Design and Simulation of Fuzzy Logic Controlled Car Parking Assist System

A.Z.Ahmad Firdaus, Au Wei Shinn, M.S.M.Hashim, N.S.Khalid, I.I.Ismail and M.J.M.Ridzuan
School of Mechatronic Engineering, Universiti Malaysia Perlis, Pauh Putra Campus, 02600 Arau, Malaysia.
ahmadfirdaus@unimap.edu.my

***Abstract*—The simulation of the autonomous self-parking car model is introduced in this paper. The system using a fuzzy logic controller in monitoring the car into a parking spot. The car will be manoeuvred to the parking space along a curvilinear path called path planning that designed to prevent crashing with the neighbouring cars or obstacles. In this project, it has used the Simulink in MATLAB to compute the system. The Simulink has an easier way to set up the rules in fuzzy control. The system imitates human thinking in performing the parking whereby it controls the steering angle and speed of the car corresponded to the situation of the car.**

***Index Terms*—Autonomous Car Parking; Path Planning; Fuzzy Logic Control; Fuzzy Interference System (FIS).**

I. INTRODUCTION

The dramatic increase of cars nowadays has profoundly highlighted in an urban area. Statistics of the total registered vehicles from the year 2010 to the year 2015 recorded an average growth rate of 5.05% and yearly increment of 30.28% of registered vehicles in Malaysia [1]. Research also has shown that increment of cars is 3.3 times growth of population [2].

Besides that, some researchers have researched by videotaping the traffic flows on few spots that nearby public parking lot, interviewing drivers in public parking lot or self-involved in searching a parking space. The increment of cars has highly rise traffic problems, particularly in big cities.

There is existing intelligent car parking system in some country. However, the investment cost is much higher compared to investing in this project. People need an autonomous self-parking car system that able to manoeuvre the car into a parking slot accurately without crashing and perform four directional parking. Therefore, in this project, MATLAB Simulink has been used to develop the path planning for the system. Besides that, the system can be trained to park the car with Fuzzy Interference System (FIS) while controlling the steering angle and speed of the car according to the current feedback position of the car.

The scope of this project is to design a collision path planning method. With the help of simulation, it helps to determine the polarity of the steering angle and speed of the car for each directional parking. By knowing the polarity, the system can be developed using the FIS to monitor the outputs values base on the situation of the car.

II. LITERATURE REVIEW

A. Path Planning and Car Modelling

Generally, there are two types of parking lot design: parallel parking and angle parking. Most of the public parking lot employs angle parking system due to the capacity. Angle parking has more stalls, and it is much easier to park compared with parallel parking [3]. This is because angle parking required lesser manoeuvres to move or move out from the parking space. Thus, the project would focus on angle parking. Angle parking consists of many types of angle placements; the most common angle parking is 90-degree parking, also known as perpendicular parking. This project is focused on 90-degree parking in one-way traffic as shown in Figure 1 [4].

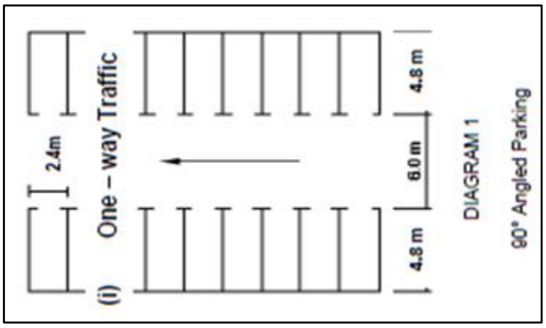

Figure 1: Type of angle parking in one-way traffic [4]

Perodua Myvi has been selected as the sample car to be represented in the simulation. The dimensions of the Perodua Myvi has been shown in Table 1 as well as in Figure 2. In this project, a model will represent the real car, and the parameters as shown below are the control of the path planning in the parking system.

Table 1
Vehicle Parameter

Parameter	Notation	Value
Length of the Vehicle between the Wheels	l	2440mm
Width of the Vehicle	b	1665mm
Distance of the Vehicle Front	l_1	709mm
Distance of the Vehicle Rear	l_2	546mm

