

# Autonomous Parallel Parking of a Car-Like Mobile Robot with Geometric Path Planning

Soe Yu Maung Maung<sup>a\*</sup>, Yin Yin Aye<sup>b</sup>, Nu Nu Win<sup>c</sup>

<sup>a</sup>*Ph.D Candidate, Department of Mechatronics Engineering, Mandalay technological University, Myanmar*

<sup>b</sup>*Lecturer, Department of Mechatronics Engineering, Mandalay technological University, Myanmar*

<sup>c</sup>*Professor, Department of Mechatronics Engineering, Mandalay technological University, Myanmar*

<sup>a</sup>*Email: soeyumaungmaung@gmail.com*

<sup>b</sup>*Email: yinyinrose7@gmail.com*

<sup>c</sup>*Email: malnuwin@gmail.com*

## Abstract

With the advancement of technology making everything so convenient in these days and ages, autonomous system is very interesting area in the innovating technology. One of the advanced booming technologies for the improvement of human race is autonomous taxi mobile transportation system. It is working well in a district area but it still thrives on making more comfortable. This research will be one point of supporting roles for automobile in parallel parking. An autonomous parallel parking of a car-like mobile robot has been developed in this study. The ultrasonic range sensors are used to detect the working environments and design an s-shaped trajectory between two parallel parked vehicles. One trail maneuver system is used to be moving along the s-shaped trajectory parking path. The s-shaped trajectory is purely based on the geometric approach path planning method. The proposed method is not dependent on the initial pose of the robot but it must be parallel with the parking space. Sensor data are used as the main decision part to change the parking states instance of to find the parking space and adjust robot orientation. Fuzzy filter is applied to stabilize the sensor data and give a quick response input. The working environment is constrained by the wall and sensor arrangement. Visual studio 2013 is used for the user interface window. The effectiveness of the proposed method is demonstrated through some experimental results with a car-like mobile robot.

**Keywords:** Parallel parking; Geometric path planning; Car-like mobile robot; S-shaped trajectory; One trail.

---

\* Corresponding author.

## 1. Introduction

Parking control system has recently attracted considerable attention in the control community. The biggest problem people face while driving is not actually driving the car, but parking it. Many researchers have tried to find the optimized solution in different ways to alleviate the driver's parking burden. There are many types of parking method: bay parking (90 degree), slide parking (45 or 65 degree), and parallel parking (180 degree). Among them, parallel parking is the most difficult parking method even experience driver may cause collision. Parallel parking means on a road where cars are parked in front of each other, bumper to bumper, length wise and there exists a gap, the car can be parked between it, in parallel [1]. A rule based parallel parking system for a prototype mobile robot has been developed in this research. This kind of system is designed to improve comfort and security of driving in constrained environments where a lot of attentions and experiences are needed to maneuver the vehicle. Since the task of parallel parking requires many repeated maneuvers in a very limited space, its nature is different from the normal driving in a huge environment. There are two different approaches to solve the parking problem. The first one, the trajectory planning approach, needs a mapping and auto-localization algorithms. In addition to this, it generates paths by using the dynamic and kinematics models of the vehicle and vehicle constraints. Then, it controls the vehicle velocity and the steering angle to ensure a position in the mapped environment [2-6]. On the other hand, the second approach tries to mimic the skills of an experienced driver by using Artificial Intelligence techniques such as finite automaton, fuzzy logic (e.g. Zhao and Collins Jr [7]) or artificial neural networks (e.g. Jenkins and Yuhua [8]). These methods do not need any path planning and their associated tracking control but they are limited to the experience of human experts and not easily extended to more general cases [9]. This research deals with the first approach. In real life, when the parking space is large enough, human driver usually steers the front wheel to maximum angle for retrieving. When the vehicle is retrieved, the driver steers the front wheel to maximum angle again but in reverse direction until the car is parallel to the road. The problem of retrieving the vehicle from the parking slot has been solved by Lo and his colleagues [10]. Since this procedure is reversible, it can be applied in parallel parking problem. This steering scenario forms a simple path composed by two identical minimum circles connected by a tangent point. In order to use this path for parallel parking, it is necessary to know the correct position where to begin the parking maneuvers in accordance with the initial position of the car. Geometric approach path planning method will help to solve this problem. In this approach, parking path will be changed in accordance with the initial position of the car; the only constraint is the car must be parallel with parking space.

### 1.1. System overview

The goal is essentially to park the mobile robot between two parallel posture cars for narrow parking space in an obstacle free environment. The operation block diagram of the parking procedure is shown in Figure.1. Firstly, the robot will move straight forward by calibrating with the wall. In this state, the robot will be working with calibration system. Simultaneously, the robot will be searching the parking space. If the robot detects the parking space, odometry system is used to measure the length of parking slot. Then, the robot will decide there have enough parking space or not by comparing with actual detecting length of the parking space and minimum theoretical length of the parking space. The minimum theoretical length of the parking space can get from the idea of retrieving the vehicle from the parking slot that can change according to the mobile robot geometry.

Sensors cannot measure the exact depth of the space if there is no wall and generally parking is done in a line with neighbouring vehicles. Therefore, it can be assumed in the real world that the depth of the space is equal to the width of the vehicle. If parking spaces are enough, the robot will take the lateral displacement value with the front car and insert in geometric equation. Then, the robot will know where the starting point is to move along the s-shaped trajectory parking path. After that, the car will move from the initial position to the starting position. Then, the robot will move backward along the two tangential arcs with the maximum right steering angle for the first arc and maximum left steering angle for the second arc like the s-shaped trajectory parking path. After moving this maneuver, the parking procedure is finished.

