

7-11-2015

An Improved Approach for Automatic Parallel Parking in Narrow Parking Spaces

Wenyi Zhou
University of Windsor

Follow this and additional works at: <http://scholar.uwindsor.ca/etd>

Recommended Citation

Zhou, Wenyi, "An Improved Approach for Automatic Parallel Parking in Narrow Parking Spaces" (2015). *Electronic Theses and Dissertations*. Paper 5292.

This online database contains the full-text of PhD dissertations and Masters' theses of University of Windsor students from 1954 forward. These documents are made available for personal study and research purposes only, in accordance with the Canadian Copyright Act and the Creative Commons license—CC BY-NC-ND (Attribution, Non-Commercial, No Derivative Works). Under this license, works must always be attributed to the copyright holder (original author), cannot be used for any commercial purposes, and may not be altered. Any other use would require the permission of the copyright holder. Students may inquire about withdrawing their dissertation and/or thesis from this database. For additional inquiries, please contact the repository administrator via email (scholarship@uwindsor.ca) or by telephone at 519-253-3000ext. 3208.

An Improved Approach for Automatic Parallel Parking
in Narrow Parking Spaces

by
Wenyi Zhou

A Thesis

Submitted to the Faculty of Graduate Studies
through the School of Computer Science
in Partial Fulfillment of the Requirements
for
the Degree of Master of Science at the
University of Windsor

Windsor, Ontario, Canada

2015

© 2015 Wenyi Zhou

An Improved Approach for Automatic Parallel Parking
in Narrow Parking Spaces

by

Wenyi Zhou

APPROVED BY:

Dr. H. Wu

Electrical and Computer Engineering

Dr. I. Ahmad

School of Computer Science

Dr. D. Wu, Advisor

School of Computer Science

February 27 2015

Declaration of Originality

I hereby certify that I am the only author of this thesis and no part of this thesis has been published or submitted for publication.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act.

I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis and have included copies of such copyright clearances to my appendix. I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

Abstract

In 2014, there are more than 500,000 parking lot collisions, which is a 4% increase compare to 2010. Many cars are being produced every day, so that the parking spots are designed to be smaller, in the meantime, the size of cars has outgrown the size of parking space since the automakers have been making larger vehicles to favor customers' demands for larger interior spaces.

As a consequence of smaller parking spaces, the possibility of human operational errors is significantly increased, which subsequently leads to accidents and traffic problems.

This paper proposes an automatic parking method for parallel parking, which can be used to park vehicles in a narrower space to reduce the chances of parking lot collisions. It is based on the kinematic model and could be easily combined with other automatic parking approaches such as fuzzy logic and artificial neural network, ultimately making the parallel parking process more effective.

Acknowledgements

I would like to express my gratitude to my supervisor Dr. Dan Wu for his invaluable assistance, patience and guidance. He taught me how to think independently and conducting research professionally. Also, I got the best support from him during my research.

Besides my supervisor, I would like to thank the rest of my thesis committee: Dr. Imran Ahmad and Dr. Huapeng Wu for their engagement and also, their insightful comments.

Dedication

I would like to dedicate this thesis to my parents and my family members who always support and encourage me.

Contents

Declaration of Originality	III
Abstract.....	IV
Acknowledgements	V
Dedication	VI
Chapter 1 Introduction.....	1
1.1 Background.....	1
1.2 Researches on Automatic Parking	4
1.2.1 Skill-based Approach.....	4
1.2.2 Path Planning-based Approach	7
1.3 Contributions.....	10
Chapter 2 Methodology	13
2.1 The Trajectory of Vehicle.....	13
2.2 The Coordinate System.....	15
2.3 The Maximum Steering Angle.....	18
2.4 The Proposed Approach.....	20
Chapter 3 Kinematics Model	26
3.1 Robot Kinematics.....	26
3.1.1 Implement the Forward Kinematics on the Vehicle Model.....	28
3.2 The Intersection Angle.....	30
3.3 The Update of the Center of Circle.....	32
3.4 The Final Equation.....	34
Chapter 4 Implement and Experiment Result	35
4.1 The Simulation using JAVA GUI.....	35
4.1.1 The Simulation Setup.....	35

4.1.2 The Simulation Process and Result.....	37
4.2 The Experiment with Lego EV3	41
4.2.1 The Experiment Setup.....	42
4.2.2 The Calibration	44
4.2.3 The Experiment Process and Result.....	46
Chapter 5 Conclusion	58
5.1 Summary	58
5.2 Future Work	58
Appendix.....	61
References	63
Vita Auctoris.....	66

List of Figures

Figure 1: The normal parallel parking process.	3
Figure 2: The classification of automatic parking.	4
Figure 3: Parallel parking process.	11
Figure 4: Vehicle model [11].	13
Figure 5: Four wheels share the same origin [11].	15
Figure 6: The coordinate system used in the proposed approach.	15
Figure 7: The coordinate system used in proposed approach while the vehicle is moving.	16
Figure 8: Geometrical relationship between the maximum steering angle and the minimum turning radius [13].	19
Figure 9: Parallel parking process.	21
Figure 10: The parameter settings in equation (3).	22
Figure 11: The arc of left turn and the tracking point (x_{ca}, y_{ca})	23
Figure 12: The arc of right turn and the tracking point (x_{cb}, y_{cb})	24
Figure 13: Two-link arm kinematic model [15].	27
Figure 14: Kinematic model implement on the proposed vehicle model.	28
Figure 15: Two-arm Kinematic model [15].	29
Figure 16: The intersection Angle.	31
Figure 17: The symmetric origin of the circle.	33
Figure 18: The dataset of the vehicle for simulation [12].	37
Figure 19: The movements of parking process.	38
Figure 20: The Lego EV3 model.	43
Figure 21: The motors of EV3.	45

Figure 22: The experiment using Lego EV3 with SCS 1.875.	48
Figure 23: The experiment using Lego EV3 with SCS 1.4.	50
Figure 24: The experiment using Lego EV3 with SCS 1.25.	52
Figure 25: The experiment using Lego EV3 with SCS 1.2.	55
Figure 26: The experiment using Lego EV3 with SCS 1.113.	57

List of Table

Table 1: Dataset of the experiment vehicle in paper [12].	9
Table 2: Smallest constraint space in different researches [12].	9
Table 3: The input and output of the simulation program.	37
Table 4: The track of each movement.	39
Table 5: Smallest constraint space in different researches.	39
Table 6: Comparison with previous research result.	40
Table 7: Different parking process from SCS 1.300 to 1.113.	41
Table 8: Smallest constraint space in Lego EV3 data set.	46
Table 9: Result of parking in different Smallest Constraint space.	47

Chapter 1

Introduction

1.1 Background

In 2014, there are more than 500,000 parking lot collisions, which is a 4% increase compare to 2010 [1]. Two major factors are contributed to the increasing accidents in parking lots.

First, with the advancements in development of society today, personal vehicles are an essential part of people's lives, especially in places with underdeveloped public transit systems. As the population grows, the demand of cars grows with it. This demand can be met with higher production of cars. In 2014, there are 5,943,329 cars are produced in USA and this is a 4.7% increase compared to 2013 [2]. However, land does not grow with population, especially for cities' mature and already over-crowded areas. In Japan, citizens have to prove that they have enough spaces to park the vehicle before they purchase the vehicle [3]. A large part of urban land usage is dedicated to vehicle parking, which has become one of the most important outcomes of urban modernization. In order to create more parking spots in highly developed urban areas, the parking spaces are designed to be smaller in these years, and yet, parking spaces are still very limited.

Secondly, not only the number of cars is increased, the size of the cars is significantly increased as well. Over the years, the size of cars has outgrown the size of parking space since the automakers have been making larger vehicles to favor customers' demands for larger interior spaces. However, the size of parking space which is governed by the department of transportation has never changed since 1994 [4]. As a result of this, car owners in UK paid as much as 500 million pounds for repairs caused by scratches and bumps in the parking lots [4].

As a consequence of a smaller parking space, the possibility of human operational errors is significantly increased, which subsequently leads to accidents and traffic problems. Parking issues are also affecting how people feel about driving. Higher stress level can be observed for city drivers.

To address the increasing parking problems and frustrations, automakers are developing and implementing various automatic parking techniques to assist drivers with their parking issues [5]. There are two advantages that automatic parking technologies are trying to address. First of all, automatic parking could deliver better safety [5]. Secondly, automatic parking could park vehicles more accurately than human drivers [5]. Thus, more cars could be parked in the same parking area with less traffic disturbance.

The work presented in this thesis aims to develop and implement an automatic parking approach for parallel parking. Parallel parking is a difficult parking method in the

urban area since sometimes drivers have to park in a tight space. The normal parallel parking process is shown in Figure 1. In this figure, the shadows indicate the walls while the rectangles represent the vehicle. The vehicle firstly drives past the target parking space and stops. Then, it starts to reverse into that space. Finally, makes some adjustments to park the vehicle in the perfect position of the parking spot.

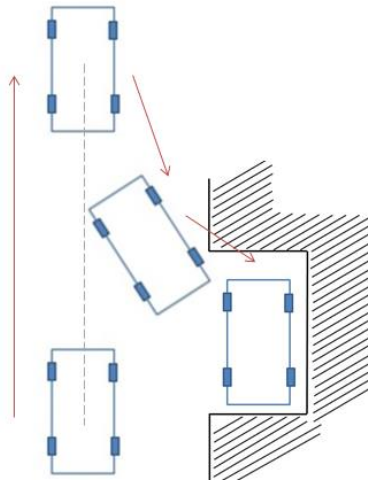


Figure 1: The normal parallel parking process.

However, people sometimes have collisions during this parking process. Especially when the driver is making some adjustments in the parking spot since the vehicle might be too close to the walls during this process. Therefore, the automatic parallel parking will definitely help the drivers make the parking process more effective and safer.

1.2 Researches on Automatic Parking

There is no doubt that automatic parking system is constantly improving over the years [5]. We can see that the automakers nowadays have been implementing the automatic parking technologies on the vehicle to provide a better driving experience for people and certainly, these automatic parking systems are benefited from previous researchers' hard working in this field.

The solutions of automatic parking problem can be generally categorized into two types: skill-based approaches and path planning-based approaches [6]. The classification is shown as below:

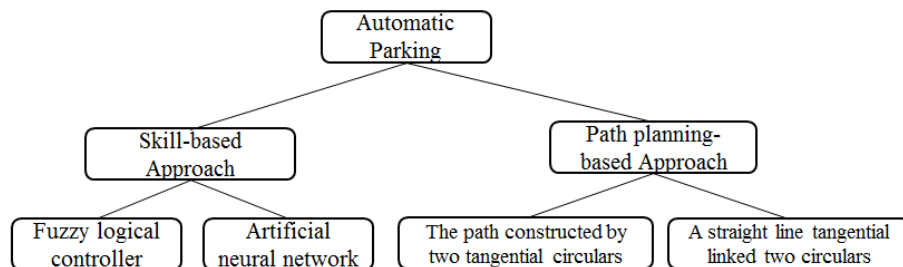


Figure 2: The classification of automatic parking.

1.2.1 Skill-based Approach

There are two main branches in skill-based parking approaches: fuzzy logical controller and artificial neural network [6].

In [7], the authors presented a skill-based approach. The idea is about a three-step maneuvering process: to begin, drive forward so that the vehicle orientation is parallel to

the parking space with the vehicle in a ready-to-reverse position. Then, the vehicle is reversed into the maneuvering space. Lastly, move forward to adjust the vehicle's position inside the parking space. The authors indicated that the fuzzy logic techniques can be applied to each step in the technique outlined.

The authors simulated the parking process with the ATRV-Jr Robot five times and also observed the robustness during the parking process. The analysis conducted indicated that the developed algorithm has the ability to parallel parking the robot into a parking spot that is 1.4 times of the length of the vehicle.

In [6], the authors proposed an algorithm based on a fuzzy logic controller, using the vehicle pose for the input and the steering rate as the output. A vision sensor and ultrasonic sensors are used to localize the vehicle. The algorithm automatically learns an optimal fuzzy if-then rule set from training data. Fuzzy logic controller parameter optimization can be achieved using a genetic fuzzy system.

The experiment of this algorithm points out that the authors have improved the current system so that parking from any position can be done with relative ease.

The third approach is brought from [8]. The presented parking processes are as follows: First of all, the vehicle model is established, and then the parking trajectory is planned into four stages based on the vehicle model: in the first stage, the vehicle adjusts the position to find a suitable parking point. Then, the vehicle starts to turn left and

system processor calculates the tangency point. Next, the vehicle starts to reverse by turning the wheels to the right. In the last stage, the vehicle adjusts to find the target parking point. Meanwhile, the constraints for parking are calculated. Fuzzy control algorithm is used to track the vehicle parking movement.

The authors stated that compared with previous methods, shorter response time is observed.

The last skill-based approach is proposed in [9]. The new method in this paper is described as follows: the segmentation of the parking path is performed to have an entering segment and a back-and-forth shuttling segment. The entering segment is based on the two-steps method, which guide the vehicle into the parking spot. The shuttling segment uses the circular arc of minimum radius of the vehicle to adjust the path until an optimal target position is found.

Multiple types of curves are taken into consideration and multiple solutions are presented. A performance measurement matrix, which includes moving distance, control efforts and path smoothness, etc., is implemented to select the ideal path solution. In the real application setting, the controller presents multiple paths and the driver is asked to select a path as the reference, which will be followed precisely by the tracking controller.

The experiment was implemented successfully and the result shows that a feasible path with any starting position based on driver's select mode can be successfully planned using segmental path planning technique.

1.2.2 Path Planning-based Approach

Compared with the skill-based approach, path planning-based approach is easier to understand since it meets people's driving habits. It is worth emphasizing that the size of the parking space has a great impact on parking complexity, however, this has not been discussed sufficiently in most of the previous research papers. Though some of them have mentioned the smallest space, there are still rooms for improvement.

In [10], the authors proposed a path planning-based algorithm. It plans the parking path by two tangential circles. The process is divided into four steps. In the first step, the vehicle moves to the assigned starting point from the initial position. Then, the vehicle moves to the tangential point of the two circles. And at the third step, the driver changes the direction of the vehicle, which is the opposite direction of the steps above. Finally, some adjustments are taken in order to park the vehicle in the target position of the parking spot.

