ASRJETS* **38** 23.