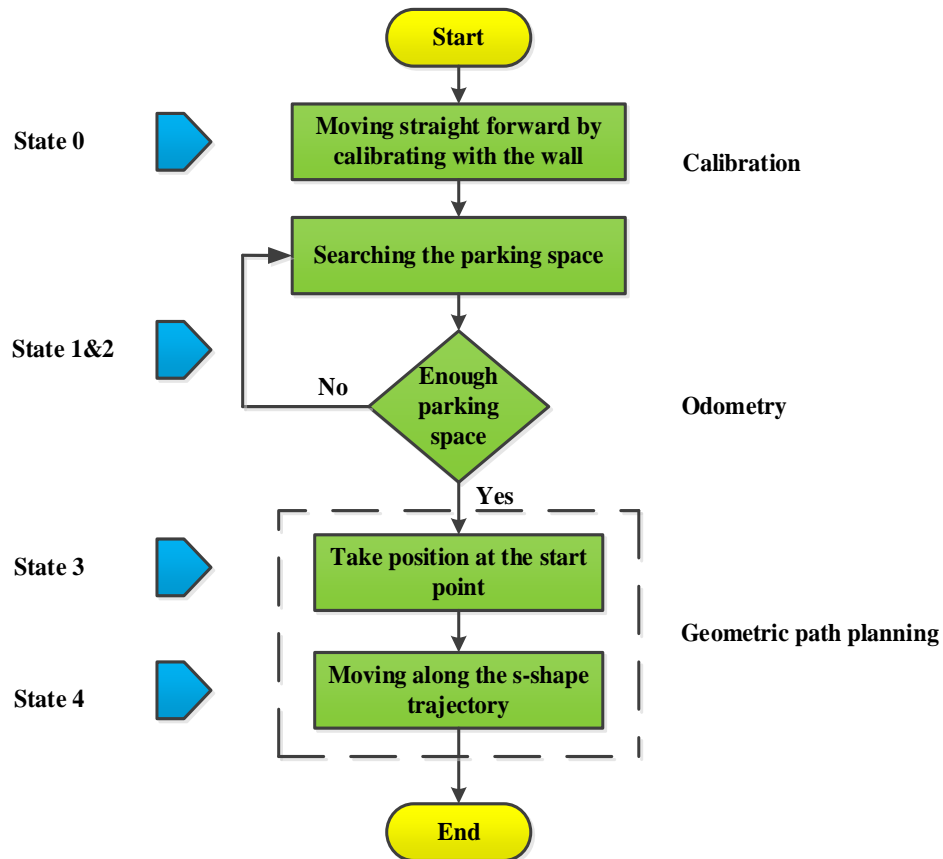
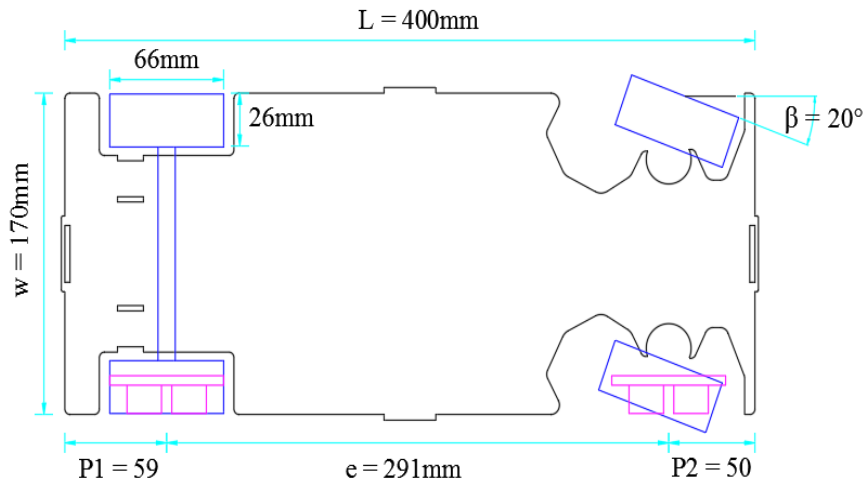


Figure 1: Operation block

## 2. Mobile robot design and sensor arrangement

### 2.1. Mobile robot design

In this research, mobile robot model is like a car which has four wheels. Two wheels in the rear are connected with one shaft and DC motor is used to drive this shaft. Optical encoder is fixed with the shaft to know and control the robot movement. Two steer wheels are connected like a link with a servo motor. Steering angle will become according to the servo motor rotation. Robot chassis is made with Formica to be light the robot's body. The robot's frame is designed with Auto CAD software and cut with the laser cutter to reduce the mechanical damping. The prototype model of the car used in this research is 1:10 scale of MARK II. The design of mobile robot and its parameters are shown in Figure.2.

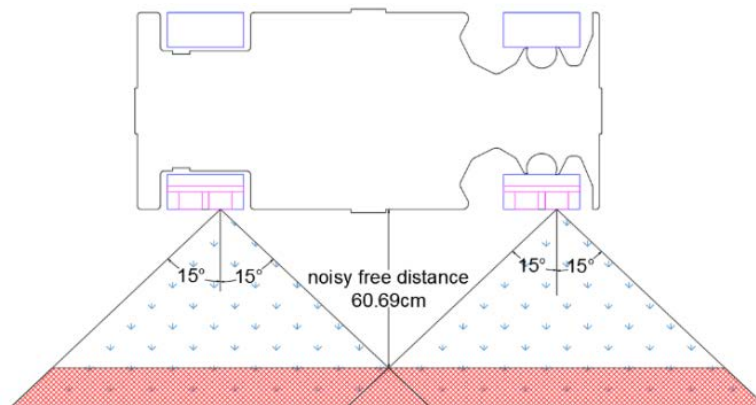


**Figure 2:** Prototype design and parameter

### 2.2. Sensor arrangement

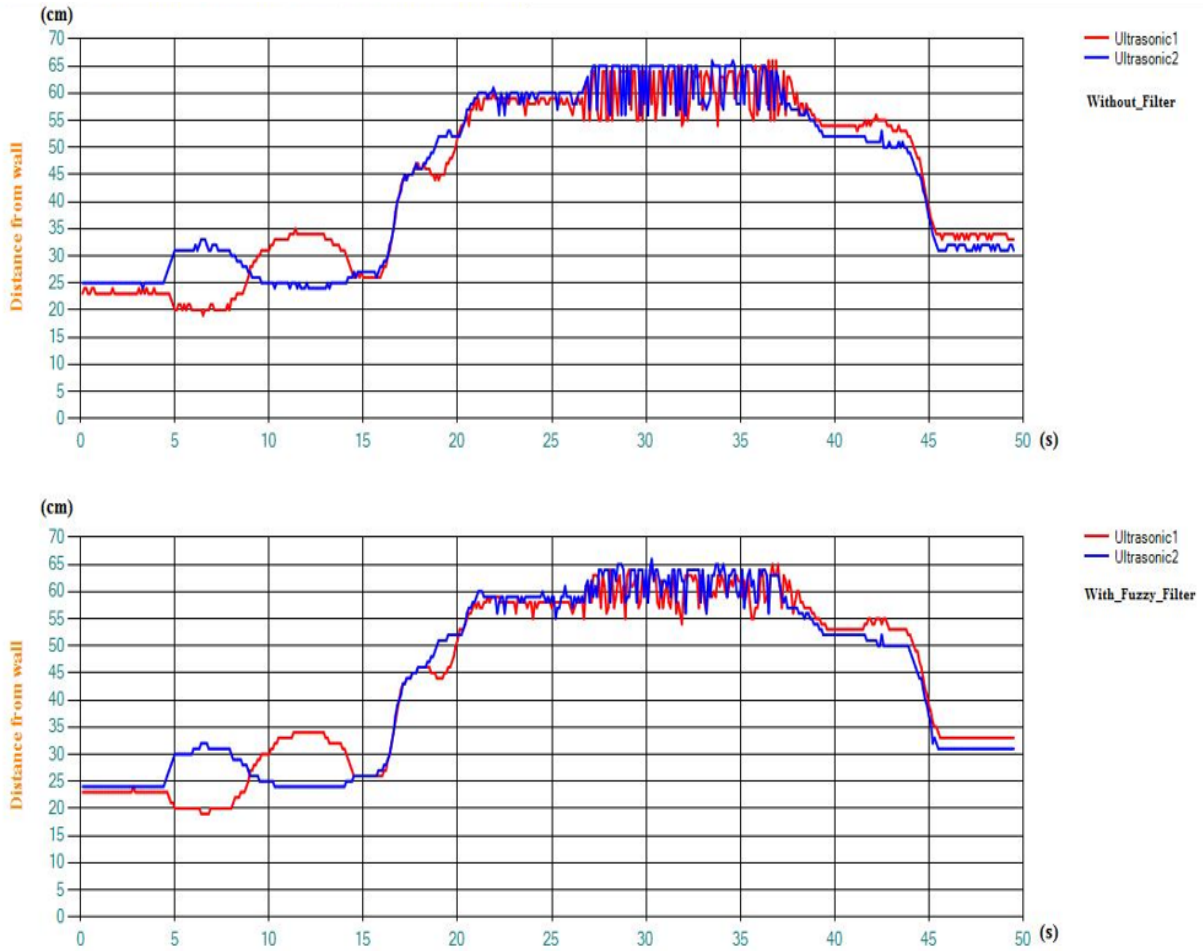
The two HC-SR04 ultrasonic sensors are used in this research. Each one is fixed at the right side of the robot above the rear and front wheels' axle. It transmits sound wave like a V-shape in every  $10\mu s$  and collect reflection wave duration to calculate the measuring distance.