After the simulation, the authors pointed out that the result shows that the path planning-based method proposed in this paper has achieved not only parking the vehicle into the spot successfully without any collisions, but also alleviating the requirement for

parking space. The selections of the starting point are more flexible and make the whole parking process more convenient.

The authors in [11] proposed two improved path planning-based approaches: geometrically minimum radius path planning-based approach and unequal radius path planning-based approach. Because the path from the starting point to the destination point are two arcs from two circles, so obviously unequal radius path planning-based approach means that two circles have different sizes, it is effective in the situation where the starting point is far away from the parking spot. The minimum radius path planning-based approach means that the two arcs are from two circles of the same size, this is effective in the situation where the parking spot is tighter and the starting point is close to the parking spot.

The authors simulated both approaches and the result shows that the entire parking process can be optimized using the combination of the two algorithms. Also, the scope of the application can be significantly increased.

A new method which improved the approach of fifth-order polynomial is presented in [12]. Penalty function and genetic algorithm are used for calculations. In order to minimize the steering angle at the destination position, with parking boundaries in consideration, an ideal starting position and the fifth order polynomial path curve is calculated and ultimately used as the parallel parking instruction path curve.

The simulation was based on a real Toyota Camry Sedan 2008 with the following dimensions as Table 1 shows and we will use the same dataset for comparison in the simulation chapter.

Name of the argument	Data
Length of the vehicle	4.825m
Width of the vehicle	1.82m
Length from front wheel axle to rare wheel axle	2.755m
The Maximum Steering Angle	45°

Table 1: Dataset of the experiment vehicle in paper [12].

The simulation successfully presented the result of parking path without any collision and the analysis indicated that smaller parking space and shorter parking time can both be achieved with this method.

In the end, the authors stated that comparing with the other four reference papers as Table 2 shows, the requirement of the length of parking spot is the smallest among all five methods. The details of this comparison will be further discussed in the simulation chapter.

Paper	[12]	[10]	[7]	[16]	[17]
Smallest constraint space	1.35	1.382	1.4	1.47	1.765

Table 2: Smallest constraint space in different researches [12].

In [14], a new approach was proposed: the car is equipped with ultrasonic sensors and cameras that gather environment mapping information constantly while driving. The

system will be searching for suitable parking spots with the smallest space requirements in consideration. Once the spot is found, the system generates parking path using the maximum turning angle with collision avoidance taken into consideration. Then the motor system controls the car to follow the path and stops vehicle after driver's confirmation.

The authors pointed out using simulation that the proposed algorithm is able to produce smooth parking path that satisfies different parking requirements.

1.3 Contributions

As we have mentioned before, the normal parallel parking process is taken as what has been shown in Figure 3. In this figure, the shadows indicate the walls while the rectangles represent the vehicle. The numbers on the center of the vehicle represent the position number of the parking process. The drivers start at position 1, then, pass the potential parking spot and stop at the position 2. From position 2 to position 4, the skill-based approaches or path planning-based approach are usually used as we have just discussed.

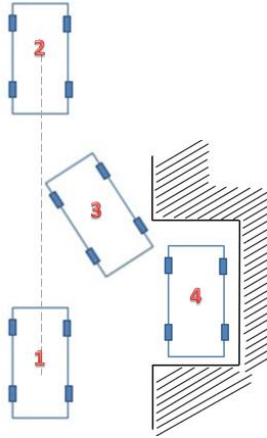


Figure 3: Parallel parking process.

In this paper, we propose a method that improves the parking process from position 3 to position 4. As we mentioned before, people sometimes have the collision during the process from position 3 to position 4 since the vehicle is getting closer to the obstacles during this process. The proposed approach will not only help the driver park safely during this process but also will decrease the required length of the spaces for parking spots. More spaces will be saved so that the cities could be able to meet the increasing demand of the parking space in the urban areas.

The proposed method also could be combined with skill-based approach or path planning-based approach. Most parking process of skill-based approach and path planning-based approach in previous researches are “one-step” parking. They start to adjust their path from position 2 and there is no more adjustment when the vehicle is on position 4, therefore, it requires a large space. In the proposed approach, the skill-based

approach or path planning-based approach is just used from position 2 to position 3, and from position 3 to position 4, the proposed method will be used.

Even though the proposed method is not a “one-step” parking solution, however, it could potentially park vehicles into tight parking spaces which previously proposed methods may fail to do so.

The rest of the paper is structured as follows. The carefully explained idea and the details of the approach will be introduced in Chapter two and three, the simulation using java GUI and Lego EV3 will be presented n Chapter four. The conclusion is in Chapter five.

Chapter 2

Methodology

In this chapter, the details of proposed approach will be presented. Sections 2.1 and 2.3 will introduce some previous research results that will be used in the proposed approach. Section 2.2 and 2.4 will introduce the main idea of the proposed approach.

2.1 The Trajectory of Vehicle

In order to study the automatic parking system, we need to study the trajectory of the vehicles first. A well-constructed vehicle model will allow us to study the trajectory better. In [11], the authors introduced a vehicle model in Figure.4 to help illustrate the movement of the vehicle.

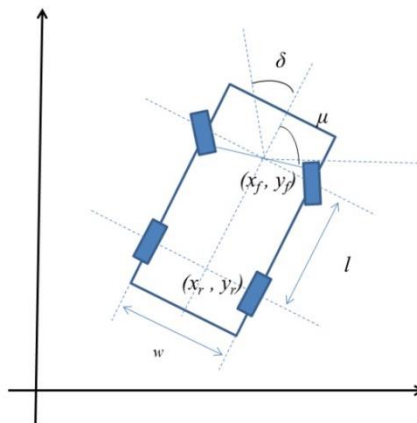


Figure 4: Vehicle model [11].

In this figure, the center of front wheel axle is represented as (x_f, y_f) while the center of rear wheel axle is represented as (x_r, y_r) . And, l represents the length from the front wheel axle to the rear wheel axle, w represents the width of the car, δ represent the steering angle, and the μ represents the orientation of the car.

The following equation is derived in [11] as the trajectory of the center of rear wheel axle:

$$(x_r - a)^2 + (y_r - b)^2 = (l \cdot \cot \delta)^2, \quad (1)$$

where $\begin{cases} a = x_{r0} - l \cdot \cot \delta \cdot \sin \mu_0 \\ b = y_{r0} - l \cdot \cot \delta \cdot \cos \mu_0 \end{cases}$, and (x_{r0}, y_{r0}, μ_0) represents the initial value of (x_r, y_r, μ) . The initial value is the value when the vehicle is at its initial position in the coordinate system.

It was shown in [11] that the trajectory of (x_r, y_r) is a standard circle from equation (1). Similarly, if the vehicle is moving with a certain steering angle, the trajectory of any point on the vehicle could be considered as a finite arc. Moreover, when the vehicle is moving, the trajectory circles of all four wheels form four circles that share the same center as shown in Figure 5.

In Figure 5, we can see the four wheels share the same center of circle. Rz is the length from the center of the rear wheel axle to the center of the circle.

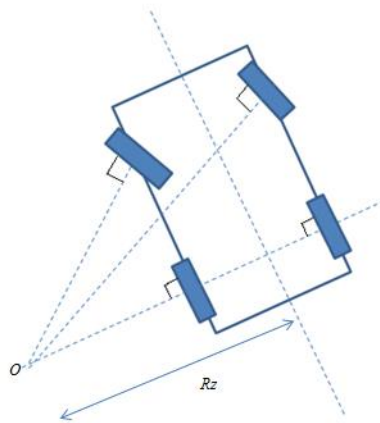


Figure 5: Four wheels share the same origin [11].

2.2 The Coordinate System

After we learned the trajectory of the vehicle, we build our coordinate system for the proposed approach.

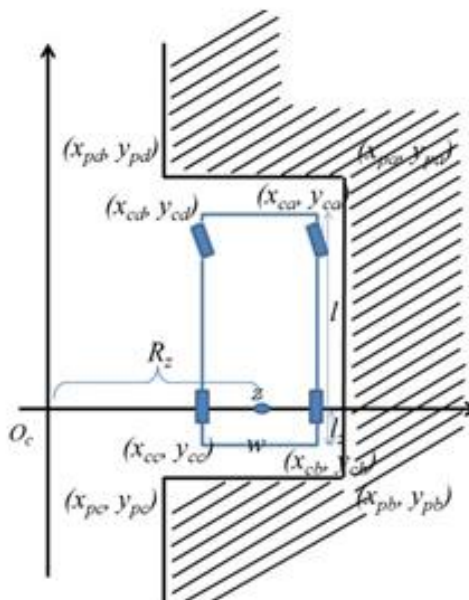
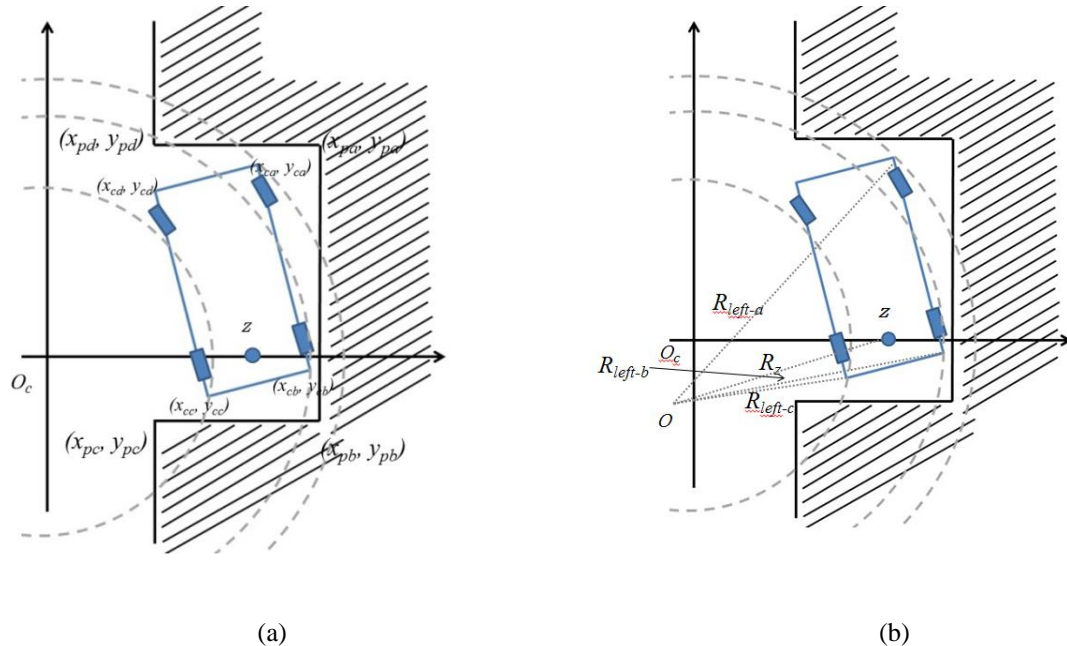


Figure 6: The coordinate system used in the proposed approach.

As Figure 6 shows, the blue rectangle represents the vehicle. The coordinate of the top-right of the vehicle is (x_{ca}, y_{ca}) , and the coordinate of the bottom-right, bottom-right and top-left are respectively (x_{cb}, y_{cb}) , (x_{cc}, y_{cc}) and (x_{cd}, y_{cd}) . Similarly, the coordinate of the top-right of the parking spot is (x_{pa}, y_{pa}) , and the coordinate of the bottom-right, bottom-left and top-left are correspondingly (x_{pb}, y_{pb}) , (x_{pc}, y_{pc}) and (x_{pd}, y_{pd}) . And, z represents the center of rear wheel axle, w represents the width of the vehicle while l represents the length of the vehicle. Also, l_z represents the length from rear wheel axle to the back of the car. R_z is the radius of the circle when the front wheels pose at their maximum steering angles. O_c is the origin of the coordinate, which is never changed during the whole calculation and the distance between O_c and the center of rear wheel axle z is R_z . Additionally, the line connecting O_c and z is parallel to the line connecting (x_{pb}, y_{pb}) and (x_{pc}, y_{pc}) .



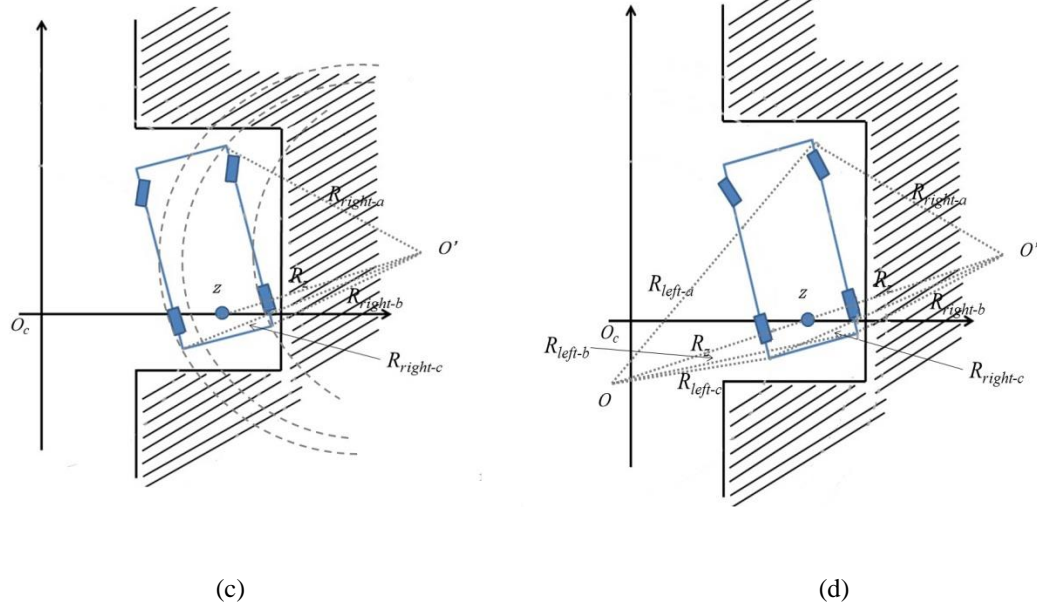


Figure 7: The coordinate system used in proposed approach while the vehicle is moving.