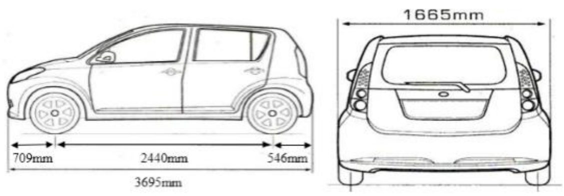

Figure 2: Dimensions of Perodua Myvi

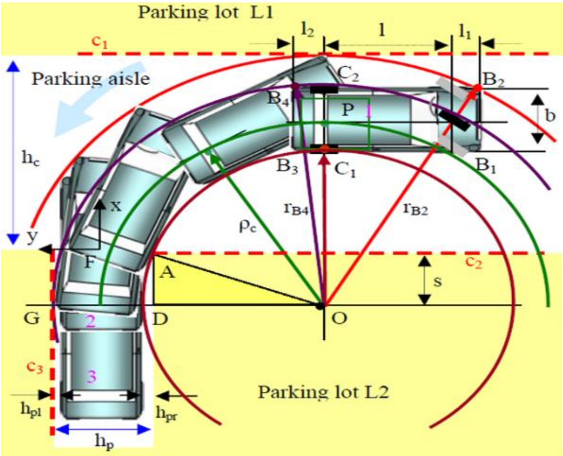

Figure 3: Geometry of collision-free reverse perpendicular parking manoeuvre [5]

Studies have agreed that the path planning able to make a curvature path turning without crashing the neighbouring vehicles[5-7]. It has denoted O as the centre of rotation of the car. It also called as Instantaneous Center of Rotation (ICR). The boundaries of the parking space shown in Figure 4 labelled as c_1, c_2 and c_3 that as a guideline. The circular motion of the car can be derived by finding the turning radius with the center O. Firstly, by defining the radius of the path turning of the car with constant steering angle, $\alpha_c = \alpha_{max}$:

$$\rho_c = {}^{l}/_{\tan(\alpha_c)} \tag{1}$$

The following steps are finding the maximum and minimum turning radius of the edges of the car, OB_2, OB_4 and OC_1, to avoid exceeding the boundaries as shown in the Figure 4. The radiuses can be derived by using Pythagoras Theorem application shown as in the Equation (2)-(4) below:

$$OB_2 = r_{B_2} = \sqrt{(l + l_1)^2 + (\rho_c + {}^{b}/_{2})^2} \tag{2}$$

$$OB_4 = r_{B_4} = \sqrt{{l_2}^2 + (\rho_c + {}^{b}/_{2})^2} \tag{3}$$

$$OC_1 = r_{C_1} = \rho_c - {}^{b}/_{2} \tag{4}$$

There are two conditions for the offset, s to ensure the collision-free perpendicular parking in a single manoeuvre for right turning situation; where $ICR\ O$ belongs to an interval of $s \in [-(\rho_c - {}^{b}/_{2}), 0]$ or $s \in [0, l_2]$. In order to avoid exceeding the boundaries, it required to define the maximum and minimum size of parking and parking corridor. With equation (2)-(3), to obtain the size of parking and parking corridor as shown below:

$$h_c = r_{B_2} - s = \sqrt{(l + l_2)^2 + \left(\rho_c + {}^{b}/_{2}\right)^2} - s \tag{5}$$

$$h_p = r_{B_4} - OD$$
$$= \sqrt{{l_2}^2 + \left(\rho_c + {}^{b}/_{2}\right)^2} - \sqrt{(\rho_c + {}^{b}/_{2})^2 - s^2} \tag{6}$$

where OD is to avoid collision between C_1 and A,

$$OD = \sqrt{(\rho_c + {}^{b}/_{2})^2 - s^2} \tag{7}$$

Simulation results the interval offset s, obtained from the Equation (4) and (5), are $s = -|s_{min}| = -1.91m$ and $s = -|s_{max}| = -0.46m$. Therefore, the path planning could be obtained with the ICR using the method above by calculating the range of the offset s. By referring on the frame F_{xy}, coordinate of point P is obtained as (x_P, y_P) and the angle between longitudinal vehicle and the x-axis of the frame F_{xy} is defined as θ that indicates the direction of the vehicle. Therefore, by applying Ackerman's rules of motion, the location of the car can be calculated as follows:

$$x_P = v\cos(\theta) \tag{8}$$

$$y_P = v\sin(\theta) \tag{9}$$

$$\theta = \frac{v}{l}\tan(\alpha) \tag{10}$$

B. *Fuzzy Logic Control*

The fuzzy logic control consists of many linguistic values IF-THEN rules, which describes the adequate control strategy. The rules basically form inference engine part as shown in Figure 4 [8], and are usually derived from the prior knowledge base or existing data pattern. In MATLAB, fuzzy logic toolbox can be used to integrate the fuzzy system into the simulation with SIMULINK. Previously such control design has been implemented in power electronics circuit design [9].