According to the sensors arrangement, there has a constraint on a measuring distance shown in Figure.3. The best data collection angle for sensor is 15 (degree). The two sensor data will be confused in one point because these are fixed parallel in one side. From this constraint, the parking robot will be working well in 60.69 (cm).



**Figure 3:** Sensor arrangement in prototype design and noisy free distance

There still has another problem with ultrasonic sensor. It works well in flat surface. If the detection surface is not flat, the reflection wave can be confused and the correct data cannot get. The other weak point is sensor data that are not stable even in a flat surface. From these facts, ultrasonic sensor is not suitable in a precision case. However, in this research it was used for the facts of cheaper and easy to get. These errors can be corrected by using filter such as average filter and fuzzy filter. Experimental test can validate these facts with fuzzy filter shown in Figure.4.



**Figure 4:** Filtering with fuzzy logic and constraint detecting distance

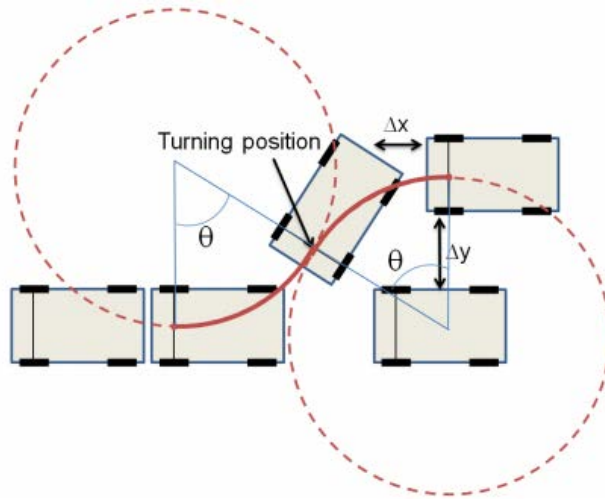
Fuzzy filter is more suitable than the average filter for this research. Average filter makes long programming time based on the number of sample data and it cannot get quick response. Fuzzy filter can get quick response on data and it is shown in equation (1). The sum of coefficient value must be '1'. To be quick response relied on income data, the coefficient of income data should be greater than the coefficient of old data. According to the chart, it can be seen that sharp noise can be stable with filter. Unstable condition is occurred during 27 to 38 seconds because the sensor detecting distance is over the constraint distance.

$$Data\_update = (income\ data * x) + (old\ data * y) \quad (1)$$

### 3. Path planning

#### 3.1. One trail maneuver system

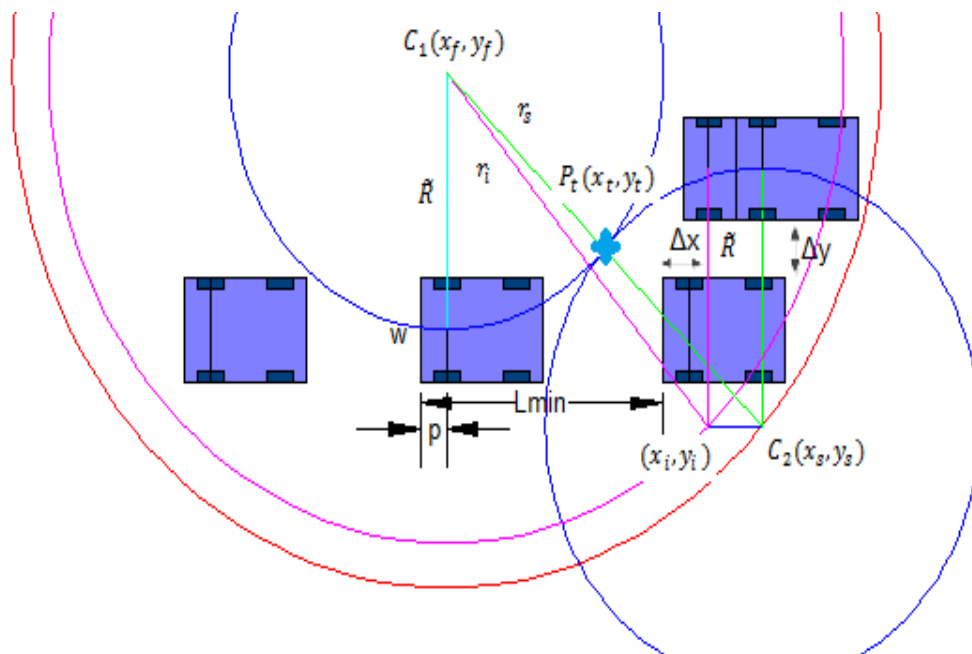
Parallel parking in one trail maneuver is used in this research. It means that the car can be retrieved without reversing the longitudinal velocity then the car can be parked without velocity sign changes too [11-13]. According to this maneuver, tangent point is the ways to change the steering angle from maximum left to maximum right. S-shape trajectory is formed by the combination of two circular arcs connected by a tangent point. The forming of s-shape trajectory in one trail maneuver system is shown in Figure.5.



**Figure 5:** Parallel parking in s-shape trajectory

**3.2. Calculation of the Longitudinal Starting Position and Turning Position in Accordance with the Initial Lateral Position**

When the parking procedure is started, the robot moves in the vicinity of the parking space and takes position parallel with the front car. At the same time, the robot will measure the size of the parking space and lateral displacement  $\Delta y$  offset from the front car. If longitudinal starting position  $x_s$  and turning position  $P_t$  in accordance with  $\Delta y$  can be directly computed, the robot will be successfully parked along the s-shape trajectory. Figure.6 is shown geometrical information diagram for finding longitudinal starting position  $x_s$  and turning point  $P_t$ .



**Figure 6:** Geometrical calculation of initial position to starting position

Firstly, the robot will move longitudinally  $x_i$  to  $x_s$  and then started steer maximum to the right while moving back to the  $P_t$ . Then counter-steer will be turned to the left and the robot will enter into the parking slot. In this section, analytical calculation is developed for these relationships.

In this research, the path planning strategy is come from final position to the starting position of the parking procedure. The final position of the robot is equal to the  $L_{min} - p$ . The ICR of  $C_1(x_f, y_f)$  is always perpendicular to the final position of the parking maneuver with a distance of  $R_i + w/2$  and defined this distance with radius  $\tilde{R}$ . The origin of coordinate (0, 0) is denoted at  $C_1(x_f, y_f)$ . Similarly, the ICR of  $C_2$  is always perpendicular to the starting position of parking maneuver  $(x_s, y_s)$  with a distance of  $\tilde{R}$ . Each radius of these two circles is the same and their arcs are tangentially connected. The circle of center  $C_1$  is always in a fixed position at the end of the maneuver and the circle of center  $C_2$  is moving tangentially connected along the arc of  $C_1$  in accordance with the lateral displacement  $\Delta y$  shown in Figure.6. It seems like center  $C_2$  is moving on the arc of radius  $2 \times \tilde{R}$  with center  $C_1$ . The initial position of the car can be calculated as follow:

$$x_i = L_{min} - p + \Delta x \tag{2}$$

$$y_i = \left[ \tilde{R} + \frac{w}{2} \right] + \left[ \tilde{R} - \left( \Delta y + w + \frac{w}{2} \right) \right] \tag{3}$$

The radius of initial position with the center of  $C_1$  can be calculated by using circular equation:

$$x_i^2 + y_i^2 = r_i^2 \tag{4}$$

The initial position and starting position of lateral displacement are the same:

$$y_i = y_s \tag{5}$$

The radius of starting position with the center  $C_1$  is

$$r_s = 2 \times \tilde{R} \tag{6}$$

Now, the longitudinal starting position  $x_s$  can be calculated by subtracting the circular equation:

$$\begin{array}{r} x_s^2 + y_s^2 = r_s^2 \\ - x_i^2 + y_i^2 = r_i^2 \\ \hline x_s^2 - x_i^2 = r_s^2 - r_i^2 \\ x_s^2 = r_s^2 - r_i^2 + x_i^2 \end{array}$$

$$r_s = 2 \times \tilde{R} \tag{7}$$

Now, the point  $P_t(x_t, y_t)$  can be calculated that is where the vehicle counter-steers.

$$y_t = \frac{y_s}{2} \tag{8}$$

$$x_t = \frac{x_s}{2} \tag{9}$$

The central angle can be calculated from the two circles' tangent point easily.