We further explain the use of this coordinate system in four different diagrams in Figure. 7. While the vehicle is moving, the (x_{ca}, y_{ca}) , (x_{cb}, y_{cb}) , (x_{cc}, y_{cc}) and (x_{cd}, y_{cd}) still represent the four corner points of the vehicle while the (x_{pa}, y_{pa}) , (x_{pb}, y_{pb}) , (x_{pc}, y_{pc}) and (x_{pd}, y_{pd}) represent the four corner points of the parking spot as shown in Figure 7 (a), and O_c still represents the origin of coordinate.

When the vehicle is moving, we call the circular trajectory of the forward moving vehicle as moving-forward circle as the dotted arc shows in Figure7 (b) and similarly call the circular trajectory of the backward moving vehicle as moving-backward circle as the dotted arc shows in Figure7 (c). In Figure7 (b), R_{left-a} , R_{left-b} and R_{left-c} represent the radius of the trajectory of the top-right, bottom-right and the bottom-left. Similarly in

Figure 7 (c), $R_{right-a}$, $R_{right-b}$ and $R_{right-c}$ represent the radius of the trajectory of the top-right, bottom-right and the bottom-left of the vehicle.

As we can see in Figure 7 (d), it shows the relationship between O and O' . O is the center of the moving-forward circle and O' is the symmetric point of the O , which is also the center of the moving-backward circle. Moreover, the knowledge from the path planning-based approaches is also used here [11]. As Figure 7 (d) shows, the center of rear wheel axle z is the tangential point to the two trajectory circle connect the moving-forward circle and moving-backward circle. So O and O' are symmetric points and z is their center point.

2.3 The Maximum Steering Angle

The maximum steering angle is an important parameter that will be used in the proposed approach.

The maximum steering angle θ_{steer} is shown in Figure 8. As we mentioned before, the trajectory circles of all four wheels form four circles that share the same center of the circle. And the angle between the line connecting this center to the center of the front wheel axle and the line connecting this center to the center of the rear wheel axle, is the maximum steering angle θ_{steer} . Once the data of the vehicle is obtained, the maximum steering angle θ_{steer} or the minimum turning radius R_z could be derived. From [13], the

geometrical relationship between the maximum steering angle θ_{steer} and the minimum turning radius R_z is shown in Figure 8.

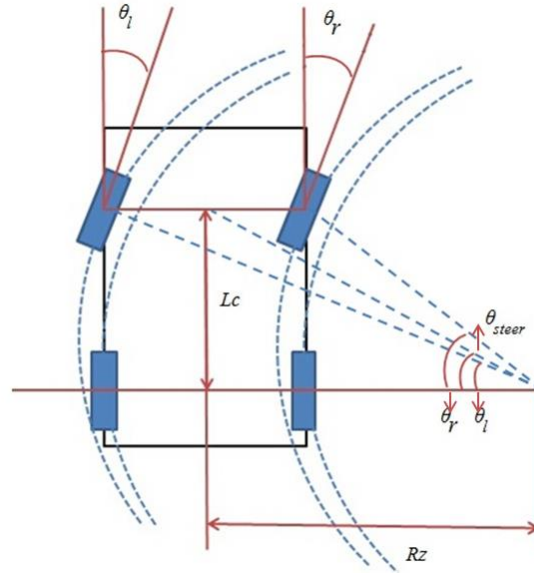


Figure 8: Geometrical relationship between the maximum steering angle and the minimum turning radius [13].

In Figure 8, θ_l and θ_r are the steering angle of the left front wheel and the right front wheel respectively. The θ_{steer} is the maximum steering angle, which can be calculated by using θ_l and θ_r . And also, the relationship between the maximum steering angle θ_{steer} and the minimum turning radius R_z can be found. In [13], the authors show the following equation:

$$\theta_{steer} = \arctan\left(\frac{2 \tan \theta_l \tan \theta_r}{\tan \theta_l + \tan \theta_r}\right),$$

$$R_z = \frac{L_c}{\tan \theta_{steer}}.$$
(2)

In the proposed method, the maximum steering angle will determine the most appropriate turning angle of each movement.

2.4 The Proposed Approach

As we mentioned in Chapter 1 or 2, an approach that improves the parking process from position 3 to position 4 is proposed in this thesis. The parking process in the proposed approach is shown as below:

- 1) The proposed approach firstly requires the length and width information of the vehicle, and also need to know the length and width information of the parking spot by sensor.
- 2) Next, it has to determine the target position of the vehicle in the parking spot, as position 4 in Figure 9.
- 3) Then, the algorithm using the target position (position 4) as the starting point to calculate and determine if it is possible to move out from the parking spot successfully without any collision, and memorize the steps from position 4 to position 3.

- 4) Finally, combine with the skilled-based approach [6-9] or path planning-based approach [10-14] to move the vehicle from position 2 to position 3 and reverse the process from position 4 to position 3.

The key process of the proposed approach is to develop an algorithm that is used for the movement from position 4 to position 3, which will be presented in the following sections.

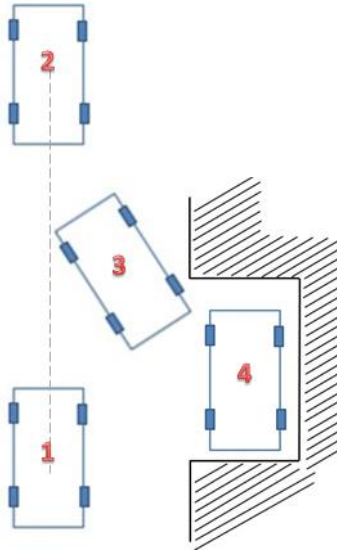


Figure 9: Parallel parking process.

The target position of the vehicle in the parking spot (as position 4 in the figure) has to be determined first. Sensor gets the information of the length of the parking spot, which is denoted as sl and the width of the parking spot, which is denoted as sw .

The coordinator of four corner points of the parking spot is shown as in equation (3) and it also shows the safe distance. All the constant parameters are shown in Figure

10 and are introduced in the previous section. The coordinates (x_{pa}, y_{pa}) , (x_{pb}, y_{pb}) , (x_{pc}, y_{pc}) and (x_{pd}, y_{pd}) are still the parameters of the parking spot. The safe distance is set as 0.1m in the equation in our proposed method. However, it can be changed base on the real situation. , We have equation (3) as below given of the geometric relationship between the vehicle and the parking spot.

$$\begin{cases} x_{pa} = R_z + sw - \frac{w}{2} \\ y_{pa} = sl - 0.1 - l_z \end{cases} \quad \begin{cases} x_{pc} = R_z - \frac{w}{2} \\ y_{pc} = -0.1 - l_z \end{cases} \quad (3)$$

$$\begin{cases} x_{pb} = R_z + sw - \frac{w}{2} \\ y_{pb} = -0.1 - l_z \end{cases} \quad \begin{cases} x_{pd} = R_z - \frac{w}{2} \\ y_{pd} = sl - 0.1 - l_z \end{cases}$$

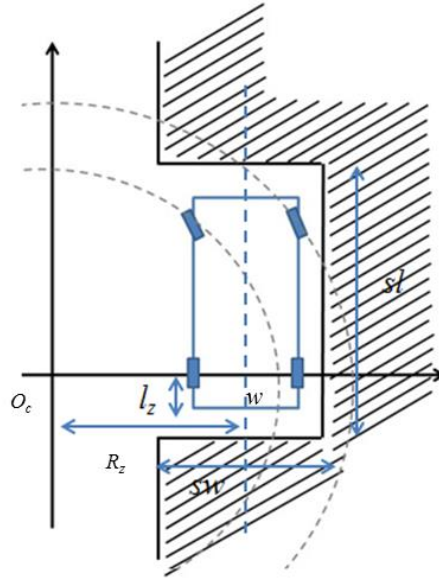


Figure 10: The parameter settings in equation (3).

After the length and width of the parking spot are known as well as the target position and data of the vehicle are known, the algorithm starts to calculate the possible vehicle movement for parking from position 4 to position 3. The algorithm will be divided into two parts: when the vehicle going forward and when the vehicle going backward. As it has been mentioned in the sections above, the trajectory of vehicle is a standard circle, so the center of the circle is changing while the direction of the vehicle is changing. The center of moving-backward circle O' is the symmetric point of the center of moving-forward circle O , and they are symmetric with the center of rear wheel axle z . The detail of the calculation of O , O' and z will be introduced in chapter three.

When the vehicle is moving forward as Figure 11 shows, we set the top-right point of the vehicle as the tracking point, the coordinate of this point is (x_{ca}, y_{ca}) .

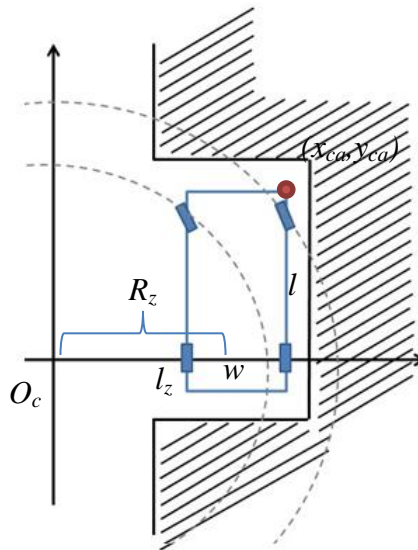


Figure 11: The arc of left turn and the tracking point (x_{ca}, y_{ca}) .

In Figure 11, the vehicle is turning left, which indicates that the top-right point of the vehicle is moving to the left side of Figure 11, x_{ca} is continuously decreasing while the vehicle is moving forward.. In the meantime, the y_{ca} starts to increase its value since the vehicle is moving forward. The geometrical relationship restricts x_{ca} and y_{ca} on the path of the trajectory circle, which has the coordinate O_c as the origin and the R_z as the radius. Also consider avoiding the collision in the parking spot, the corners of the vehicle should not hit the wall or other obstacles, so the value of y_{ca} should be less than y_{pa} . Similarly the value of x_{cb} should be less than x_{pb} , the value of y_{cc} should be less than y_{pc} . All these constraints can be summarized in Equation (4) below. In this equation, x_0 and y_0 are the initial value of x_{ca} and y_{ca} , the initial value is the value when the vehicle is parked at the target position.

$$y_{ca} = \text{Max} \left(y_0 + \sqrt{(l - l_z)^2 + \left(R_z + \frac{w}{2}\right)^2 - (x_{ca} - x_0)^2} \right) \quad (4)$$

$y_{ca} < y_{pa}, x_{cb} < x_{pb}, y_{cc} > y_{pc}$

where $x_{ca} \in [x'_{ca}, x_{pd}]$.

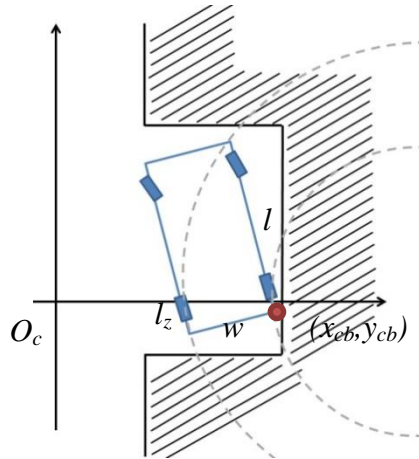


Figure 12: The arc of right turn and the tracking point (x_{cb}, y_{cb}) .

Similarly, when the vehicle is moving backward as Figure 12 shows, we set the bottom-right point of the vehicle as the tracking point, the coordinate of this point is (x_{cb}, y_{cb}) .

The tracking point x_{cb} firstly increases since the vehicle is moving backward, in this situation, the bottom-right point of the vehicle is moving to the right side of Figure 12. In the meantime, y_{cb} starts to decrease its value since the vehicle is moving backward. And the geometrical relationship restricts x_{cb} and y_{cb} on the path of the trajectory circle, R_z is still the radius but the origin is totally different this time. Also considering avoiding collision in the parking spot, the corners of the vehicle should not hit the wall, so the value of y_{ca} should be less than y_{pa} . Similarly, the value of x_{cb} should be less than x_{pb} , the value of y_{cc} should be less than y_{pc} . All these constraints can be summarized in Equation (5) below, in which x_0 and y_0 are the initial value of x_{cb} and y_{cb} , and the initial value is the value when the vehicle is parked at the target position.

$$y_{cb} = \text{Max} \left(y_0 - \sqrt{l_z^2 + \left(R_z - \frac{w}{2}\right)^2 - (x_0 - x_{cb})^2} \right) \quad (5)$$

$y_{ca} < y_{pa}, x_{cb} < x_{pb}, y_{cc} > y_{pc}$

where $x_{cb} \in [x'_{cb}, x_{pb}]$.

Once the coordinate of the (x_{ca}, y_{ca}) and (x_{cb}, y_{cb}) are known, the localization of the other three points of the vehicle are still needed, since the position of the vehicle has to be tracked. This will be introduced in Chapter 3.

Chapter 3

Kinematics Model

In this chapter, a brief introduction of robot kinematics model will be presented. Implementing the robot kinematics on the proposed vehicle model could help us track the vehicle during its movement.

Also, by using the geometric solution of robot forward kinematic, the parking process can be calculated without heavily relying on sensors.

3.1 Robot Kinematics

The main focus of the study of robot kinematics is the motion of bodies but without considering the forces or moments that result from the motion itself [15]. Robot manipulator behaviours require sophisticated robot kinematics modeling in order to be properly analyzed. Thus, formulating the appropriate kinematics models for suitable robot mechanism is very important in studying and analyzing the motion of robot manipulator, which is commonly known as robot kinematics [15].

There are two types of robot kinematics, forward kinematics and inverse kinematics. Inverse kinematic is the reverse process of forward kinematics.

Figure 13 will help us explain these two types of robot kinematics more clearly. In this figure, there is a two-link arm kinematic model. In this figure, l_1 and l_2 represent the links, θ_1 and θ_2 represent the angles of joints.