The result of fuzzy rule-based will be in the form of fuzzy sets. These linguistics results cannot directly produce the desired output in presentable form. Therefore, it needs to be going through with defuzzification method to obtain the result. Defuzzification is a reverse process of fuzzification that converting membership value into the crisp value as. In other words, defuzzification is a process that transforms the fuzzy set that obtained by inference into a crisp value that can be used as the desired output where the inference engine is used to simulates human reasoning process by making fuzzy inference on the value of the input with the IF-THEN rules.

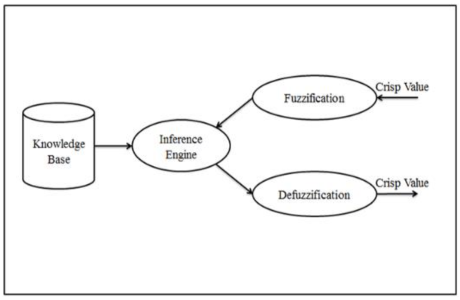

Figure 4: Fuzzy logic control [8]

III. METHODOLOGY

A. *Mathematical Model*

In the mathematical derivation, by scaling down the real size of a car to a mobile car-like robot, the path planning is derived for the mobile robot to move along with the x-axis on the local frame. Figure 5 shows the configuration of the path planning for the mobile robot. The mobile robot replicates a real car which the front wheels can make left and right turning with a maximum angle of 30 degrees while the rear wheel of the robot is fixed parallel and installed with a motor. The motor allowed the robot to move forward and reverse. By referring Ackerman's theorem, the car has a local reference frame denotes as Y_L and X_L with referring the global reference frame denotes as Y_R and X_R. The y-axis of local reference frame will always towards the centre of rotation on the global reference frame.

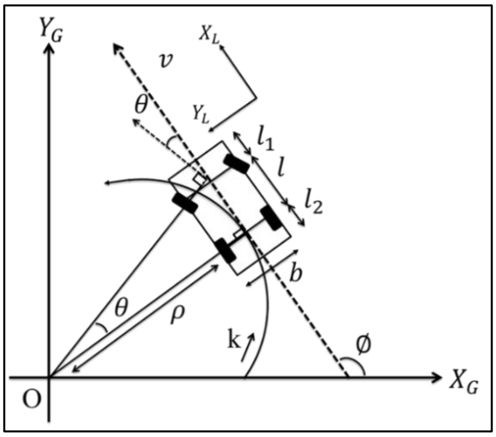

Figure 5: Configuration of Path Planning

Table 2
Description of the Kinematics Model Car

O	Centre of rotation	v	Speed of the car
Y_G and X_G	Axis of global reference frame	θ	Steering angle
Y_L and X_L	Axis of local reference frame	l	Length of between the wheels
$\emptyset$	Orientation of the car	l_1	Length of front bumper
k	Curvature of the path	l_2	Length of rear bumper
ρ	Radius of curvature	b	Width of the car

As mentioned before, Perodua Myvi is chosen as the test car for this project with parameters shown in Table 2. Thus, the radius of curvature with the constant steering angle of 30 degrees is calculated as below:

$$\rho_c = {}^{2.44}\!/_{\tan(30)} = 4.2\text{m} \tag{11}$$

and the path curvature is

$$k = {}^{1}\!/_{\rho} = {}^{1}\!/_{{}^{l}\!/_{\tan(\theta)}} = {}^{\tan(\theta)}\!/_{l} \tag{12}$$

With the circular motion formula, we obtained $\emptyset = k \times v$ as the velocity of the mobile robot is measured at 0.7m/s with the actual speed of the car on "D gear". Thus, the coordinates of the position can be determined by the orientation of the robot. The equations that derived:

$$\dot{\emptyset} = k \times v = {}^{\tan(\theta)}\!/_{l} \times v = \frac{v}{l}\tan\theta$$
$$\Rightarrow \dot{\emptyset} = \frac{0.7}{2.44}\tan(30) = 0.1656 \tag{13}$$

Therefore, the movement of the car along $\dot{x}$ and $\dot{y}$ axes can be determined:

$$\dot{x} = v\cos\emptyset = 0.7\cos(30) = 0.606\text{m/s} \tag{14}$$

$$\dot{y} = v\sin\emptyset = 0.7 \ \sin(30) = 0.35\text{m}/s \tag{15}$$

The turning angle of the mobile robot must be coordinated with the speed of the mobile robot as it moves. Thus, the constant steering angle with constant speed with results in higher speed along the x-axis and slower movement along the y-axis. By taking the dimension of the car and the parking, the path planning can be determined.