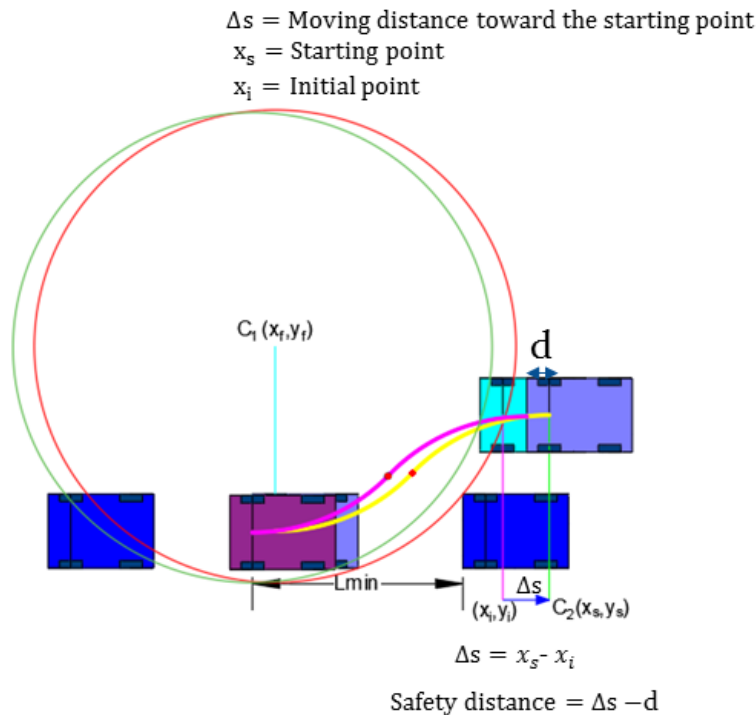
$$\text{Central angle} = \sin^{-1}\left(\frac{x_t}{R}\right) \text{ (Degree)} \tag{10}$$

The arc length is calculated for the robot to be followed the s-shape path.

$$\text{Arc length} = R \times \left(\text{central angle} \times \frac{180}{\pi}\right) \tag{11}$$

### 3.3. Safety path

Significant and practical keys issues in motion planning are safety and optimality. Firstly, predefined collision free path must be considered based on the geometry of the vehicle. The test of the robot maneuver is in an obstacle free indoor environment. Front and rear parallel posture cars are only needed to count for the probability of collision. According to the path planning in section B, it can be seen that the robot will be parked along the yellow colour s-shaped path and the red circle is the front bumper moving path. The front bumper of the robot has high probability of collision with the front car rear bumper shown in Figure.7.  $\Delta s$  is the distance to move from initial point to the starting point and the parking maneuver will start at this point. The changing of starting point will act on the parking path. The parking path can be moved by subtracting some distance  $d$  from the  $\Delta s$ . Finally, the green circle bumper collision free path is got from moving the yellow colour s-shaped path to the purple colour s-shaped path.



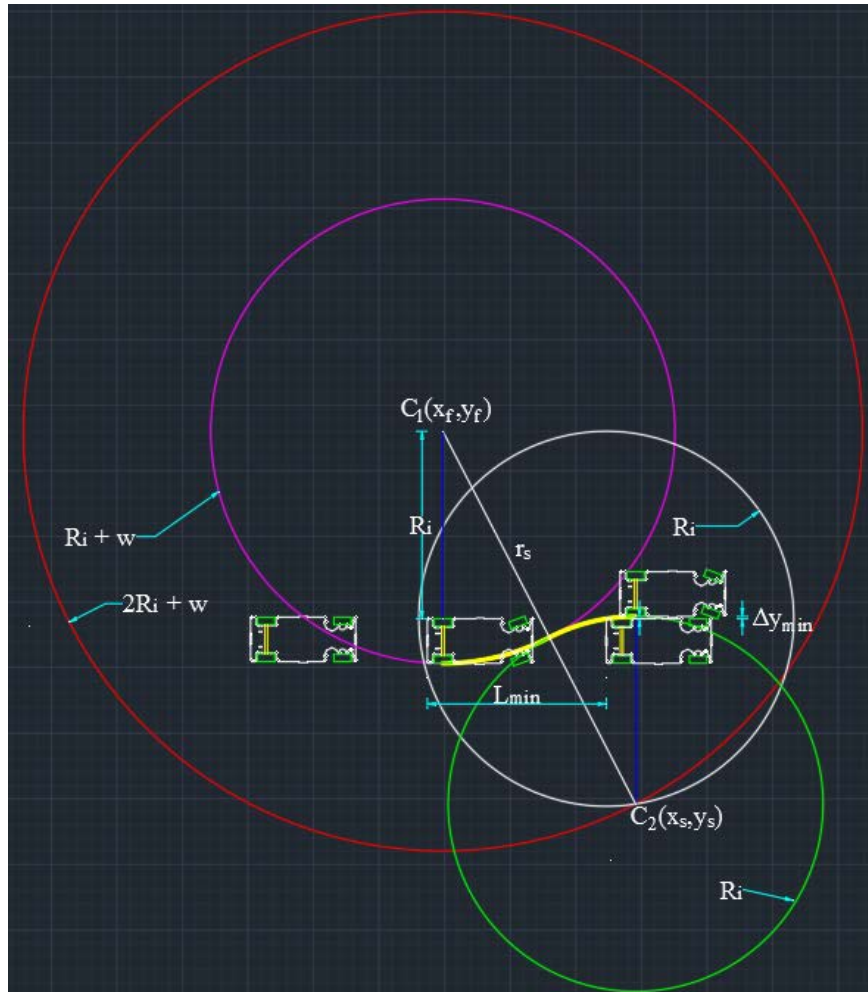
**Figure 7:** Geometrical calculation of initial position to starting position



#### 4. Minimum and maximum lateral distance

Lateral distance means the amount of space between the parking robot and the second car parallel offset. The s-shape trajectory parking path can be changed based on the lateral distance. The larger the lateral distance is, the longer the parking path follows. The geometrical calculations of minimum and maximum lateral distance are shown in Figure 8 and 10.