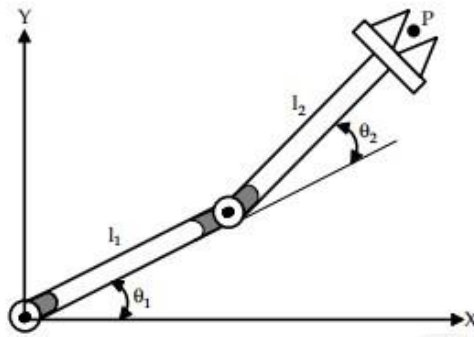


Figure 13. Two-link arm kinematic model [15].

We can explain the two kinds of robotic kinematics briefly below [15]:

- Forward Kinematics is used when the length of each link and the angle of each joint is known, and can be used to calculate the position of any point.
- Inverse Kinematics is used when the length of each link and the position of some points on the robot is known, and can be used to calculate the angles of each joint needed to obtain that position.

In this thesis we will use forward kinematics which is also called direct kinematics.

3.1.1 Implement the Forward Kinematics on the Vehicle Model

In the proposed approach, a 2-dimensional arm's forward kinematic model is used. The using of forward kinematic model allows us to be able to calculate the coordinate of the corner points of the vehicle by using some links and the angles of joints.

As we can see in Figure 14, we implement the forward kinematic model on our vehicle model.

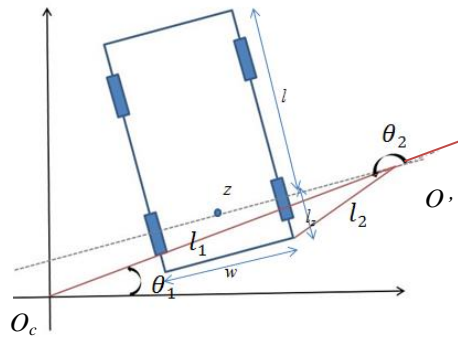


Figure 14: Kinematic model implement on the proposed vehicle model.

We set the line that connecting the origin O_c of the coordinate system and the center of moving-backward circle, which is O' , as the first link l_1 in the kinematic model. And, respectively, the second link l_2 is the line connecting O' , the center of moving-backward circle, and the bottom-right point of the vehicle. The second link may also be top-right of the vehicle since it depends on whether the vehicle is moving forward or moving backward as we mentioned in chapter two. Angle θ_1 is the angle between the l_1 and the x- axis while the θ_2 is the angle between l_1 's extended line and l_2 in anticlockwise.

From [15], the authors presented the figure 15, where l_1 , l_2 are the links, θ_1 and θ_2 are the angles of the joints, and p is a point that could be calculated by using forward kinematic.

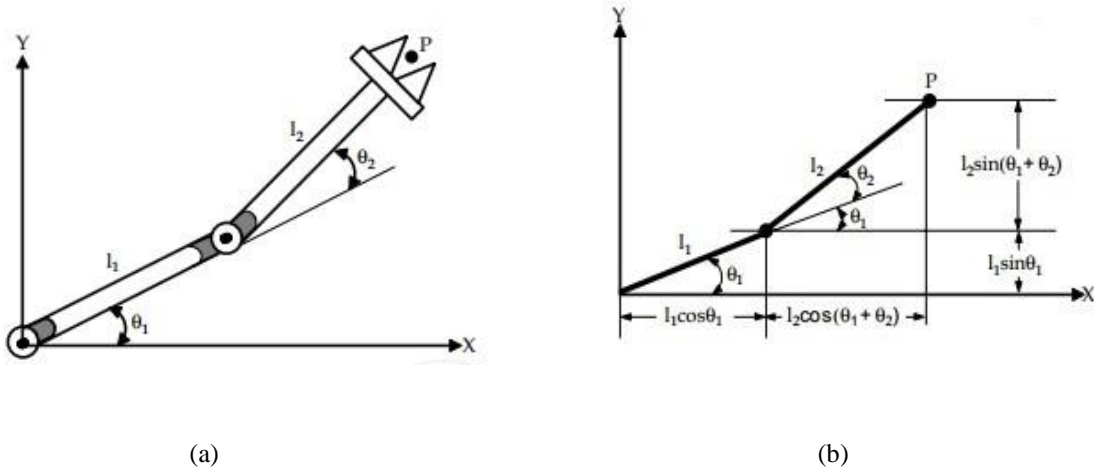


Figure 15: Two-arm Kinematic model [15].

Figure 15 (a) presents a manipulator, and its spatial geometry can be broken down into smaller geometry problems as shown in Figure 15 (b).

As we can easily see in Figure 15 (b), we have the following equation (6):

$$P_x = l_1 \cdot \cos \theta_1 + l_2 \cdot \cos \theta_{12} \quad (6)$$

$$P_y = l_1 \cdot \sin \theta_1 + l_2 \cdot \sin \theta_{12},$$

where $\cos \theta_{12} = \cos \theta_1 \cdot \cos \theta_2 - \sin \theta_1 \cdot \sin \theta_2$, and $\sin \theta_{12} = \sin \theta_1 \cdot \cos \theta_2 + \cos \theta_1 \cdot \sin \theta_2$, and $\cos \theta_{12}$ denotes $\cos (\theta_1 + \theta_2)$.

In [15], it was demonstrated that θ_2 can be derived as below:

$$\theta_2 = \text{Arc tan} \left(\pm \frac{\sqrt{1 - (p_x^2 + p_y^2 - l_1^2 - \frac{l_2^2}{2l_1l_2})}}{p_x^2 + p_y^2 - l_1^2 - \frac{l_2^2}{2l_1l_2}} \right) \quad (7)$$

However, there are two possible values for θ_2 and it is not easy to rule out any one of them since the difference of the two results are small, it has to use some other restrictions to find the right one, which make the calculation process even more complex. Therefore, we introduce the notion of intersection angle to solve this problem.

3.2 The Intersection Angle

The intersection angle is the angle between the line connecting O' , the center of the moving-backward circle and the tracking point, and the line connecting O' and any corner points on the vehicle. The intersection angle helps us to know the angle of the joint instead of finding the right θ_2 in two possible θ_2 from equation (7).

Here we are using (x_{cc}, y_{cc}) as an example. We are looking for the coordinate of the bottom-left point of the vehicle which is (x_{cc}, y_{cc}) . The intersection angle δ is the angle that between the line connecting O' and the point (x_{cb}, y_{cb}) and the line connecting O' and (x_{cc}, y_{cc}) . θ_2 is the angle between the l_1 's extended line and the l_2 in anticlockwise while θ_3 is the angle between the l_1 's extended line and the l_3 in anticlockwise. l_3 is the link connect the O' and (x_{cc}, y_{cc}) . In this figure, l is still the length from the head of the vehicle to the rear wheel axle while the l_z is the length from the rear wheel axle to the end of the vehicle. W is the width of the vehicle and z is still the center of the rear wheel axle.

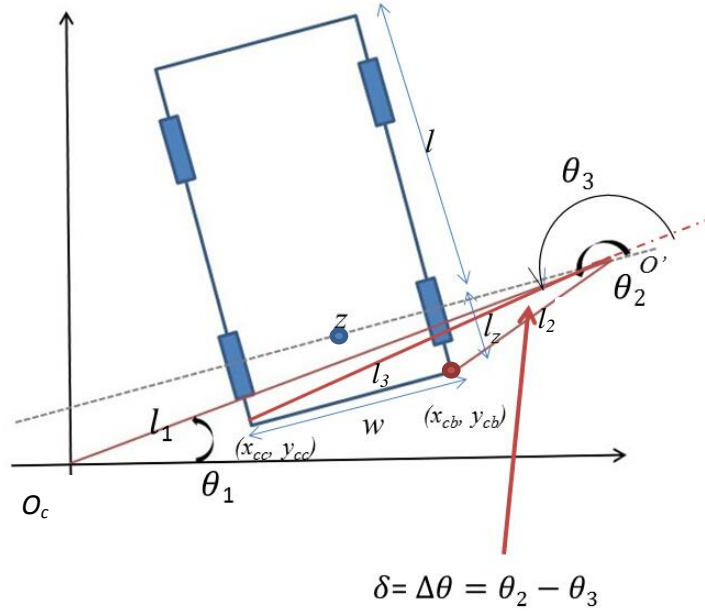


Figure 16: The intersection Angle.

Here we use (x_{cb}, y_{cb}) as the tracking point, so the coordinate of it is known as we introduced in Chapter 2. Then, because l_1 , l_2 and $\cos \theta_1$ are also known, by using equation (6), the $\cos \theta_{12}$ could be deduced. And for (x_{cc}, y_{cc}) , the angle of the joint is the difference between θ_{12} and δ as shown in Figure 16.

Therefore, we have the following equation $\cos \theta_{12} = \cos \theta_1 \cdot \cos \theta_2 - \sin \theta_1 \cdot \sin \theta_2$ and $\sin \theta_{12} = \sin \theta_1 \cdot \cos \theta_2 + \cos \theta_1 \cdot \sin \theta_2$ are equivalent to the equation (8) by using the intersection angle:

$$\cos \theta_{12\delta} = \cos \theta_{12} \cdot \cos \delta - \sin \theta_{12} \cdot \sin \delta \quad (8)$$

$$\sin \theta_{12\delta} = \sin \theta_{12} \cdot \cos \delta + \cos \theta_{12} \cdot \sin \delta.$$

3.3 The Update of the Center of Circle

As we mentioned in chapter two, since the vehicle is moving, the center of the rear wheel axle z also has been changed. Besides, when the vehicle changes its turning direction, the center of the trajectory circle is changed as well, thus, the update of the center of the trajectory circle is required.

The new z is required to be calculated at first. Since the bottom-right point (x_{ca}, y_{ca}) of the vehicle is known when it is the tracking point, so we use the relationship between this point and z to derive the coordinate of z as explained in Figure 17.

In Figure 17, φ_1 is the angle between l_z and the line connecting (x_{cb}, y_{cb}) to z , φ_2 is the angle between the line connecting top-right corner and bottom-right corner, and x-axis. Since (x_{ca}, y_{ca}) and (x_{cb}, y_{cb}) are known as we discussed in chapter two, the distance between (x_{cb}, y_{cb}) and z are fixed, so we could derive that $\varphi_1 = \arctan \frac{w/2}{l_z}$, $\varphi_2 = \arctan \frac{y_{ca} - y_{cb}}{x_{cb} - x_{ca}}$, also $\varepsilon = \varphi_2 - \varphi_1$.

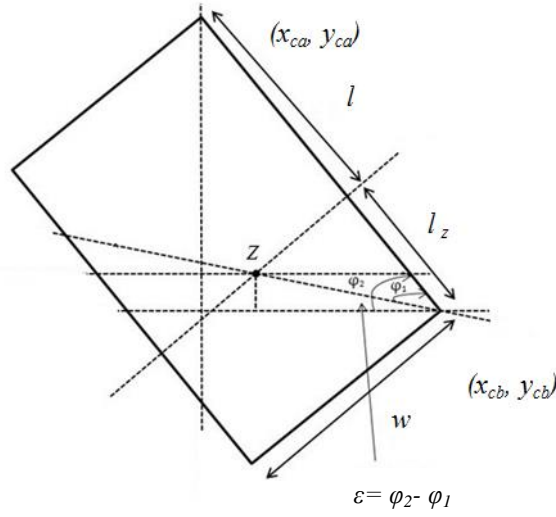


Figure 17: The symmetric origin of the circle.

L_z is still the length from the rear wheel axle to the back of the vehicle while the w is the width of the vehicle. Δx and Δy are the change of the coordinate of z on x-axis and y-axis respectively. Then we have equation (9):

$$\Delta x = \cos \varepsilon \cdot \sqrt{l_z^2 + \left(\frac{w}{2}\right)^2}$$

$$x_z = x_{cb} - \Delta x;$$

$$\Delta y = \sin \varepsilon \cdot \sqrt{l_z^2 + \left(\frac{w}{2}\right)^2}$$

$$y_z = y_{cb} - \Delta y;$$
(9)

After z is updated by equation (9) above, O' , the center of the moving-backward circle is not difficult to be calculated based on the information of O , the center of the moving-forward circle and z .

3.4 The Final Equation

So, combine the equations (4) to (9) that we derived in the previous two chapters,

we have the final equation as equation (10) to equation (12) shown:

$$\left\{ \begin{array}{l} x_{cb} = x_0 + \sqrt{l_z^2 + \left(R_z + \frac{w}{2}\right)^2} \cdot \left\{ \frac{x_{ca} - x_0}{\sqrt{(l-l_z)^2 + \left(R_z + \frac{w}{2}\right)^2}} \cdot \cos \left[\arccos \frac{R_z + \frac{w}{2}}{\sqrt{(l-l_z)^2 + \left(R_z + \frac{w}{2}\right)^2}} + \arccos \frac{R_z + \frac{w}{2}}{\sqrt{l_z^2 + \left(R_z + \frac{w}{2}\right)^2}} \right] \right. \\ \left. + \frac{y_{ca} - y_0}{\sqrt{(l-l_z)^2 + \left(R_z + \frac{w}{2}\right)^2}} \cdot \sin \left[\arccos \frac{R_z + \frac{w}{2}}{\sqrt{(l-l_z)^2 + \left(R_z + \frac{w}{2}\right)^2}} + \arccos \frac{R_z + \frac{w}{2}}{\sqrt{l_z^2 + \left(R_z + \frac{w}{2}\right)^2}} \right] \right\} \\ y_{cb} = y_0 + \sqrt{l_z^2 + \left(R_z + \frac{w}{2}\right)^2} \cdot \left\{ \frac{y_{ca} - y_0}{\sqrt{(l-l_z)^2 + \left(R_z + \frac{w}{2}\right)^2}} \cdot \cos \left[\arccos \frac{R_z + \frac{w}{2}}{\sqrt{(l-l_z)^2 + \left(R_z + \frac{w}{2}\right)^2}} + \arccos \frac{R_z + \frac{w}{2}}{\sqrt{l_z^2 + \left(R_z + \frac{w}{2}\right)^2}} \right] \right. \\ \left. - \frac{x_{ca} - x_0}{\sqrt{(l-l_z)^2 + \left(R_z + \frac{w}{2}\right)^2}} \cdot \sin \left[\arccos \frac{R_z + \frac{w}{2}}{\sqrt{(l-l_z)^2 + \left(R_z + \frac{w}{2}\right)^2}} + \arccos \frac{R_z + \frac{w}{2}}{\sqrt{l_z^2 + \left(R_z + \frac{w}{2}\right)^2}} \right] \right\} \end{array} \right. \quad (10)$$