$$\begin{aligned} OB_2 &= r_{B_2} \\ &= \sqrt{(2.44 + 0.709)^2 + \left(4.2 + {}^{1.665}\!/_{2}\right)^2} \\ &= 5.937m \end{aligned} \tag{16}$$

$$OB_4 = r_{B_4} = \sqrt{(0.546)^2 + \left(4.2 + {}^{1.665}\!/_{2}\right)^2} = 5.062m \tag{17}$$

$$OC_1 = r_{C_1} = 4.2 - {}^{1.665}\!/_{2} = 3.368m \tag{18}$$

The parking size and parking corridor of single traffic in the car parking guideline of Wilayah Persekutuan Kuala Lumpur has been chosen as the test parking space. Hence, $h_c = 6m$ and $h_p = 2.4m$, the radius of curvature, ρ can be determined and by finding the center of rotation, $O(x_O, y_O)$ the position for the car to make perpendicular parking can be found. With respect to the inertial frame, F_{xy} the position of O are based on $x_O \in [-|s|_m, -|s|_{max}]$ and $y_O = -\rho_c$ to move to the parking place.

$$\begin{aligned} -|s|_m &= -\sqrt{\left(\rho_c - {}^{b}\!/_{2}\right)^2 - \left(\rho_c - {}^{h_p}\!/_{2}\right)^2} \\ &= -\sqrt{\left(4.2 - {}^{1.665}\!/_{2}\right)^2 - \left(4.2 - {}^{2.4}\!/_{2}\right)^2} \\ &= -1.53m \end{aligned} \tag{19}$$

$$\begin{aligned} -|s|_{max} &= h_c - r_{B_2} = 6.0 - 5.937 \\ &= 0.063m \end{aligned} \tag{20}$$

$$\therefore x_O \in [-1.53m, 0.063m] \ and \ y_O = -4.2m \tag{21}$$

Thus, from Equation (21), $x_O = -1.53m$ and $y_O = -4.2m$ and the position, that the car makes the parking, $P(x,y) = (6.2m, 4.2m)$ is found as shown in Figure 5. As concerned previously, the radius of $r_{B_2} - |s|_m = 5.937 - 1.53 = 4.407m$ is less than h_c. So, the car would not collide on obstacles in boundary c_1. While the car also would not exceed boundary c_2 and c_3. From Equation (7), $OD = 3.0m$, while $OG = \text{OD} + h_p = 3 + 2.4 = 5.4m$. Apparently, OC_1 is greater than OD while OB_4 is less than OG. Thus, the collision-free path planning is determined.

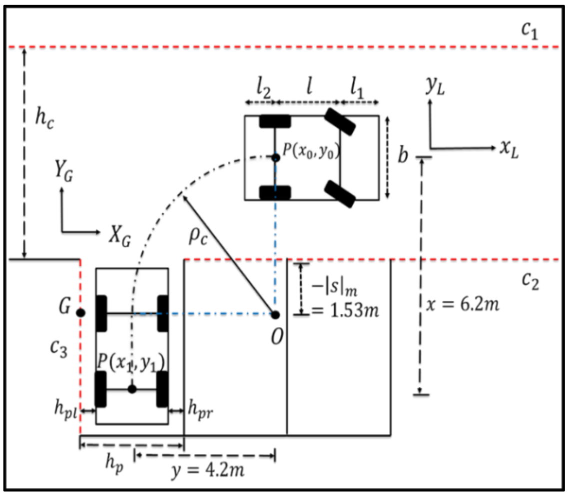

Figure 5: Geometry of perpendicular reverse right

B. Open-Loop System Simulation Design

Using MATLAB Simulink, an open-loop system is developed to obtain the output result for monitor the steering angle and speed of the car. The simulation model is designed with the Equation (8)-(10) by assuming the maximum constant speed of the car is 0.7m/s and max steering angle is 45 degree. The speed of the car is controlled in a manner of the ratio by default FullBreak, QuarterSpeed, HalfSpeed and FullSpeed as shown in Table 3. The result of the open-loop simulation is to prove that this model is correct and suitable for manipulating the path planning.

Table 3
Speed Ratio

Speed	Ratio
FullSpeed	1.00
HalfSpeed	0.50
QuarterSpeed	0.25
FullBreak	0.00

Figure 6 below shows the Simulink of the Open-loop system that derived from the model of path planning. The XY Graph will plot the results of the path planning at each time steps. While the steering angle and speed are using a repeating sequence with a set of value that based on logic, think in designing the parking method.