##### 4.1. Minimum lateral distance



**Figure 8:** Parking with minimum lateral distance

Circular equation;

$$(x - h)^2 + (y - k)^2 = r^2 \quad (12)$$

Center of coordinate  $C_f(0,0)$  is used as origin. The red circle is for the second circle moving center point to be tangentially connected with the violet circle.

$$(x - 0)^2 + (y - 0)^2 = r_s^2 \quad (13)$$

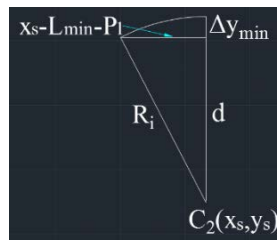
To avoid collision with the second car, the parking robot rear moving path and its circular equation is:

$$h = L_{min} - P_1$$

$$k = R_i$$

$$(x - h)^2 + (y - k)^2 = r_s^2 \quad (14)$$

By solving equations (13) and (14), it can be got two tangent points  $(x_1, y_1)$  and  $(x_2, y_2)$ . The larger x and smaller y as  $(x_s, y_s)$  will be defined as  $C_2$  center point. And then, the minimum lateral distance can be calculated.

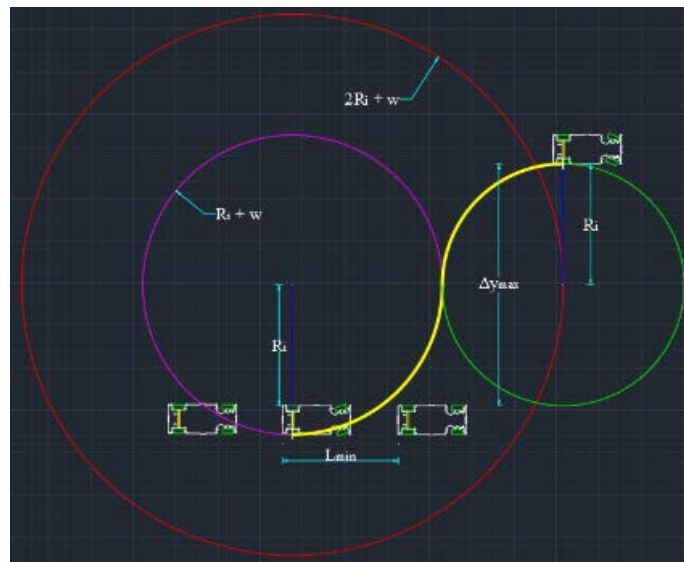


**Figure 9:** Finding the minimum lateral distance

$$d = \sqrt{R_i^2 - (x_s - (L_{min} - P_1))^2} \quad (15)$$

$$\Delta y_{min} = R_i - d \quad (16)$$

#### 4.2. Maximum lateral distance



**Figure 10:** Parking with maximum lateral distance

$$\Delta y_{max} = 2R_i \quad (17)$$

## 5. Working environment and parking algorithm

### 5.1. Working environment

In this research, the parking will be done in a known environment. It is assumed that there has a wall that is right side of the parking robot and there have a gap between the robot and the wall. The gap distance is constraint on two ultrasonic sensors working area. It has explained in section III. According to the prototype design and sensor constraint area, the maximum gap length is 60.69 (cm) and the two front and rear parked cars can be located in this range. It was also needed to count minimum detecting length (2cm) of ultrasonic sensor. Therefore, the maximum located length is (60.96cm-2cm) and the minimum located length can be approximately zero. For the parking robot, the minimum offset from the wall must be at least its width when the parked cars are located at the minimum offset. For the parking robot, left side of the two parked car should be approximately marginally parallel. The predefined working environment and the whole parking procedure are shown in Figure. 11.

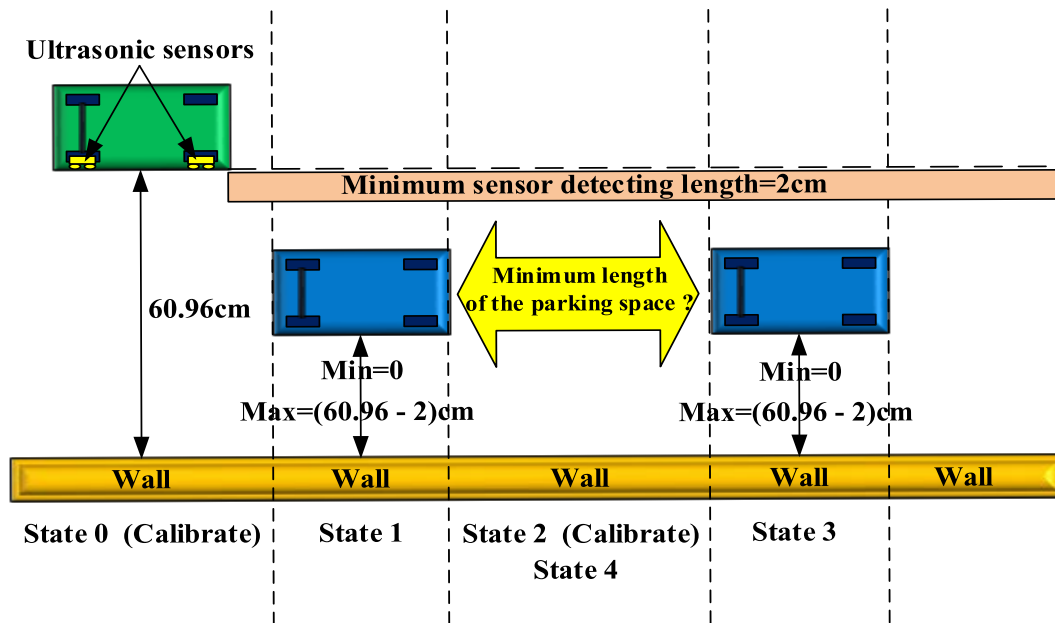


Figure 11: Predefined working environment and parking procedure

### 5.2. Parking algorithm

There are five states in the overall parking procedure. The changing of each state depends on the sensor data differences. The two ultrasonic sensors are fixed at the right side of the robot and it was defined front one is  $U_1$  and rear one is  $U_2$ . The detail procedure can be well known from the program flow chart shown in Figure. 12.