$$\left\{ \begin{array}{l} x_{cc} = x_0 + \sqrt{l_z^2 + \left(R_z - \frac{w}{2}\right)^2} \cdot \left\{ \frac{x_{ca} - x_0}{\sqrt{(l-l_z)^2 + \left(R_z + \frac{w}{2}\right)^2}} \cdot \cos \left[\arccos \frac{R_z + \frac{w}{2}}{\sqrt{(l-l_z)^2 + \left(R_z + \frac{w}{2}\right)^2}} + \arccos \frac{R_z - \frac{w}{2}}{\sqrt{l_z^2 + \left(R_z - \frac{w}{2}\right)^2}} \right] \right. \\ \left. + \frac{y_{ca} - y_0}{\sqrt{(l-l_z)^2 + \left(R_z + \frac{w}{2}\right)^2}} \cdot \sin \left[\arccos \frac{R_z + \frac{w}{2}}{\sqrt{(l-l_z)^2 + \left(R_z + \frac{w}{2}\right)^2}} + \arccos \frac{R_z - \frac{w}{2}}{\sqrt{l_z^2 + \left(R_z - \frac{w}{2}\right)^2}} \right] \right\} \\ y_{cc} = y_0 + \sqrt{l_z^2 + \left(R_z - \frac{w}{2}\right)^2} \cdot \left\{ \frac{y_{ca} - y_0}{\sqrt{(l-l_z)^2 + \left(R_z + \frac{w}{2}\right)^2}} \cdot \cos \left[\arccos \frac{R_z + \frac{w}{2}}{\sqrt{(l-l_z)^2 + \left(R_z + \frac{w}{2}\right)^2}} + \arccos \frac{R_z - \frac{w}{2}}{\sqrt{l_z^2 + \left(R_z - \frac{w}{2}\right)^2}} \right] \right. \\ \left. - \frac{x_{ca} - x_0}{\sqrt{(l-l_z)^2 + \left(R_z + \frac{w}{2}\right)^2}} \cdot \sin \left[\arccos \frac{R_z + \frac{w}{2}}{\sqrt{(l-l_z)^2 + \left(R_z + \frac{w}{2}\right)^2}} + \arccos \frac{R_z - \frac{w}{2}}{\sqrt{l_z^2 + \left(R_z - \frac{w}{2}\right)^2}} \right] \right\} \end{array} \right. \quad (11)$$

$$\left\{ \begin{array}{l} x_{cd} = x_0 + \sqrt{(l-l_z)^2 + \left(R_z - \frac{w}{2}\right)^2} \cdot \left\{ \frac{x_{ca} - x_0}{\sqrt{(l-l_z)^2 + \left(R_z + \frac{w}{2}\right)^2}} \cdot \cos \left[\arccos \frac{R_z - \frac{w}{2}}{\sqrt{(l-l_z)^2 + \left(R_z - \frac{w}{2}\right)^2}} - \arccos \frac{R_z + \frac{w}{2}}{\sqrt{(l-l_z)^2 + \left(R_z + \frac{w}{2}\right)^2}} \right] \right. \\ \left. - \frac{y_{ca} - y_0}{\sqrt{(l-l_z)^2 + \left(R_z + \frac{w}{2}\right)^2}} \cdot \sin \left[\arccos \frac{R_z - \frac{w}{2}}{\sqrt{(l-l_z)^2 + \left(R_z - \frac{w}{2}\right)^2}} - \arccos \frac{R_z + \frac{w}{2}}{\sqrt{(l-l_z)^2 + \left(R_z + \frac{w}{2}\right)^2}} \right] \right\} \\ y_{cd} = y_0 + \sqrt{(l-l_z)^2 + \left(R_z - \frac{w}{2}\right)^2} \cdot \left\{ \frac{y_{ca} - y_0}{\sqrt{(l-l_z)^2 + \left(R_z + \frac{w}{2}\right)^2}} \cdot \cos \left[\arccos \frac{R_z - \frac{w}{2}}{\sqrt{(l-l_z)^2 + \left(R_z - \frac{w}{2}\right)^2}} - \arccos \frac{R_z + \frac{w}{2}}{\sqrt{(l-l_z)^2 + \left(R_z + \frac{w}{2}\right)^2}} \right] \right. \\ \left. + \frac{x_{ca} - x_0}{\sqrt{(l-l_z)^2 + \left(R_z + \frac{w}{2}\right)^2}} \cdot \sin \left[\arccos \frac{R_z - \frac{w}{2}}{\sqrt{(l-l_z)^2 + \left(R_z - \frac{w}{2}\right)^2}} - \arccos \frac{R_z + \frac{w}{2}}{\sqrt{(l-l_z)^2 + \left(R_z + \frac{w}{2}\right)^2}} \right] \right\} \end{array} \right. \quad (12)$$

Chapter 4

Implement and Experiment Result

In this chapter, the simulation using JAVA GUI and experiment with Lego EV3 will be presented. The details of the simulation are presented in section 4.1, which includes the programming environment, the introduction of dataset, and the results. The details of the experiment with Lego EV3 are presented in section 4.2, which includes the introduction of the Lego EV3 robot, the calibration before the experiment, and the results.

4.1 The Simulation using JAVA GUI

In this section, the details of the simulation using JAVA GUI of the proposed method are discussed. The result shows that the proposed method successfully saves up to 17% of the required spaces of parking spots compared to previous research.

4.1.1 The Simulation Setup

We use JAVA for programming and use JAVA GUI to draw the position of each movement of the vehicle in this experiment. The IDE we use for Java is Eclipse Kepler with the jre 7u55 and jdk 8u5.

The standard of comparisons we use is Smallest Constraint Space (SCS). It is used in many other previous researches; we also use it here so that our result could be comparable with others' results. From [12], SCS is defined as:

$$\text{Smallest Constraint Space (SCS)} = \frac{\text{length of parking spot}}{\text{length of the vehicle}} .$$

As we can see from the definition, the smaller of the SCS value, the less spaces of the parking spot is required.

The data we used in simulation is based on Toyota Camry Sedan 2008 model with the following dimensions: the length is 4.825 m, width is 1.82 m, and the length from the front wheel axle to the rear wheel axle is 2.755 m. The maximum steering angle is 45 degree. Our simulation follows the proportion of the Toyota vehicle and scales proportionally so that the result could fit screen. However, the author in [12] did not mention the safe distance, and in our simulation we used 0.1m as our safe distance here as we have mentioned before in Chapter two, equation (3).

Figure 18 shows a straight forward picture about the vehicle we simulate. It is the same vehicle model from [12] since we will compare it with our proposed approach. In Figure 18, l_1 is the length from the head of the vehicle to the front wheel axle, l is the length from the front wheel axle to the rear wheel axle. Similarly l_2 is the length from the rear wheel axle to the back of the vehicle and L is the total length of the vehicle.

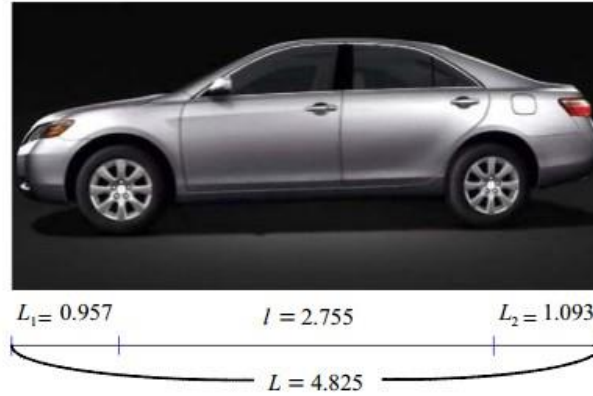


Figure 18: The dataset of the vehicle for simulation [12].

4.1.2 The Simulation Process and Result

Table 3 shows the required input information for simulation and the outputs of the simulation program.

Input Information	Output of the simulation
<ul style="list-style-type: none"> • Length sl and width sw of the parking spot • Length from rear wheel axle to the back of the car l_z • Maximum steering angle θ_{steer} • Length l and width w of the vehicle • Length of the safe distance 	<ul style="list-style-type: none"> • Length of the smallest parking spot • The GUI result • The coordinate of four corner points of the vehicle during the parking process

Table 3: The input and output of the simulation program.

We input the data into the program, the results indicate that the smallest parking spot has a length of 5.3674 meter, so the SCS is $5.3674/4.825$, which is 1.1124.

The result is shown in Figure 19. The red rectangles represent the vehicle while the black lines represent the wall or any objects that the vehicle cannot hit. The number at the top-right corner of the rectangles are counters, we set each red rectangle in the figure represent a ‘movement’ of the vehicle, so as we can see there are 10 movements of the parking process in Figure 19. The counter number 0 means the final stopped position of the vehicle and number 10 is the starting point of the automatic parallel parking.

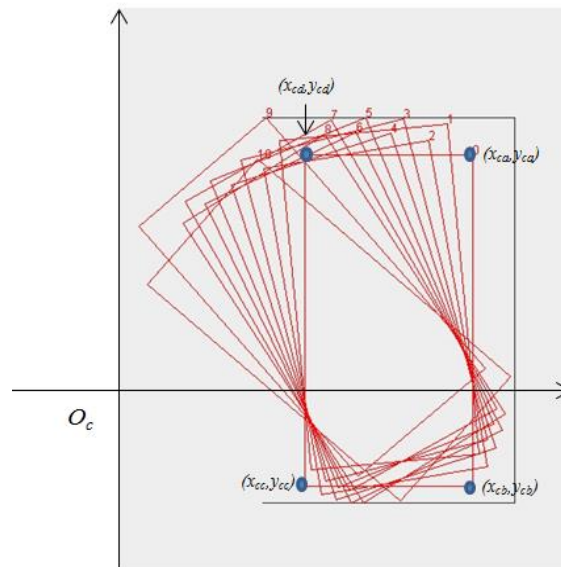


Figure 19: The movements in a parking process.

Table 4 shows the coordinate of four corner points of the vehicle during the parking process. The number on the left column is the number of each movement as shown in Figure 19.

Counter	Coordinate			
0	$(Xca, Yca) = (4, 3)$	$(Xcb, Ycb) = (4, -1)$	$(Xcc, Ycc) = (2, -1)$	$(Xcd, Ycd) = (2, 3)$
1	$(Xca, Yca) = (3.7, 3.363034)$	$(Xcb, Ycb) = (4.076342, -0.61922)$	$(Xcc, Ycc) = (2.085214, -0.80739)$	$(Xcd, Ycd) = (1.708872, 3.174863)$
2	$(Xca, Yca) = (3.484931, 3.173651)$	$(Xcb, Ycb) = (4.176342, -0.76614)$	$(Xcc, Ycc) = (2.206447, -1.11185)$	$(Xcd, Ycd) = (1.515036, 2.827945)$
3	$(Xca, Yca) = (3.184931, 3.433534)$	$(Xcb, Ycb) = (4.186667, 0.439)$	$(Xcc, Ycc) = (2.2504, -0.93987)$	$(Xcd, Ycd) = (1.248664, 2.932666)$
4	$(Xca, Yca) = (3.033046, 3.252048)$	$(Xcb, Ycb) = (4.286667, -0.54643)$	$(Xcc, Ycc) = (2.387428, -1.17324)$	$(Xcd, Ycd) = (1.133807, 2.625237)$
5	$(Xca, Yca) = (2.733046, 3.445531)$	$(Xcb, Ycb) = (4.254496, -0.25382)$	$(Xcc, Ycc) = (2.404822, -1.01454)$	$(Xcd, Ycd) = (0.883371, 2.684806)$
6	$(Xca, Yca) = (2.621526, 3.269091)$	$(Xcb, Ycb) = (4.354496, -0.33602)$	$(Xcc, Ycc) = (2.551942, -1.2025)$	$(Xcd, Ycd) = (0.818971, 2.402606)$
7	$(Xca, Yca) = (2.321526, 3.412512)$	$(Xcb, Ycb) = (4.290286, -0.06944)$	$(Xcc, Ycc) = (2.54931, -1.05382)$	$(Xcd, Ycd) = (0.58055, 2.428132)$
8	$(Xca, Yca) = (2.239861, 3.239804)$	$(Xcb, Ycb) = (4.390286, -0.13298)$	$(Xcc, Ycc) = (2.703894, -1.20819)$	$(Xcd, Ycd) = (0.553469, 2.164591)$
9	$(Xca, Yca) = (1.539861, 3.446147)$	$(Xcb, Ycb) = (4.158345, 0.422318)$	$(Xcc, Ycc) = (2.646431, -0.88692)$	$(Xcd, Ycd) = (0.027946, 2.136905)$
10	$(Xca, Yca) = (1.439367, 2.946085)$	$(Xcb, Ycb) = (4.458345, 0.3221)$	$(Xcc, Ycc) = (3.146307, -1.18748)$	$(Xcd, Ycd) = (0.127329, 1.436596)$

Table 4: The tracking data of each movement during a parking process.

Table 5 shows the Smallest Constraint Space of other researchers' methods [12].

As we can see from Table 5, the best SCS of previous researches is 1.35. The approach that is proposed in [12] used the same Toyota model and it requires 6.51375 meters for the length of parking spot while the proposed approach in this thesis just requires 5.3674 meters.

Paper	[12]	[10]	[7]	[16]	[17]
SCS	1.35	1.382	1.4	1.47	1.765

Table 5: Smallest constraint space in different researches.

Table 6 shows the comparison of the approach from [12] and the proposed approach. As it is shown in Table 6, the proposed approach has improved the possibility to park into a tight parking spot from SCS 1.350 to SCS 1.113. When the SCS is smaller than 1.110, both of the approaches could not park the vehicle in the parking spot. As we

have mentioned before, the proposed method is not a “one-step” method but it does enable to park the vehicle into a narrower space.