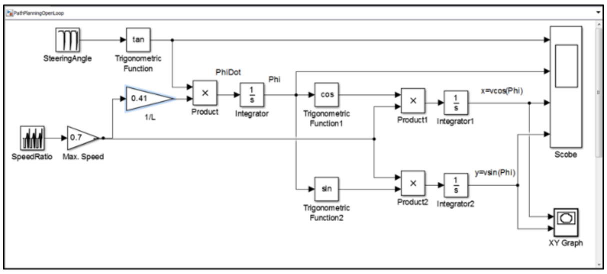

Figure 6: Simulink of Open-Loop System

As noticed the values of the steering angle and speed ratio, it has a similar pattern for every path motion. Therefore, the simulation results have obtained in Figure 9 and 10 to show the path planning motion for forward left, forward right, reverse left and reverse right respectively. The mobile robot is assumed oriented on the side of the vacant parking slot, and it will move itself to the parking place along the path. The mobile robot is initial at the red spot, and the parking is targeted it will move into the parking space by itself in either direction. The open-loop simulation test for the system by using the measurement of the steering angle and speed of the car to obtain path planning of the model. The results are shown in xy-graph by showing the distance travelled, and the distance is measured in meter.

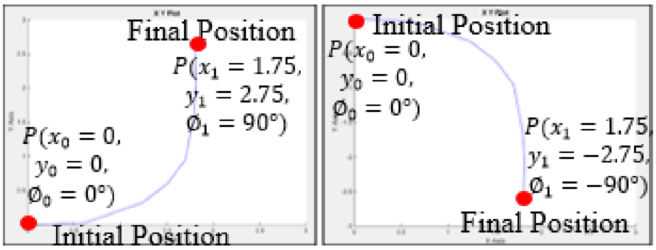

Figure 7: Graph for forward left (1) and right (2) motion

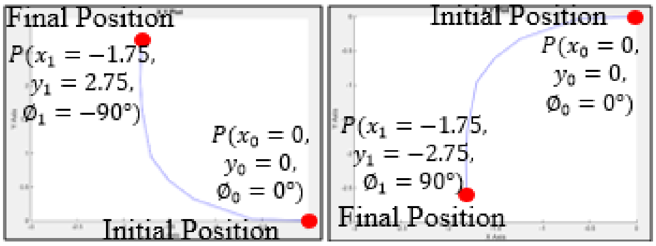

Figure 8: Graph for reverse left (3) and right (4) motion

C. Closed-Loop System Simulation Design

The closed-loop system is to determine the feedback output values by calculating the changes on each time value. Figure 9 shows the block diagram of the closed-loop simulation. The fuzzy logic controller will manipulate the steering angle and speed of the car corresponded to the inputs from the error of the three aspects. Figure 10 is the closed-loop system of the model in MATLAB Simulink. The system derives the system plant in the simulation model and inputs with a fuzzy logic controller. Three inputs represent the data from the sensor to the FLC. The FLC will manipulate the inputs to output the model with the feedback obtained by the model.

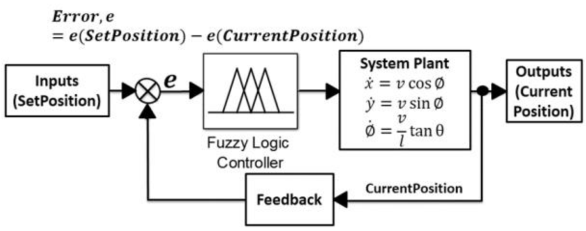

Figure 9: General control block diagram of closed-loop simulation

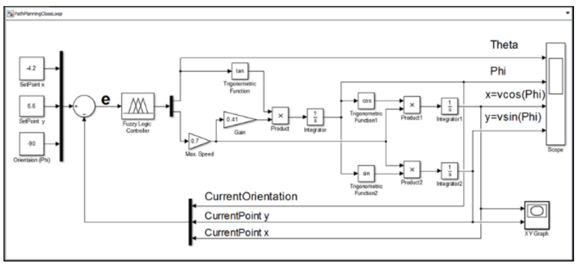

Figure 10: Simulink of Closed-Loop System

1) Fuzzy Logic Controller Design

Three input parameters and two output parameters are obtained in FIS. The input parameters obtained from the error of the position and orientation of the car while the output parameters present the required steering angle and speed to park the car into the parking space. The polarity of each direction is determined and shown in Figure 11. The FIS will manipulate the parameters of the car heading to respective direction Figure 12 shows the design case for the output parameters at each directions parking.