$$U_{12} \text{ difference} = U_1 \text{ update} - U_2 \text{ update};$$

$$U_1 \text{ difference} = U_1 \text{ update} - U_1 \text{ old};$$

$$U_2 \text{ difference} = U_2 \text{ update} - U_2 \text{ old};$$

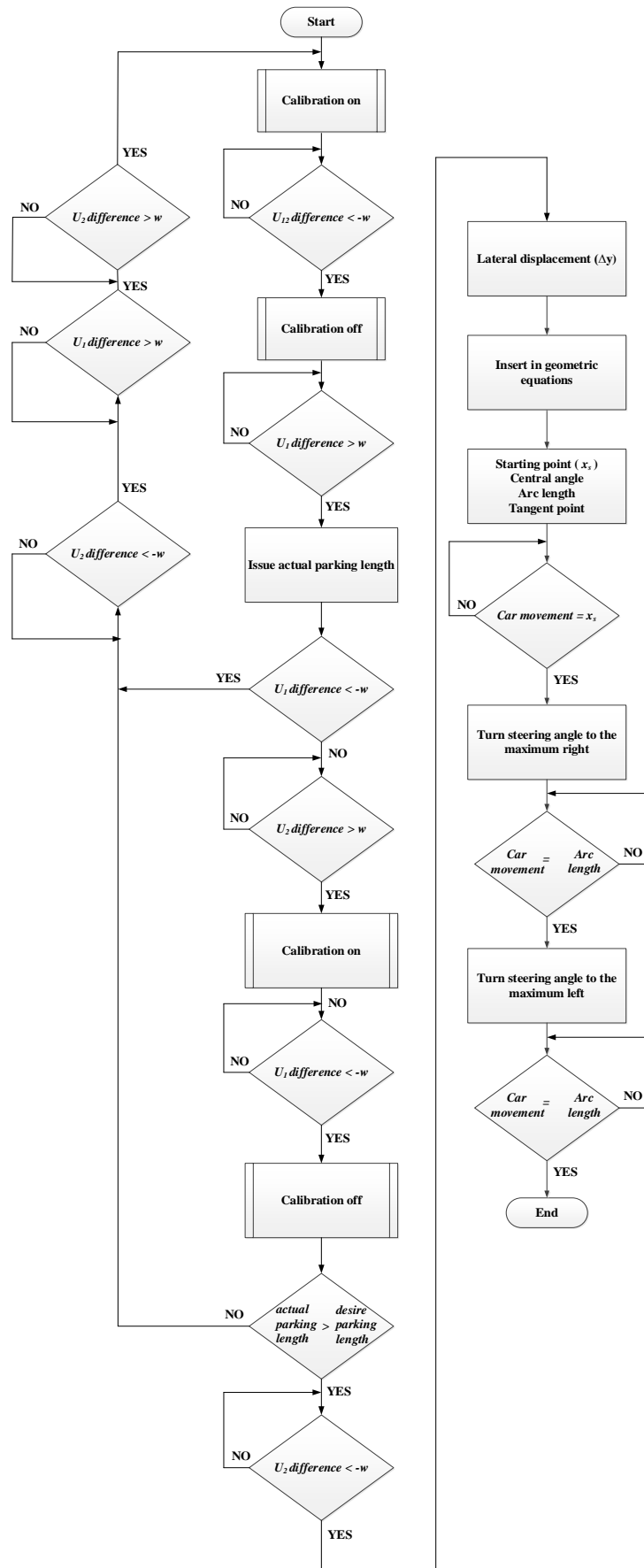


Figure 12: Program flow chart

State 0: Firstly, the robot will be moving straight forward along the wall relying on the sensors difference calibration. The steering angle will adjust based on the front and rear sensor data differences. Calibration system is used in this state. 90 degree is the fixed standard steering angle for the straight ahead and the coefficient 1.5 is calculated from the relation of steering angle and sensor difference.

*If ( $U_{12} \text{ difference} > 1$ ); the robot is heading to the left, change steering angle to the right  $\{90 - (1.5 * U_{23} \text{ difference})\}$*

*If ( $U_{12} \text{ difference} < -1$ ); the robot is heading to the right, change steering angle to the left  $\{90 + (1.5 * U_{23} \text{ difference})\}$*

State 1: In this state, the parking robot detects the first car and starts counting the first car length. It is assumed that the robot is moving straight, so calibrate state will be off and robot is moving with fixed steering angle.

*If ( $U_{12} \text{ difference} < -w \ \&\& \ \text{state} == 0$ ) state=1;*

State 2: Robot detects the end of first car and starts counting the parking length. And then, the robot will make decision parking or not. There have also calibration in state 2 based on the parking length. Firstly,  $U_1$  difference will be greater than the parking car width. It was assumed the parking robot is over the first car. If  $U_1$  difference less than again the parking car width before  $U_2$  difference greater than the parking car width, the robot will decide there has no enough parking space and go to state 0. If the rear sensor  $U_2$  is over the first car and the front sensor  $U_1$  is still in the parking space at that time calibration will be making. The robot will decide again make parking or not when the front sensor  $U_1$  detect the second car at that time calibration will be off. There has enough parking space or not which can be known by comparing actual detected length of the parking space and minimum length of the parking space. If the actual parking length is greater than the minimum parking length the robot will make parking. If not, the robot will find the next parking space and go to state 0.

*If ( $U_1 \text{ difference} > w \ \&\& \ \text{state} == 1$ ) state=2;*

*If ( $U_1 \text{ difference} > w \ \&\& \ U_2 \text{ difference} > w$ ) Calibrate on;*

*If (actual parking length < minimum parking length) not parking*

*If (actual parking length > minimum parking length) make parking*

State 3: The car will stop when the rear sensor detected the second car. Simultaneously, the lateral displacement value ( $\Delta y$ ) can get from the  $U_2$  update. Starting position, central angle, tangent point and arc length can be calculated from the geometric equations by inserting the lateral displacement value. And then, the robot will move toward the starting position.

*If ( $U_2 \text{ difference} < -w \ \&\& \ \text{state} == 2$ ) state=3;*

State 4: S-shape path is formed by two identical circle tangentially connected and it can be divided into two arc

length. Firstly, the robot will move backward along the s-shape path with maximum right steering angle until the robot reaches the tangent point. After that, the robot will turn the maximum left steer again and move along the second arc. Then the parking will be finished.

## 6. Interfacing system

Visual studio 2013 is used for the user interface window. The purpose of using interface is to test the parking algorithm and to know the operating sequence in detail. The prototype model can also be controlled by interface window. There have four interface windows in user interface and all the windows are connected with a cascade system.

Car parameter window can be seen in Figure.13. In this window, it can be defined as steering angle, safety region and the parameters of prototype model. Different model can also be saved in this window for the purpose of easy usage. Connect button is used to interface with the prototype model and interface window. After the parameters has defined, the confirm button is used to change the next window.

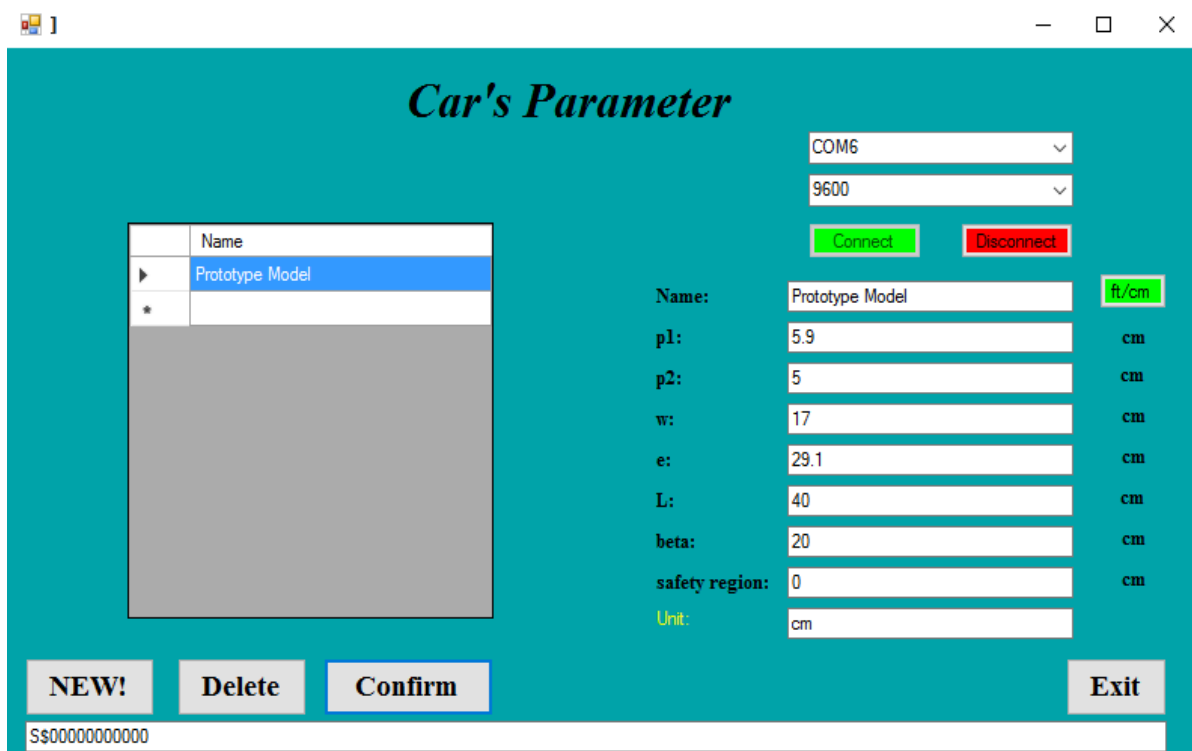


Figure 13: Car parameter window

The second window is main panel window shown in Figure.14. According to the operation sequences, each state and its description are described at the top of the window text box. Front and rear sensors data are described at the right side of the model picture. The other important data are also described in this window. The start button is used to start the operation. Real time graph window and reference path data window can be changed from this window.