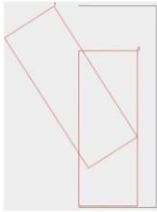



SCS	Length of parking spot (meter)	[12]	Number of steps	Proposed Approach	Number of steps
1.350	6.51375	√	One-Step	√	One Step
1.300	6.27250	×	N/A	√	One Steps
1.280	6.17600	×	N/A	√	One Steps
1.250	6.03125	×	N/A	√	Three Steps
1.200	5.79000	×	N/A	√	Three Steps
1.180	5.69350	×	N/A	√	Four Steps
1.150	5.54875	×	N/A	√	Four Steps
1.113	5.37022	×	N/A	√	Ten Steps
1.110	5.35575	×	N/A	×	N/A

Table 6: Comparison with previous research result.

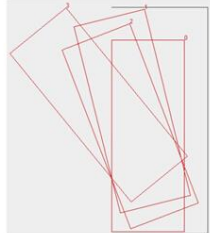
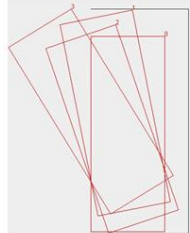

Compared with the new approach that this paper proposed, the proposed method potentially saves up to $\frac{1.350-1.113}{1.350} = 17.6\%$ of the parking space.

Table 7 shows the different parking processes from SCS 1.300 up to 1.113.

As we can see, when the SCS is 1.3, the space of parking spot is spacious as well as when the SCS is 1.28, the vehicle could park into the spot without any stops. Then, the SCS 1.25 and SCS 1.2 require the vehicle makes some movements for park in the spot. As the parking spot is becoming tighter, SCS 1.18 and SCS 1.15 requires more movements to finish the process. Last but not least, as we can see when the SCS is 1.113, the simulation shows the most complicated movements for the parking process.

scs	1.3000	1.2800	1.2500	1.2000
Length of parking spot (meter)	6.27250	6.17600	6.03125	5.79000
Simulation				

(a)

scs	1.1800	1.1500	1.1130
Length of parking spot (meter)	5.69350	5.54875	5.37022
Simulation			

(b)

Table 7: Different parking process from SCS 1.300 to 1.113.

4.2 The Experiment with Lego EV3

In this section, the detail of the experiment with Lego EV3 is presented. Some collisions are observed when the parking spots are very tight, this may due to the

deviations of the hardware and experimental environment. Even so, the experimental results show that the proposed approach appears to work as expected.

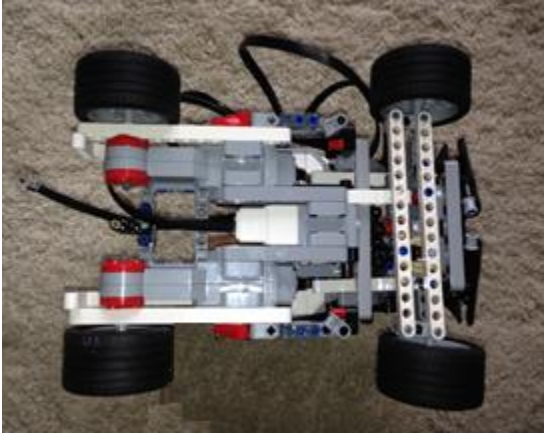
4.2.1 The Experiment Setup

Lego EV3 is the new member of Lego MINDSTORMS series. After the success of its second generation Lego MINDSTORMS NXT 2.0 robot, the Lego company has launched its third generation robot Lego EV3 in MINDSTORMS robotics product line [18].

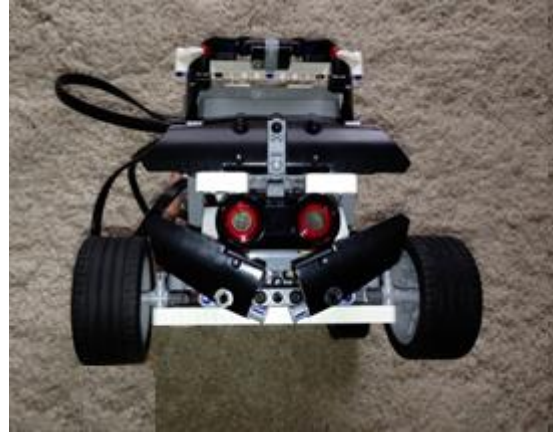
For the experiment, we use the Lego EV3 robot (EV3) as a model of the real vehicle and we use the leJOS as the programming language. leJOS is a firmware replacement for the Lego MINDSTORMS series' programmable bricks. The Java virtual machine inside allows Lego EV3 robots to be programmed in JAVA [19].

The IDE we use for Java is Eclipse Kepler with the jre 7u55 and jdk 8u5, and the leJos firmware is the leJOS_EV3_0.8.1-beta for the EV3.

Figure 20 shows the different sides of the EV3 model vehicle. Figure 20 (a) shows the chassis of the vehicle, Figure 20 (b) and Figure 20 (c) show the front side and back side of the vehicle respectively, and Figure 20 (d) is the right side of the vehicle.



(a) Chassis



(b) Front View



(c) Back View



(d) Side View

Figure 20: The different views of the vehicle assembled from Lego EV3.

This vehicle model is finished by using EV3 original package and the expended package by following the instruction of RAC3 TRUCK on the EV3 official website [20]. This is one of the models that could be used as the real vehicles (4 wheels, rear drive) among all of their instruction guideline online.

The length of the Lego EV3 RAC3 TRUCK is 24cm while the width is 16.5cm. The length from the front of the vehicle to the front wheel axle is 3cm, and the length from the front wheel axle to the back of the vehicle is 4cm. We also set the maximum steering angle as 45 degrees.

4.2.2 The Calibration

Before starting the experiment, the calibration has to be done firstly because there are some deviations either from the wheels or from the odometer. The performance of experiment is affected by the condition of the floor, the component of the vehicle model, the program and other factors.

The first calibration is the turning speed when the EV3 vehicle is moving. Since the EV3 vehicle constructed is a rear wheel drive type vehicle and during the process of changing the direction, the front wheels are changing too fast while the rear wheels, in the meantime, are not stopping moving forward or backward, so there is always a side slip phenomenon. To calibrate this, we add a stop command to the two motors after each movement as shown below in the source code.

Motor.A.stop();//A is the name of one of rear motors

Motor.C.stop();//C is the name of one of rear motors

The other calibration is, after adding the command above, the motor A and motor C cannot stop at the same time, which makes the wheels turning more angles than we expected.

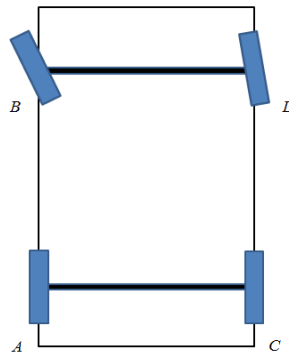


Figure 21: The motors of the EV3.

As we can see in Figure 21, because A and C are the rear wheel motor so if the motor A stops first and motor C keeps moving, or, motor C stops first and motor A keeps moving, the motor B and motor D still posit the corresponding wheels in a turning angle, so the vehicle will still move when it is supposed to stop. In this situation, it will affect the movements in all the following steps and may lead the vehicle to hit the wall or other objects. After some research, the modified version of code is shown as below:

```
Motor.A.stop(true);
```

```
Motor.C.stop();
```

In this two-line code, the motor in the first line will wait a second for the motor in the second line and the final result is that the two motors could stop at the same time.

The finally calibration is the wheels. Both wheels have around a10 degree angle to the left under natural conditions, thus an extra 10 degrees to the right were added and calculated in the program.

4.2.3 The Experiment Process and Result

The program we used here are the same as in the simulation part. But the outputs of the program are the real movements of the EV3 robot instead of the simulated movements in the GUI.

The EV3 does not have the same proportional data of the vehicle as the real Toyota has in last experiment. However, we set up a similar experiment and compare the results using the smallest constraint space. The experiment data is shown in Table 8.

SCS	Length of parking spot (cm)
1.875	45
1.400	33
1.250	30
1.200	28.8
1.113	26.712

Table 8: Smallest constraint space in Lego EV3 data set.

We conducted experiments using the SCS data above. We experiment three times for each SCS. The experiment is considered to be successful if in two out of three experiments, the EV3 does not hit the wall and exit the parking spot successfully without any collisions. Otherwise, it is considered to fail.

After the experiment, the result is shown in Table 9.

SCS	Length of parking spot (cm)	Proposed Approach	Note
1.875	45	√	
1.400	33	√	
1.250	30	√	
1.200	28.8	×	Minor collision
1.113	26.712	×	Minor collision

Table 9: Result of parking in different Smallest Constraint spaces using Lego EV3.

As we can see, though the length of the parking spot is becoming tighter, the proposed approach is still able to finish some parking process. However, we have noticed that during some of the movements, the EV3 slightly hit the wall.

Here are some photo records of the experiments. All of the photos are captured from the successful experiments. In the photos, the red lines represent the wall or any other objects that the vehicles are not allowed to hit. The ruler that on the right side of the photos indicate the length of the parking spot.

The order of the photos is shown as below:

Movement 1	Movement 2
Movement 3	Movement 4
Movement 5	Movement 6
Movement 7	Movement 8
Movement 9	Movement 10
Movement 11	Movement 12

For SCS is 1.875, which means the length of the parking spot is 45cm. The vehicle could be parked in the parking spot in just one movement without any collisions as it is shown below.

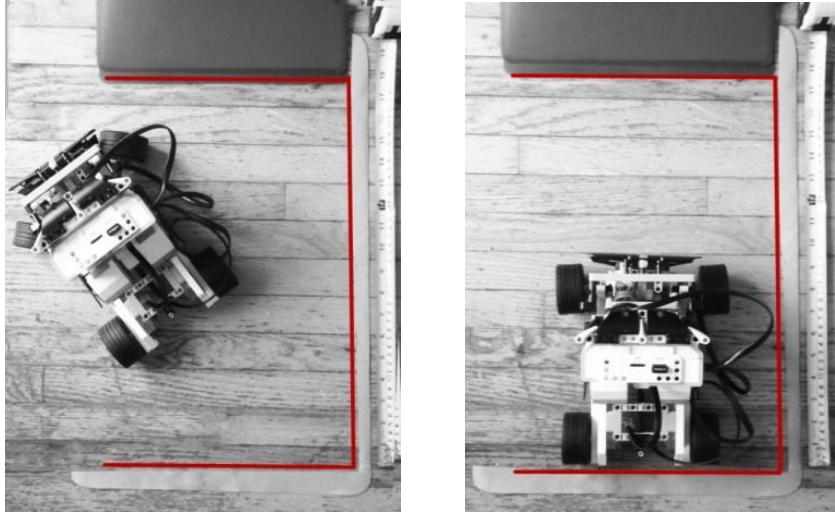


Figure 22: The experiment using Lego EV3 with SCS 1.875.

When the SCS is 1.4, the length of the parking spot is 33cm. Eight movements are required to finish the parking as it is shown below. We can see that there is also no any collision.



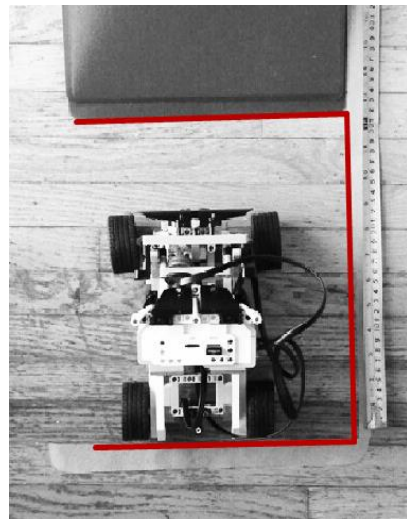
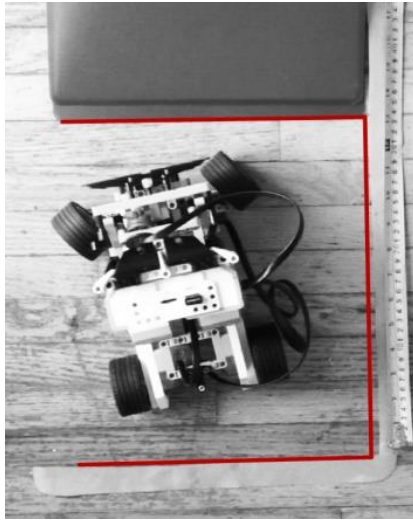
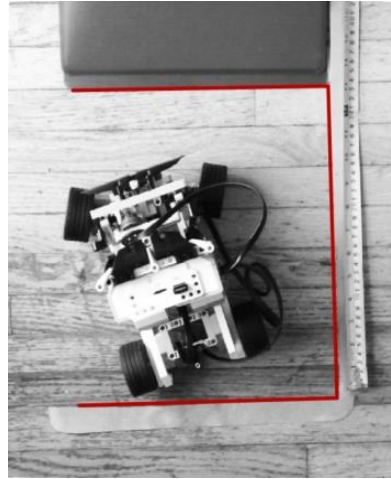
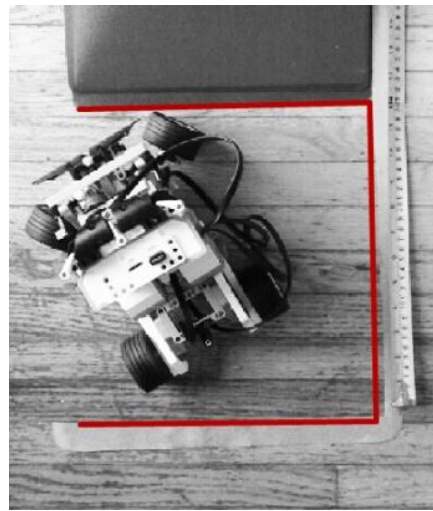
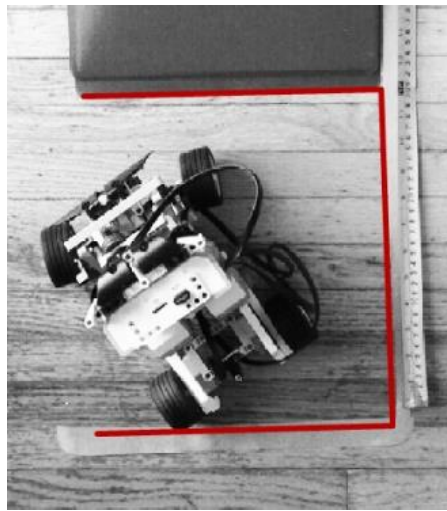
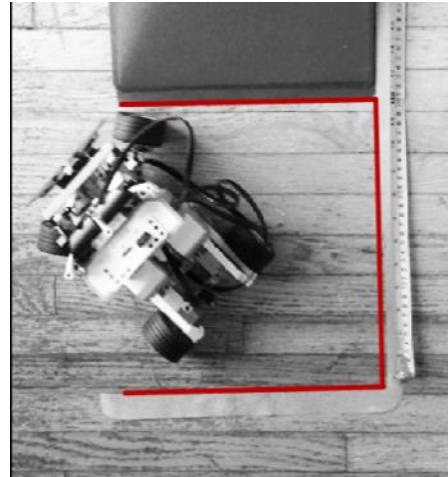
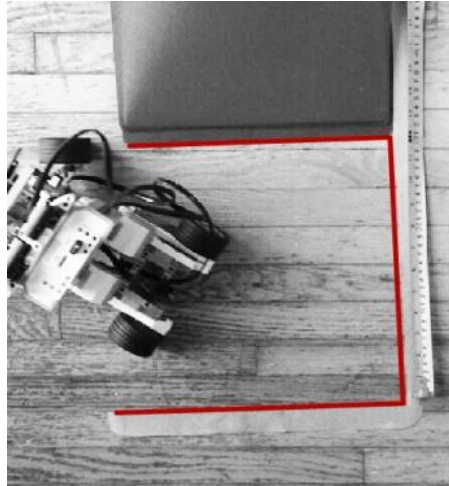


Figure 23: The experiment using Lego EV3 with SCS 1.4.