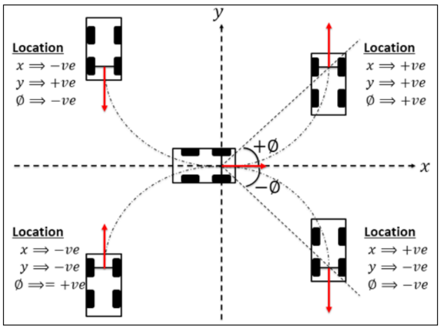

Figure 11: Polarity of the direction

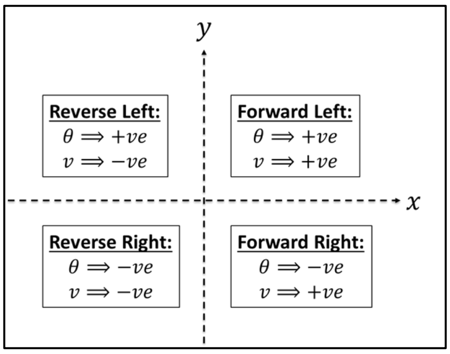

Figure 12: Output parameters design

2) Fuzzy Membership Function

The fuzzy membership function is to determine the degree of trueness or percentage of agreed on the particular membership function. Each membership function has a range, and the degree of trueness is determined within the range. Seven membership functions in each parameter are zero, positive and negative range of maximum, average and minimum value respectively. The steering angle has the similar labelled membership function as the input parameter while the speed is labelled as FullBrake, positive and negative range of highest, average and lowest speed respectively.

3) Fuzzy Rule-Based Condition

The fuzzy rules are fixed according to the conditions and test data based on Open-loop simulations. It is like a guidance rule to ensure the system know what to do on particular conditions.

Table 4
Fuzzy Rule-Base

Error	y_{max}	y_{ave}	y_{min}	$\emptyset_{max}$	$\emptyset_{ave}$	$\emptyset_{min}$
x_{max}	v_{max}	v_{max}	v_{max}	θ_{min}	θ_{max}	θ_{ave}
x_{ave}	v_{max}	v_{max}	v_{ave}	θ_{max}	θ_{ave}	θ_{max}
x_{min}	v_{max}	v_{ave}	v_{min}	θ_{ave}	θ_{max}	θ_{min}

The fuzzy rule-base in designing the parking system has been developed in Table 4 according to the situation. Table 4 has shown the rule base fundamental in designing the rule of the fuzzy sets with three input parameters and two output parameters.

IV. Results and Discussion

The results of the system have been obtained in MATLAB Simulink. The closed-loop simulation of the model computes the results. With the actual size measurement of the car and car park, a scale down ratio has been made to be replaced by the real apparatus.

The car is assumed located at the centre of four different direction parking along with x-axis in local frame. The fuzzy logic controller monitors the system according to with the rule-base has made. The direction of the car is differentiated with the polarity of the car. There are four different direction parking results in the xy-graph. The coordinate of the car is calculated with collision-free path planning method.

A. Forward Left Parking

Figure 13 shows the forward left parking which plot in xy-graph. By using the kinematics model, the path of the car is developed, and it has shown how the car moved into the parking space. The estimated time for the parking is 10 second.

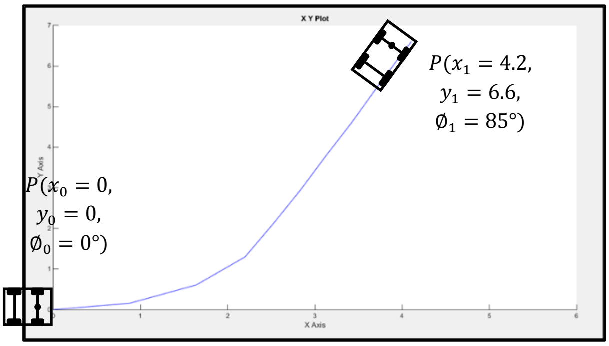

Figure 13: Forward left parking plotted in XY-graph

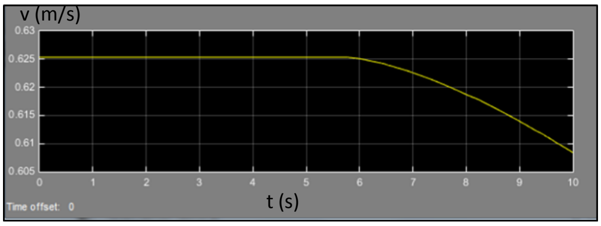

Figure 14: Speed of the forward left parking

The speed of the car is 0.625m/s shown in Figure 14, the limits of the speed is within 0m/s to 0.7m/s measured from the actual speed of D gear of Perodua Myvi. The error of Phi can be calculated based on the output:

The final orientation of the car is, $\emptyset = 85°$
Thus error,

$$e_\emptyset = \frac{\text{Targeted Orientation} - \text{Resulted Orientation}}{\text{Targeted Orientation}}$$
$$= \frac{90° - 85°}{90°} \times 100\% = 5.56\%$$

B. *Reverse Left Parking*

Figure 15 shows the forward left parking which plot in xy-graph. The speed of the car is -0.625m/s shown in Figure 16. The error of Phi can be calculated based on the output. The final orientation of the car is, Ø=-85°e_Ø=-5.56%.

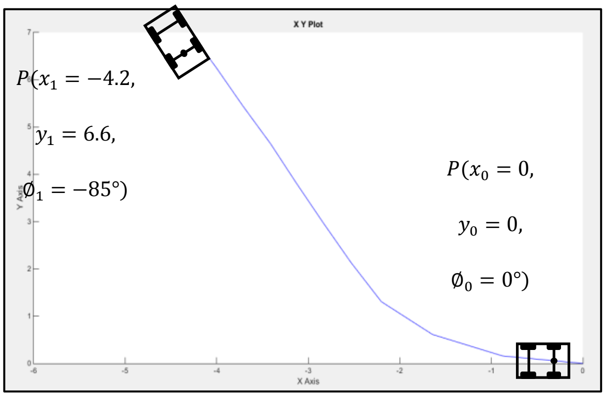

Figure 15: Reverse left parking plotted in XY-graph

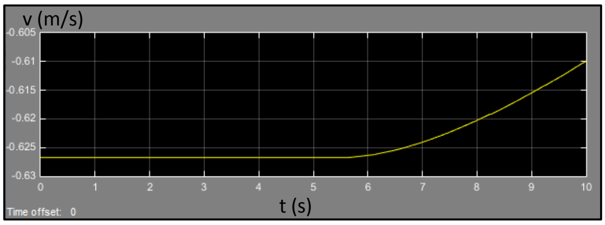

Figure 16: Speed of the reverse left parking

C. *Forward Right Parking*

Figure 17 shows the forward right parking which plot in xy-graph. The speed of the car is 0.625m/s shown in Figure 18. The error of Phi can be calculated based on the output. The final orientation of the car is, $\emptyset = -35° e_{\emptyset} = -61.11\%$

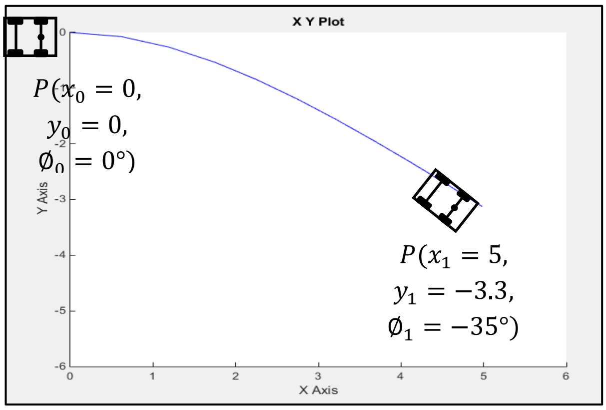

Figure 17: Forward right parking plotted in XY-graph

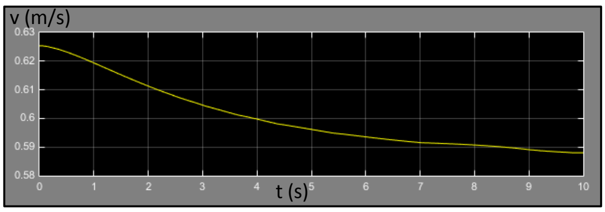

Figure 18: Speed of the forward right parking

D. *Reverse Right Parking*

Figure 19 shows the reverse right parking which plot in xy-graph. The speed of the car is -0.625m/s shown in Figure 20. The error of Phi can be calculated based on the output. The final orientation of the car is, $\emptyset = 35 e_{\emptyset} = 61.11\%$.