Figure 14: Main panel window

In real time graph window, all the operating data are drawing with real time shown in Figure.15. The changing of steering angle, car movement, state and distance can be well known from this window. All the charts are working at the same time and linking each other.

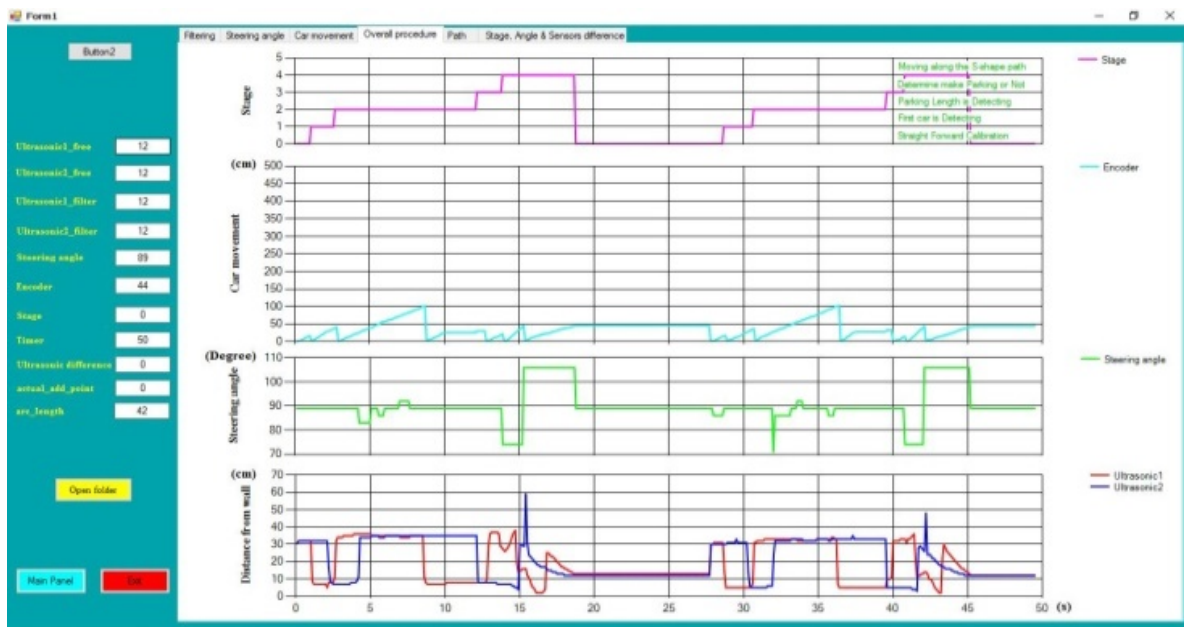


Figure 15: Real time graph window

Localization and coordinate data are described in this path data window shown in Figure.16. The data in each text box is coming from the geometric path planning equations. The blank text boxes can calculate data after knowing the lateral distance value.

Minimum Parking Length				Lateral Distance				Initial Coordinate Points		Starting Distance		Starting Coordinate Points	
68.1989900303272								$X_i$	$Y_i$			$X_s$	$Y_s$
$R_0$	$R_l$	$R$	$R_e$	Tangent Coordinate Point				Unit		Safety Distance		$r_i$	
88,0827	71,4515	79,9515	94,79712	$X_t$	$Y_t$	cm		0					

Arc length:  Central Angle:

Main panel Exit

Figure 16: Path data window

### 7. Experimental result

In this section, experimental result of parking procedure is presented step by step shown in Figure. 17. Firstly, the robot is finding the parking space by using ultrasonic sensors as shown in Figure.17(a). If the length of the parking space is not large enough, the robot will not be parked.