For SCS is 1.25, the length of the parking spot is 30cm. Seven steps are required to finish the parking process as shown in the following photos. We observe that for some of the movements, the head of the vehicle really close to the wall in this situation.



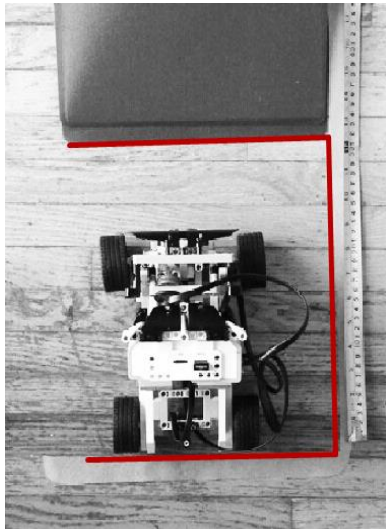
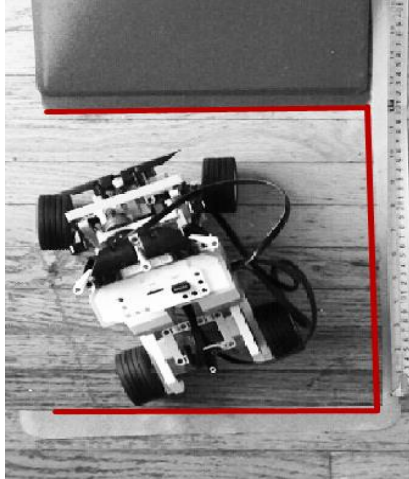
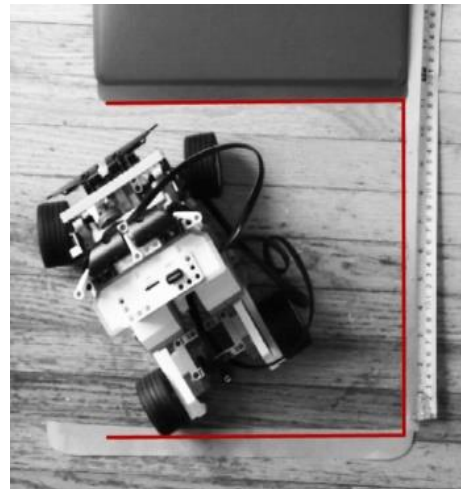
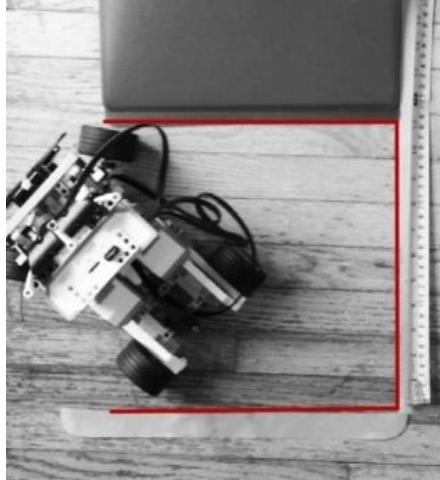
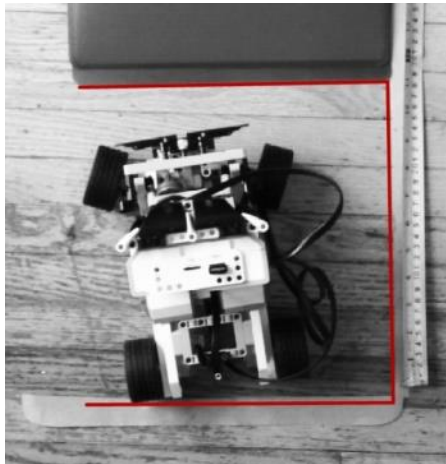
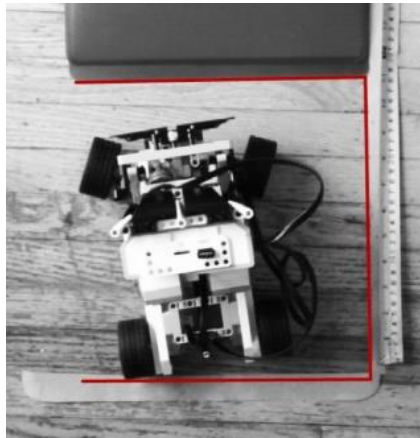
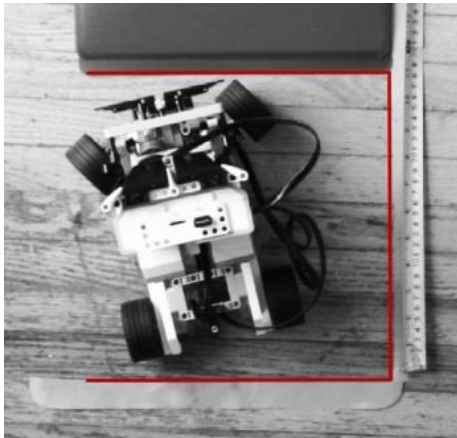
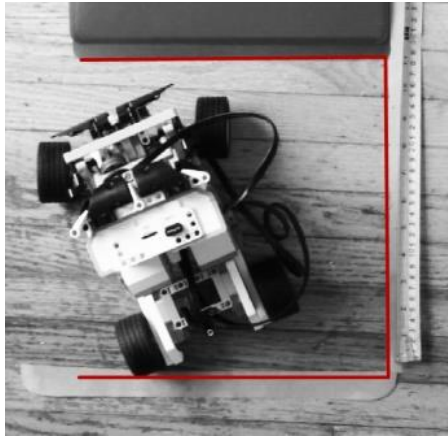


Figure 24: The experiment using Lego EV3 with SCS 1.25.

When the SCS is 1.2, the length of the parking spot is 28.8cm. The vehicle took twelve steps to finish the parking process. We have noticed there are some small collisions happened during the experiment.





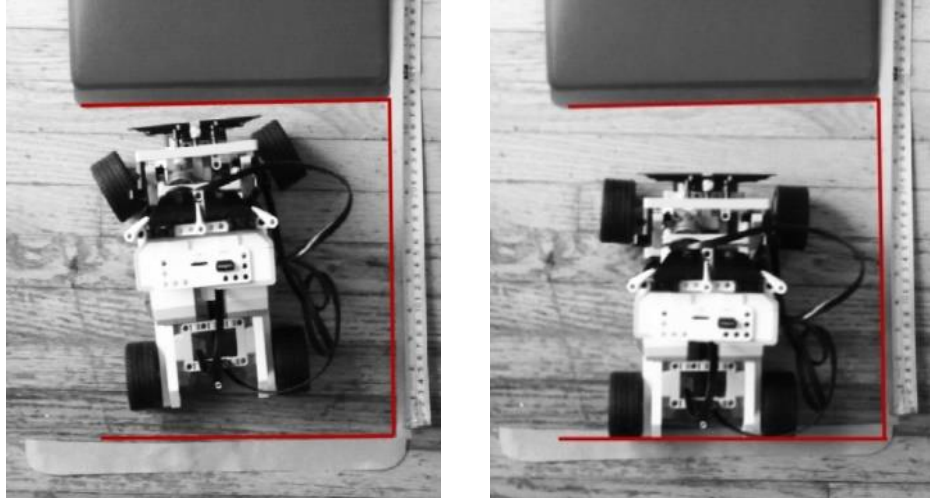
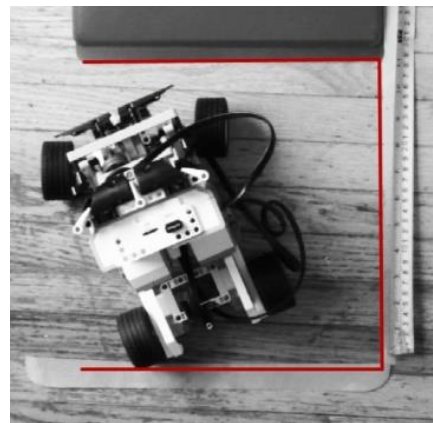
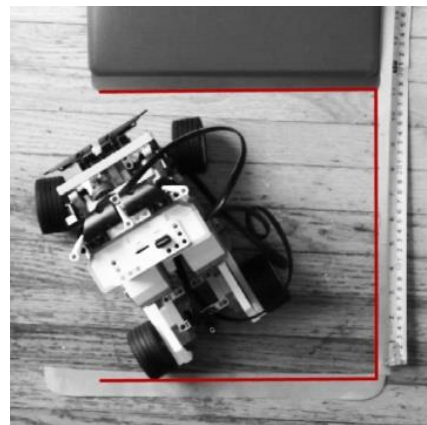
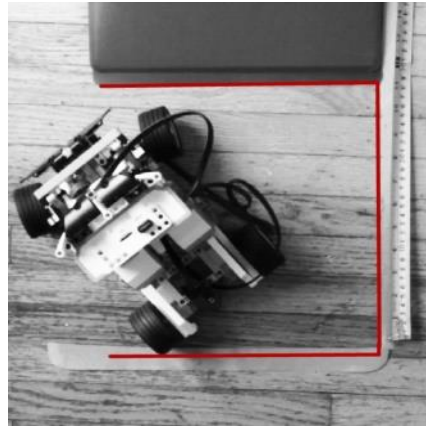
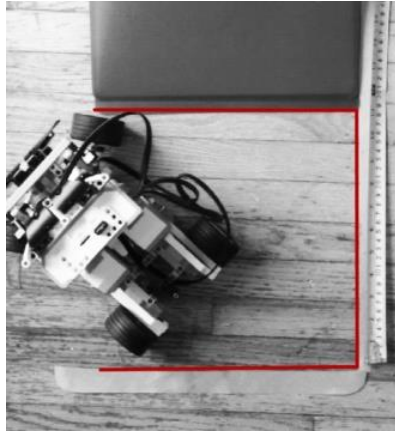


Figure 25: The experiment using Lego EV3 with SCS 1.2.

For SCS is 1.113, the length of the parking spot is 26.712cm. Also twelve movements are taken to finish the parking process. We also observed some small collisions.



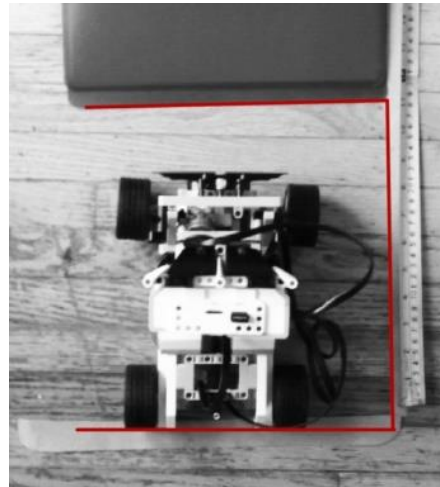
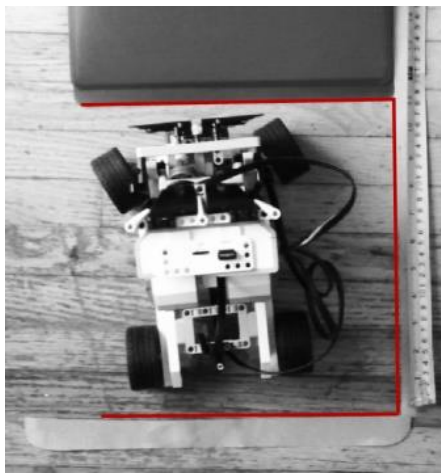
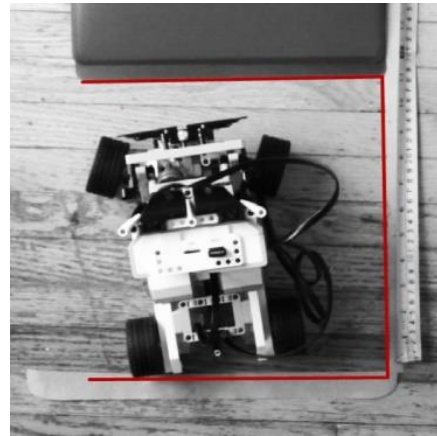
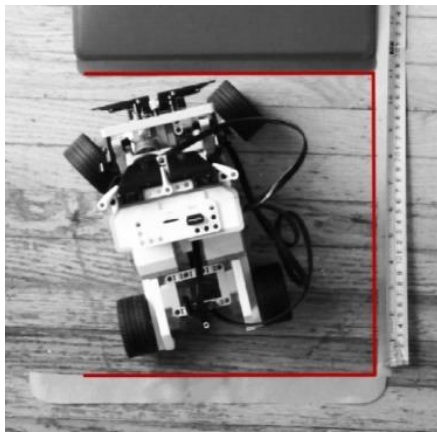
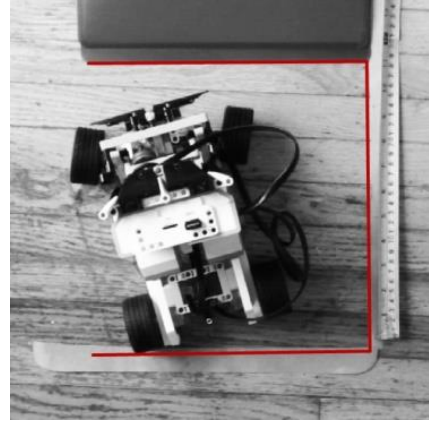
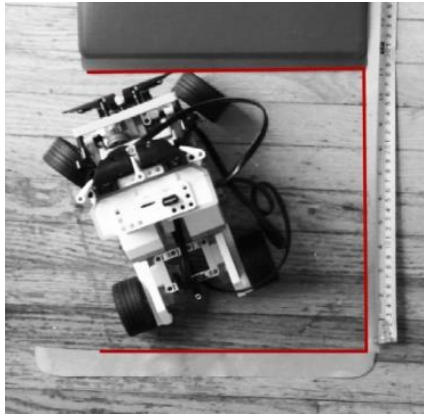


Figure 26: The experiment using Lego EV3 with SCS 1.113.