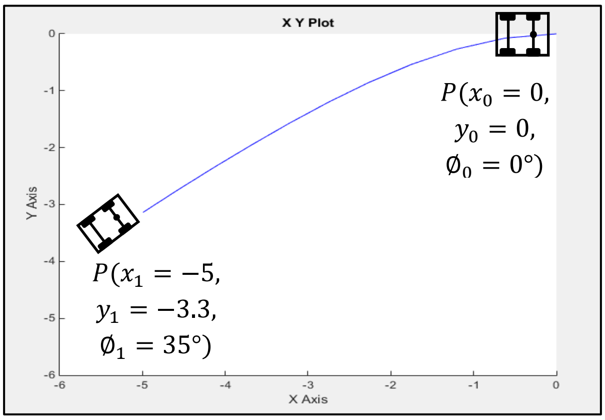

Figure 19: Reverse right parking plotted in XY-graph

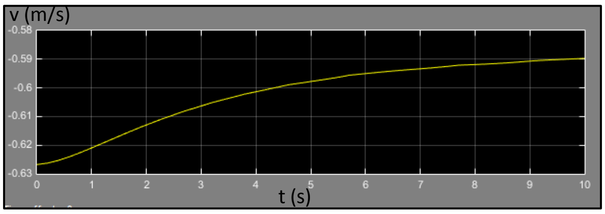

Figure 20: Speed of the reverse right parking

In the forward and reverse left parking, the fuzzy logic controller managed to obtain the nearest result in the orientation of the car with 94% accuracy to the expected orientation of the car. However, the system still has its weakness which has shown by the right parking for both forward and reversed motion. It results in a high percentage of error of the orientation and also the position of the car.

V. CONCLUSION

The design of the model in autonomous self-parking system is approached to achieve the objective. In order to park the car into the parking space without crashing neighbouring cars or obstacles, a collision-free path planning calculation is developed, corresponding to the actual parameters of the car and also the parking space dimension. The basic kinetic mechanism has been applied in designing the general motion for the path planning along with xy-axis. For 90degree angled parking, the final orientation of the car must be perpendicular to the original orientation of the car. In the open-loop simulation, the car model has been tested with four basic parking position: forward left, forward right, reverse left and reverse right, where parking objective has been achieved. For the closed-loop simulation, the fuzzy logic controller has been implemented for automatic control. Based on instantaneous car position, final parking car position and present parking options, the Fuzzy logic controller controls steering angle and also the speed of the car such that the car maintains its course towards parking destination. The model applies closed-loop simulation system to give the instantaneous position feedback to the control system. The error is minimised as result of fuzzy inference system design in the fuzzy logic controller. However, simulation results showed high orientation positional accuracy of 94% only for left parking modes. Further improvement on fuzzy rule base design is expected to improve accuracy on right orientation parking modes.

ACKNOWLEDGEMENT

The author would like to express gratitude and acknowledgement to School of Mechatronic Engineering, Universiti Malaysia Perlis for facilitating the research works.

REFERENCES

[1] Official portal of road transport department Malaysia. Retrieved May 8, 2016, from Jabatan Pengangkutan Jalan Malaysia, http://www.jpj.gov.my/en/web/guest/statistics.

[2] O. Chiu Chuen, M. R. Karim, and S. Yusoff, "Mode choice between private and public transport in Klang valley, Malaysia," *The Scientific World Journal*, vol. 2014, pp. 1–14, 2014.

[3] C. M. Monsere, "Safety Comparison of Angle and Parallel Parking," Oregon Department of Transportation, Traffic Management Section Traffic Engineering Services Unit, Mar. 15, 2011.

[4] Guidelines For Car Parking And Internal Traffic Circulation. (2014).

[5] Petrov, Fawzi Nashashibi, Member, IEEE, and Mohamed Marouf, P., Nashashibi, F., & Marouf, M. (n.d.). Path Planning and Steering Control for an Automatic Perpendicular Parking Assist System.

[6] D. C. Shoup, "Cruising for parking," *Transport Policy*, vol. 13, no. 6, pp. 479–486, Nov. 2012.

[7] J. Lee and M. Kim, "Design of fuzzy controller for car parking problem using evolutionary multi-objective optimization approach," *2010 IEEE International Symposium on Industrial Electronics*, Jul. 2010.

[8] J.-Y. Lee and J.-J. Lee, "Multiple Designs of Fuzzy Controllers for Car Parking Using Evolutionary Algorithm," in Proceedings of International Conference on Mechatronics, Kumamoto Japan: Department of Electrical Engineering and Computer Science, KAIST, 2007, p. 6.

[9] A. Z. A. Firdaus, M. Normahira, K. N. Syahirah, and J. Sakinah, “Design and simulation of Fuzzy Logic Controller for boost converter in renewable energy application,” in *Proceedings - 2013 IEEE International Conference on Control System, Computing and Engineering, ICCSCE 2013*, 2013.