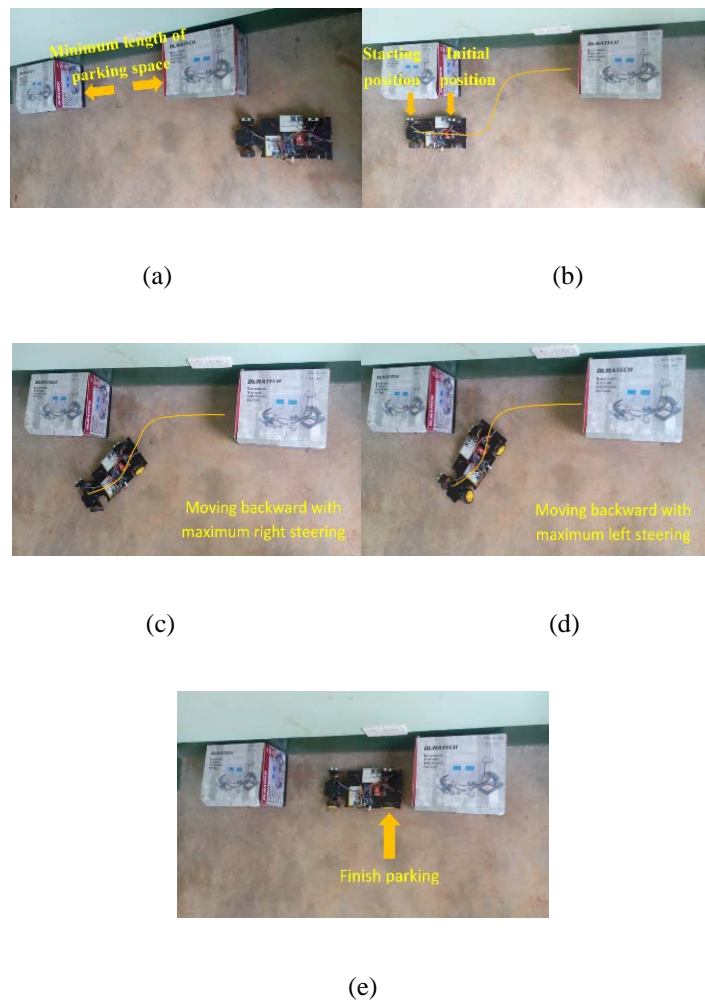


Figure 17: Experimental test for overall parking procedure

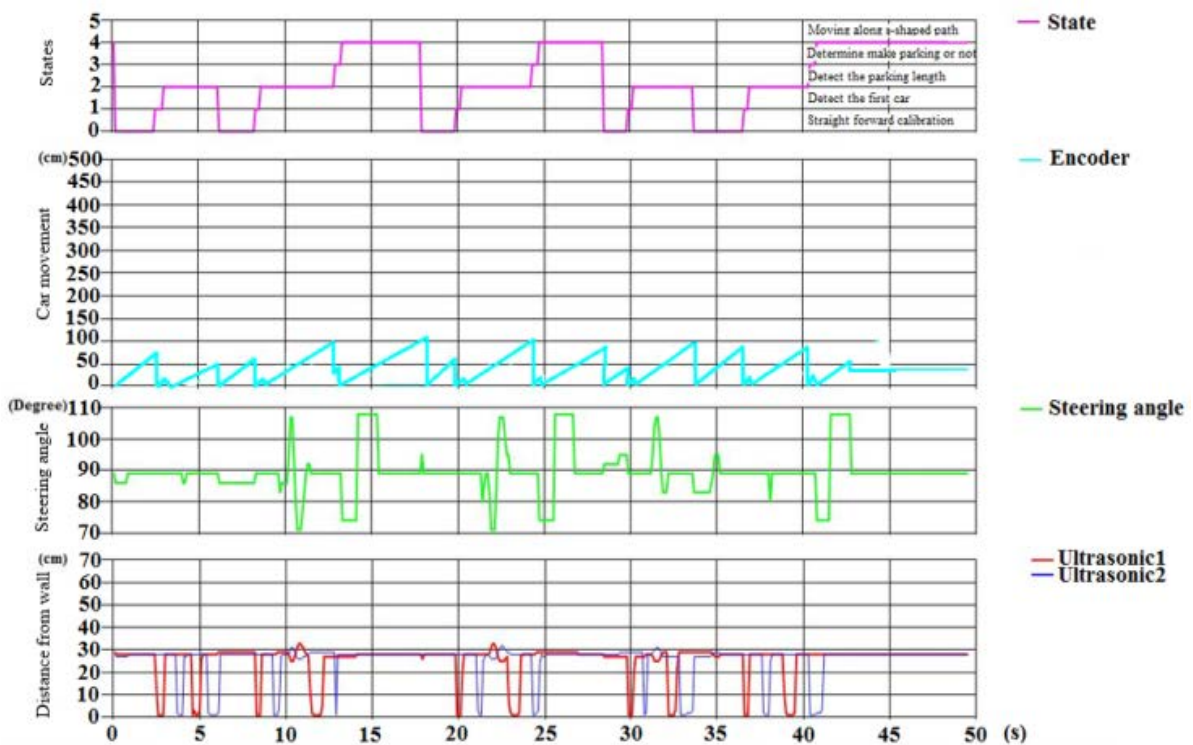


When the robot gets the minimum parking space, the robot will move from initial parallel position to the starting position according to the lateral distance. In Figure.17(b), the robot stops at the initial position.

Figure.17(c) shows the robot which is entering to the parking slot with maximum right steering. Firstly, the servo motor rotates to become maximum right steering and then DC motor will rotate the rear shaft with constant speed until the counter steering point.

In Figure.17(d), when the robot reaches at the counter steering point, the servo motor will rotate again to become maximum left steering and the robot will move back to the final point. While the steering angle is changing, the linear velocity will be zero.

The robot stops at the end of the second parking length as shown in Figure.17(e). After that, the robot will be parallel with the two parked cars.



**Figure 18:** Testing of overall parking procedure

The experimental result is validated with the Visual Basic Simulink shown in Figure.18. According to the overall parking chart, it can be seen that the parking is done three times in 50s. The function of each chart is operated at the same time and related to each other. It can be known that doing parking is obviously seen from steering angle chart and state chart. In steer angle chart, the rectangular wave means the robot is moving along the s-shape path. The value of ultrasonic 1 and 2 differences is used as the input for the changing of parking state. The steering angle will also be variable with this difference in calibration state. Robot movement is decided whether the parking space is enough or not in state 2. If the parking spaces are not enough, the robot will be finding the next parking space.

## **8. Conclusion**

An autonomous parallel parking of a car-like mobile robot based on geometric approach path planning method has been presented in this paper. The experimental results confirmed our system is effective and feasible for parallel parking problem. The s-shaped trajectory based on the geometric path planning was designed by using the sensor measurement of the known working environment. The noisy sensor data were reduced and the quick response inputs were given by using the fuzzy filter. According to the result, our approach was able to park in a narrow parking space without collision.

## **9. Recommendations**

In this research, parallel parking problem has been solved well with the prototype model but there have constraints and limitations. Steering angle can be variable to maximum 20 degree according to the mechanical design of the mobile robot for the parking. The working environment should be obstacle free indoor environment. The maximum offset between the wall and the robot is 60.69cm based on the sensor arrangement. The initial pose of the robot must be parallel with parking space and the position of parking environment has to parallel parking pattern.

## **Acknowledgements**

First of all, to his parents, a million thanks to them for the non-stop encouragements and supports, for being there not only during pleasant times but also during tough times. The author is thankful to all teachers in Department of Mechatronic Engineering, Mandalay Technological University for their effective guidance, helpful suggestions and supervision for this paper.

## **References**

- [1] Igor E. Paromtchik, Christian Laugier, "Autonomous Parallel Parking of a Nonholonomic Vehicle", in Proceedings of the Intelligent Vehicles Symposium, pp.13-18, 1996.
- [2] Kleber de Oliveira Andrade, André Carmona Hernandes and Marcelo Becker "A rule-based controller simulation for an autonomous parallel parking of a car-like robot using laser sensors", Mobile Robotics Lab –Center for Robotics of São Carlos, Brazil, 2012.
- [3] Demirli, K., Khoshnejad, M. "Autonomous parallel parking of a car-like mobile robot by a neuro-fuzzy sensor based controller, Fuzzy Sets and Systems", Vol.160, n.19, pp.2876-2891, 2009.
- [4] Dudek, G., Jenkin, M. "Computational principles of mobile robotics". Cambridge University Press, New York, NY, USA, 2000.
- [5] Jiang, K., Seneviratne, L.D. "A sensor guided autonomous parking system for nonholonomic mobile robot" Proc. of IEEE Int. Conf. Robotics & Automation, Vol. 1, pp. 311-316, 1999.

- [6] Laugier, C., Fraichard, T. "Sensor-based control architecture for a car-like vehicle," *Intelligent Robots and Systems*, Vol. 1, pp. 216-222, 1998.
- [7] Zhao and E. G. Collins Jr. "Robust automatic parallel parking in tight spaces via fuzzy logic Robotics and Autonomous Systems", 51 (2005) 111-127.
- [8] E. Jenkins and H. P. Yuhas. "A Simplified Neural Network Solution Through Problem Decomposition: The Case of the Truck Backer-Upper." *IEEE Transactions on Neural Network*. Vol. 4.,No.4. July 1993
- [9] Laumond, P. E. Jacobs, M. T aix and R. M. Murray "A motion planner for nonholonomic mobile robots" *IEEE Trans. Robotics and Automation*, vol. 10 (5), October, 1994.
- [10] K. Lo, A. B. Rad, C. W. Wong and M. L. Ho "Automatic parallel parking" *IEEE Conf. Intelligent Transportation Systems*, October, 2003.
- [11] Bourke. "Intersection of the two circles"  
<http://local.wasp.uwa.edu.au/~pbourke/geometry/2circle/April, 1997>.
- [12] Sungwoo CHOI, Clément Boussard and Brigitte d'Andréa-Novel "Easy Path Planning and Robust Control for Automatic Parallel Parking" *Preprints of the 18th IFAC World Congress Milano (Italy) September 2, 2011*.
- [13] Lee, M. Kim, Y. Y oum, and W. Chung "Control of a car-like mobile robot for parking problem" *Proc. IEEE Internat. Conf. Robotics and Automation*, 1999.