Chapter 5

Conclusion

5.1 Summary

In this thesis, we proposed an automatic parallel parking approach which improves the possibility of parking the vehicle into tight parking spaces. We used the robot kinematics to build the vehicle model and used geometric relationship to restrict and track the vehicle's movement to avoid collisions.

Compared with the other approaches, the proposed approach has some improved features:

- It is a geometrical approach which needs fewer sensors.

The only information that this approach requires from the sensors, is the length and width of the parking spot. All the calculations are based on the size of the parking spot and the size of the vehicle.

- It utilizes parking spaces more effectively.

Compared with previous approaches, this approach could potentially save parking space up to 17.6%, which means for the same rooms, this approach will be able to park more vehicles.

- It could be combined with both skill-based approach and path planning-based approach. Thus, the automatic parking will suit more situations.

Though some of the previous researches could make the parking process as a ‘one-step’ parking, but the requirement for spaces is larger. The combination of previous approach and the proposed approach will help promoting the efficiency of the automatic parking.

In experiments, compared with previous researches, the proposed method saved up to 17.6% parking spaces in simulation. When the proposed method is implemented on the LEGO EV3 robot, we can see it also works well most of the time. However, some collisions are observed since there are some limitations on the Lego EV3 we used in the experiment. The proposed approach requires a high level of accuracy when the vehicle is parking in an extreme tight space, but this requirement is not met on LEGO EV3. Even so, the experiment demonstrates that the proposed approach appears to be feasible.

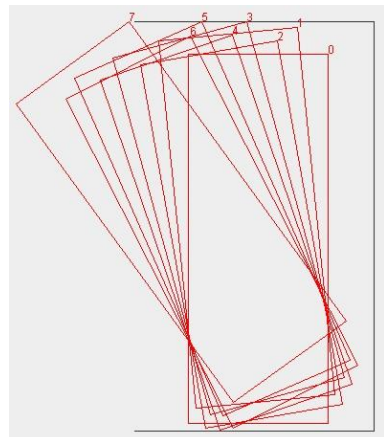
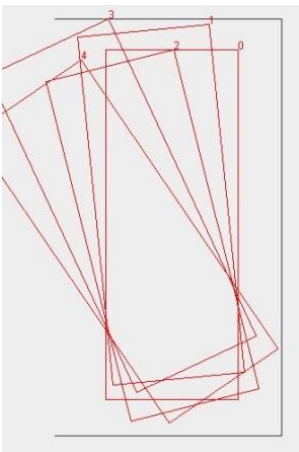
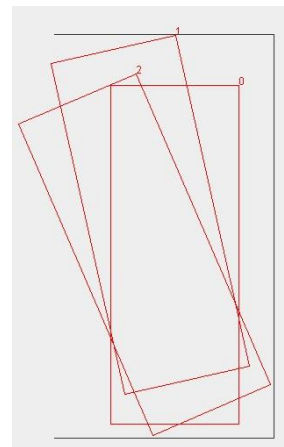
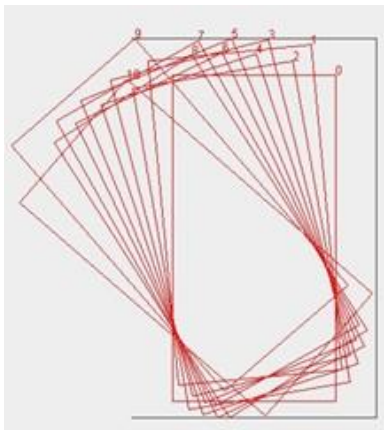
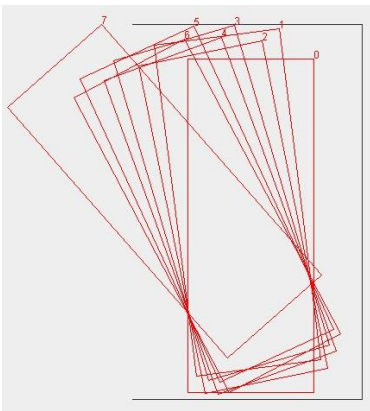
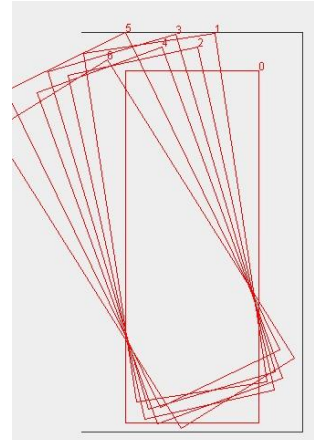
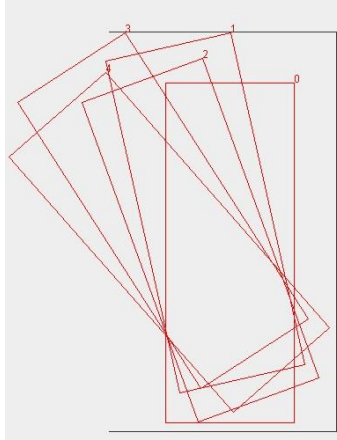
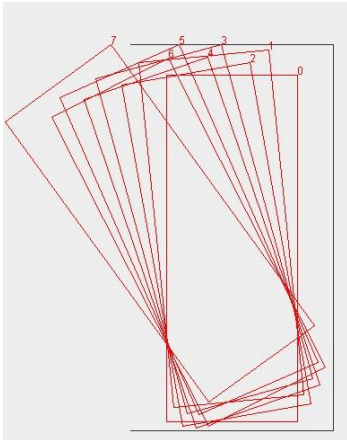
5.2 Future Work

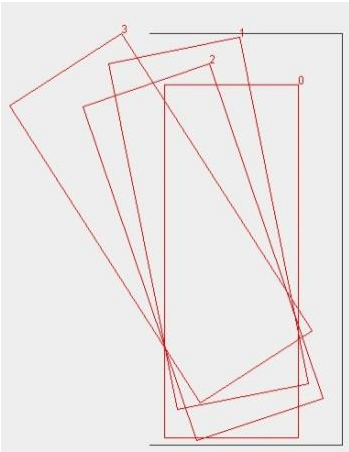
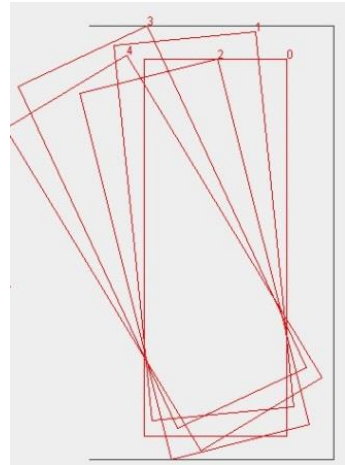
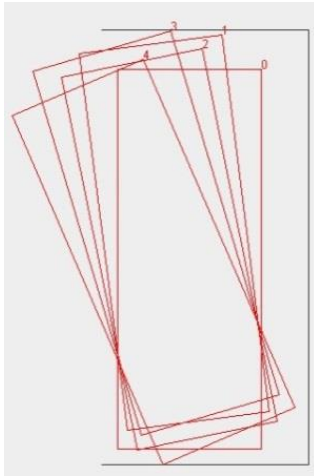
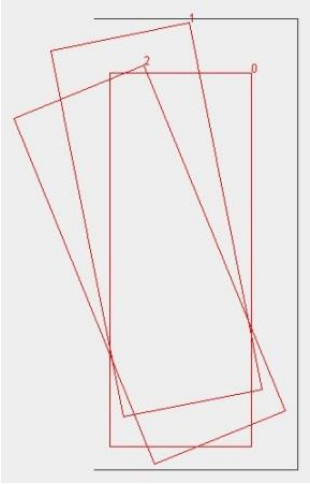
There is no doubt that the combination of skill-based approaches or path planning-based approaches will expand the limitation of the length of parallel parking spots in the future. What’s more, if the accurate performance could get improved, this method will help to provide a better experience on the automatic parallel parking for all the drivers and also will help to save and provide more spaces for parking in the urban

area in big cities. Research on how to keep the appropriate safe distance for this approach is also a task that is worth further study in the future.

Appendix

Some results of different length of the parking spot.





References

- [1] Mail News Online. *Drivers have 1,400 car park prangs a day: Vehicles getting bigger and spaces staying the same size blamed for accidents that cost drivers £716m a year.* November 2014. URL: <http://www.dailymail.co.uk/news/article-2838510/Drivers-1-400-car-park-prangs-day-Vehicles-getting-bigger-spaces-staying-size-blamed-accidents-cost-drivers-716m-year.html#ixzz3P60ptQzJ> (Click date: 2015/02/27)
- [2] OICA. *2014 Production Statistics.* URL: <http://www.oica.net/category/production-statistics/> (Click date: 2015/02/27)
- [3] The globe and mail. *Smaller parking spots: Smart's plan for growth.* October 2012. URL: <http://www.theglobeandmail.com/report-on-business/smaller-parking-spots-smarts-plan-for-growth/article4672220/> (Click date: 2015/02/27)
- [4] Mail News Online. *The big squeeze: £500m bill for bumps and scrapes as cars get wider but parking spaces don't.* September 2013. URL: <http://www.dailymail.co.uk/news/article-2422335/Shrinking-parking-spaces-wider-cars-leads-500million-repairs-year-bumps-scrapes.html#ixzz3P6e8emXo> (Click date: 2015/02/27)
- [5]KPMG. *Self-driving cars: The next revolution.* 2012. URL: <http://www.kpmg.com/Ca/en/IssuesAndInsights/ArticlesPublications/Documents/self-driving-cars-next-revolution.pdf> (Click date: 2015/02/27)

- [6] Ryu, Y.-W., Oh, S.-Y., & Kim, S. (2006). Robust automatic parking without odometry using enhanced fuzzy logic controller. In *Fuzzy Systems, 2006 IEEE International Conference on* (pp. 521–527).
- [7] Zhao, Y., & Collins Jr, E. G. (2005). Robust automatic parallel parking in tight spaces via fuzzy logic. *Robotics and Autonomous Systems*, 51(2), 111–127.
- [8] Liang, Z., Zheng, G., & Li, J. (2012). Automatic parking path optimization based on Bezier curve fitting. In *Automation and Logistics (ICAL), 2012 IEEE International Conference on* (pp. 583–587).
- [9] Cheng, K., Zhang, Y., & Chen, H. (2013). Planning and control for a fully-automatic parallel parking assist system in narrow parking spaces. In *Intelligent Vehicles Symposium (IV), 2013 IEEE* (pp. 1440–1445).
- [10] Yang N., Liang S., & Wang H. (2011). Trajectory planning method and simulation research of parallel parking. *Electronic Measurement Technology*, 1, 011.
- [11] Wang, D., Liang, H., Mei, T., & Zhu, H. (2011). Research on self-parking path planning algorithms. In *Vehicular Electronics and Safety (ICVES), 2011 IEEE International Conference on* (pp. 258–262).
- [12] Lin, Z., Li, Q., Liang, Y., & Cheng, D. (2012). Parallel parking algorithm based on autonomous path planning. In *Application research of computers* (Vol. 29).
doi:10.3969/j.issn.1001-3695.2012.05.029

- [13] Li, J., & Sun, X. (2011). A parking algorithm for Parking Assist System. In *2011 International Conference on Electric Information and Control Engineering (ICEICE)* (pp. 86–90). doi:10.1109/ICEICE.2011.5777724
- [14] Liang, Z., Zheng, G., & Li, J. (2012). Application of Fuzzy Control Strategy in Automatic Parking Path Planning. In *2012 4th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)* (Vol. 2, pp. 132–135). doi:10.1109/IHMSC.2012.128
- [15] Kucuk, S., & Bingul, Z. (2006). Robot kinematics: forward and inverse kinematics. In Sam, C. (Ed.), *Industrial Robotics: Theory Modelling Control* (pp. 117-148). Germany: Pro Literatur Verlag.
- [16] Jiang, H. (2010). Research of parallel parking system control strategy. Thesis paper.
- [17] Jiang, K., & Seneviratne, L. D. (1999). A sensor guided autonomous parking system for nonholonomic mobile robots. In *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on* (Vol. 1, pp. 311–316).
- [18] *Lego EV3 official website*. 2014. URL: <http://www.lego.com/en-us/mindstorms/products/31313-mindstorms-ev3> (Click date: 2015/02/27)
- [19] *Lego lejos official website*. 2014. URL: <http://www.lejos.org/ev3.php> (Click date: 2015/02/27)
- [20] *Lego EV3 official website*. 2014. URL: <http://www.lego.com/en-us/mindstorms/build-a-robot/rac3-truck> (Click date: 2015/02/27)

Vita Auctoris

NAME: Wenyi Zhou

PLACE OF BIRTH: Beijing, China

YEAR OF BIRTH: 1990

EDUCATION: B.Sc., in Computer Science
Beijing University of Civil Engineering and Architecture.
2012

M.Sc., in Computer Science
University of Windsor.
2015